



Guide du développeur pour Amazon Nova 2

Amazon Nova



Amazon Nova: Guide du développeur pour Amazon Nova 2

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Qu'est-ce qu'Amazon Nova 2 ?	1
Concepts clés	1
Modèles Amazon Nova	2
Que pouvez-vous construire ?	2
Avantages	3
Comment fonctionne Amazon Nova	3
Tarification	4
Étapes suivantes	4
Nouveautés d'Amazon Nova 2	5
Nouveaux modèles	5
Nouvelles fonctionnalités et capacités	5
Intégrations multimodales Nova	5
Réflexion et raisonnement approfondis	5
Outils intégrés	6
Création d'agents IA	6
Meilleure compréhension des documents	6
Meilleure compréhension de la vidéo	6
Personnalisation de modèles	7
Amazon Nova Forge	7
Étapes suivantes	7
Prise en main	8
Prise en main dans la console	8
Explorer le terrain de jeu de texte	9
Démarez avec l'API	9
Obtention des informations d'identification pour accorder un accès programmatique	11
Attribution des autorisations Amazon Bedrock à un utilisateur ou à un rôle	13
Générer une réponse à une invite de texte à l'aide d'un modèle Amazon Nova	13
Inférence de base	15
Important : Configuration du délai d'expiration	16
Sujets d'inférence de base	16
Utilisation de l'API Converse	17
Structure de la demande	17
Utilisation des instructions du système	18
Paramètres d'inférence	19

Utiliser le raisonnement	20
Utilisation de l'API Invoke	23
Structure de la demande	23
Paramètres de demande	24
Réponses en streaming	24
Streaming avec ConverseStream	24
Streaming avec InvokeModelWithResponseStream	25
Types d'événements en streaming	25
Utilisation des intégrations Amazon Nova	26
Fonctions principales	26
Génération d'intégrations	27
Inférence à la demande	31
Avantages	31
Utilisation de l'inférence à la demande	31
Tarification	32
Quotas et limites	32
Utilisation des fonctionnalités de Nova	33
Raisonnement	33
Compréhension multimodale	34
Types de contenu pris en charge par modalité	34
Compréhension des images	35
Compréhension des vidéos	38
Compréhension des documents	40
Discours à parole	44
Fonctions principales	44
Prise en main	45
Définir les importations et la configuration	45
Définir la classe SimpleNovaSonic	46
Initialiser le client	47
Gérer les événements	47
Démarrer la séance	47
Gérer l'entrée audio	50
Terminer la session	52
Gérer les réponses	52
Lire l'audio	54
Capturer l'audio	55

Exécution de la fonction principale	56
Exemples de code	57
Exemples pour débiter	57
Cas d'utilisation avancés	58
Atelier pratique	59
Invites de conversation vocale	59
Directives importantes	59
Système de référence recommandé : invite vocale	60
Configuration de l'invite vocale	60
Configuration rapide du système	61
Accord sur le genre pour les langues sexospécifiques	62
Chaîne de pensée pour le discours : raisonnement constitutionnel	63
Utilisation abusive de phrases suggérées	64
Concepts de base	66
Cycle de vie des événements	66
Séquence du flux d'événements	68
Gestion des événements d'entrée avec l'API bidirectionnelle	68
Gestion des événements de sortie avec l'API bidirectionnelle	77
Barge-in	84
Contrôlabilité à tour de rôle	86
Entrée intermodale	87
Support linguistique et fonctionnalités multilingues	90
Gestion de l'historique des discussions	92
Configuration de l'outil	99
Appel d'outils asynchrone	106
Intégrations	109
Agents à mèches	109
Intégrations du framework	112
Systèmes avancés avec Nova	123
Présentation de	123
Pensée étendue	123
Comment fonctionne la pensée étendue	123
Options de configuration	126
Modèles pris en charge	127
Réflexion approfondie avec utilisation d'outils	127
Comprendre les jetons de raisonnement et les prix	128

Questions fréquentes (FAQ)	128
Utilisation d'outil	129
Comprendre le flux de travail d'utilisation des outils	129
Création d'un outil	130
Appelez un outil	133
Outils du système intégrés	136
Création des agents d'IA	139
Création d'un agent	139
Invocation d'un agent	141
Ancrage Web	144
Fonctionnement de l'ancrage Web	144
Comment utiliser l'ancrage Web	145
Disponibilité par région	146
Structure de réponse	146
Sécurité de mise à la terre	147
Gestion des erreurs	147
Autorisations requises pour les outils intégrés	147
Ingénierie rapide	149
Qu'est-ce que l'ingénierie rapide	149
Commencer par une ingénierie rapide	149
Bonnes pratiques	150
Comprendre les rôles	150
Créez des instructions précises	151
Mettre l'accent sur certaines sections de l'invite	170
Utilisation du rôle système	171
Fournir des exemples (invites à quelques essais)	180
Systèmes d'appel d'outils	190
Techniques d'incitation avancées	191
Encourager les entrées multimodales	213
Directives générales relatives au multimodal	213
Compréhension des documents et des images	214
Compréhension des vidéos	38
Modération du contenu	226
Principaux principes	226
Configuration recommandée	227
MLCommons taxonomie universelle	227

Exemples d'incitation	229
Code et résolution des problèmes	235
Bibliothèque de codes	235
Exemples d'API Converse	235
InvokeModel Exemples d'API	255
Résolution des problèmes	260
Authentification et configuration	260
Accès au modèle refusé	261
Problèmes de disponibilité régionaux	261
Configuration du délai d'attente	261
Problèmes de réponse de l'API	262
Erreurs liées au mode raisonnement	263
Problèmes liés à l'utilisation des outils	264
Référence d'API et de SDK	268
Schéma de demande et de réponse	268
Structure de demande complète	268
Structure de réponse complète	273
Référence du SDK	276
Personnalisation du modèle Amazon Nova	278
Approches de personnalisation	278
Plateformes de personnalisation	279
Prérequis généraux	280
Formules Amazon Nova	281
Obtention de formules de base	282
Recettes Getting Forge	283
Modèles et algorithmes disponibles	283
Amazon Nova 2.0 Lite	284
Sur les emplois SageMaker de formation	291
Bonnes pratiques pour la personnalisation d'Amazon Nova	292
Kit de développement logiciel Nova Forge	297
Entraînement	304
Suivi des progrès au fil des itérations	341
Évaluation	346
Entraînement itératif	415
Sur SageMaker HyperPod	415
Exigences de calcul pour les modèles Amazon Nova 2	416

Bonnes pratiques pour la personnalisation d'Amazon Nova	292
Kit de développement logiciel Nova Forge	423
Configuration du cluster HP	423
Guide des commandes essentielles	423
Accès et configuration de Nova Forge	434
Entraînement	436
Évaluation	478
MLflow surveillance	531
Entraînement itératif	533
Présentation de	533
Comment ça marche	534
Quand utiliser la formation itérative	535
Exemple de flux de travail : SFT → RFT	536
Suivi des progrès au fil des itérations	538
Bonnes pratiques	540
Considérations de coût	542
Limitations	543
Résolution des problèmes	544
SageMaker Inférence	546
Caractéristiques	546
Modèles et instances pris en charge	547
AWS Régions prises en charge	550
Images de conteneurs prises en charge	550
Bonnes pratiques	550
Support	551
Prise en main	551
Référence des API	573
Évaluer les modèles	582
Détection des abus pour Amazon Nova Forge	589
Inférence Amazon Bedrock	590
Déployer un modèle personnalisé pour l'inférence à la demande	591
Inférence à la demande	597
Limitations	599
Nova Forge	600
Conditions préalables	600
Abonnez-vous à Nova Forge	601

Autres prérequis	601
Accès et configuration de Nova Forge	601
Obtenir les documents de Nova Forge	601
Étape 1 : Abonnez-vous à Nova Forge	602
Étape 2. Configuration de HyperPod l'infrastructure	608
Paramètres de modération du contenu	608
Pré-entraînement continu et mi-entraînement	609
Quels sont les points de contrôle intermédiaires et pourquoi sont-ils nécessaires ?	609
Quels sont les points de contrôle disponibles ?	610
Quel point de contrôle utiliser ?	611
Comment utiliser le mixage de données pour Nova 2.0 ?	612
Disséquer les catégories de mixage de données	614
Réglage précis supervisé	618
Introduction	618
Quand utiliser SFT ?	619
Quand utiliser le paramètre Efficient et quand utiliser le SFT à rang complet ?	620
Mixage de données pour SFT	620
Exemples de configuration YAML	621
Modéliser les points de contrôle	623
Approches de formation	624
Pré-requis avant de commencer	625
Une approche systématique pour réussir le SFT	625
Séquençage d'expériences et mélange de données	626
Préparer le jeu de données pour SFT	627
Méthodes d'évaluation	627
Bonnes pratiques	292
Préparer la configuration du Job de formation	630
Configurer un bloc de mixage de données	631
Comment lancer une offre d'emploi	638
Comment surveiller Job	646
Comment évaluer votre modèle après SFT ?	648
Leçons apprises et conseils	651
Apprentissage par renforcement	652
Apportez votre propre orchestrateur pour des évaluations agentiques en plusieurs étapes ..	652
Boîte à outils pour une IA responsable	654
Boîte à outils pour une IA responsable	654

Utilisation responsable	655
Consignes	655
Recommandations	656
Quotas	658
Demande d'augmentation des quotas Amazon Nova	
Amazon Nova	658
Contrôle	660
Sécurité	661
Créez des ressources avec CloudFormation	662
En savoir plus sur CloudFormation	662
Historique de la documentation	663
.....	dclxiv

Qu'est-ce qu'Amazon Nova 2 ?

Amazon Nova fournit des modèles de base multimodaux qui traitent le texte, les images, les vidéos, les documents et le discours. Avec la prise en charge de jusqu'à 1 million de jetons de contexte et des capacités de raisonnement avancées, les modèles Amazon Nova 2 vous permettent de créer des applications d'IA sophistiquées qui comprennent des entrées complexes et génèrent des réponses précises.

Vous pouvez créer des chatbots interactifs, analyser des documents et des vidéos, créer des agents d'intelligence artificielle dotés d'un raisonnement étendu et développer des applications vocales.

Concepts clés

Avant de découvrir les modèles Amazon Nova, familiarisez-vous avec les concepts de base suivants :

Modèles de fondation

Systèmes d'IA préentraînés disponibles en différentes tailles et fonctionnalités auxquels vous pouvez accéder via une API.

Inférence

Processus d'envoi d'une demande à un modèle et de réception d'une réponse générée.

Raisonnement

Capacité de réflexion étendue qui permet aux modèles de décomposer des problèmes complexes et de montrer leur step-by-step analyse avant de fournir des réponses.

Multimodal

La capacité de traiter et de comprendre plusieurs types de saisie simultanément : texte, images, vidéos et documents en une seule demande.

RAG (génération augmentée par extraction)

Technique qui combine les réponses du modèle avec vos propres sources de données afin de fournir des réponses contextuelles plus précises.

Modèles Amazon Nova

Amazon Nova 2 inclut les modèles suivants, chacun optimisé pour différents cas d'utilisation :

Modèle	Modalités d'entrée	Modalités de sortie	Cas d'utilisation
Nova 2 Lite	Texte, images, vidéos, documents	Texte	Applications à volume élevé donnant la priorité à la rapidité et à la rentabilité
Nova 2 Sonic	Discours, texte	Discours, texte	Applications à commande vocale avec temps de réponse rapides
Intégrations multimodales Nova	Texte, images, documents, vidéo, audio	Intégrations	Recherche sémantique, systèmes de recommandation et correspondance de similarité

Tous les modèles prennent en charge jusqu'à 1 million de jetons de contexte et peuvent générer jusqu'à 65 536 jetons en une seule réponse. Les modèles dotés de capacités de raisonnement peuvent effectuer une réflexion approfondie pour résoudre des problèmes complexes étape par étape.

Que pouvez-vous construire ?

Voici des exemples de ce que vous pouvez créer avec Amazon Nova :

- Assistant documentaire intelligent - Traitez des documents volumineux avec jusqu'à 1 million de jetons de contexte pour répondre aux questions et extraire des informations (avec RAG)
- Applications de raisonnement complexes - Résolvez des problèmes en plusieurs étapes avec une réflexion approfondie qui montre l' step-by-stepanalyse du modèle (ou avec un raisonnement)
- Pipeline d'analyse vidéo - Extrayez des informations, générez des résumés et identifiez les moments clés du contenu vidéo à grande échelle (Nova 2 Lite)

- Agent d'intelligence artificielle à commande vocale - Créez des agents conversationnels capables de comprendre la saisie vocale et de répondre en langage naturel (Nova 2 Sonic)

Avantages

Amazon Nova offre les avantages suivants :

Compréhension multimodale

Traitez le texte, les images, les vidéos, les documents et le discours en une seule demande. Les modèles Amazon Nova comprennent les relations entre les différents types d'entrée.

Contexte étendu

Support d'un million de jetons maximum vous permettant de traiter des bases de code complètes, de longs documents ou des conversations prolongées sans perdre le contexte.

Raisonnement avancé

Les modèles dotés de capacités de raisonnement permettent de décomposer les problèmes complexes et de présenter une step-by-step analyse, ce qui améliore la précision des tâches en plusieurs étapes.

Déploiement flexible

Accédez à des modèles via Amazon Bedrock sans aucune infrastructure pour les gérer, ou personnalisez les modèles grâce à un ajustement précis et à un apprentissage par renforcement.

Outils intégrés

Utilisez le Web Grounding pour accéder aux informations en temps réel et un interpréteur de code pour exécuter du code Python sans intégrations externes.

Comment fonctionne Amazon Nova

Les modèles Amazon Nova sont des modèles de base auxquels vous pouvez accéder via Amazon Bedrock. Le flux de travail de base est le suivant :

1. Votre application envoie une demande à Amazon Bedrock avec vos paramètres d'entrée et de configuration.
2. Le modèle Amazon Nova traite vos entrées en appliquant un raisonnement s'il est configuré.

3. Le modèle génère une réponse et la renvoie à votre application.

Vous pouvez améliorer les réponses en utilisant RAG pour intégrer vos données, en activant des outils intégrés ou en personnalisant les modèles grâce à un ajustement précis.

Tarification

La tarification d'Amazon Nova est basée sur les jetons d'entrée et de sortie traités. Les différents modèles ont différents niveaux de tarification :

- Nova 2 Lite - Optimisé pour le traitement rentable de gros volumes
- Nova 2 Sonic - Tarification équilibrée pour les applications à commande vocale

Pour obtenir des informations sur les prix actuels, consultez les [tarifs d'Amazon Bedrock](#).

Étapes suivantes

- Pour en savoir plus sur les nouvelles fonctionnalités d'Amazon Nova 2, consultez [Nouveautés d'Amazon Nova 2](#).
- Pour commencer à utiliser Amazon Nova, consultez [Commencer à utiliser Amazon Nova 2](#).
- Pour en savoir plus sur les principales fonctionnalités d'inférence, voir [Inférence de base](#).
- Pour personnaliser les modèles en fonction de votre cas d'utilisation, voir [Personnalisation des modèles Amazon Nova sur l'IA SageMaker](#).

Nouveautés d'Amazon Nova 2

Amazon Nova 2 apporte des améliorations significatives en termes de capacités de compréhension, de création et de reconnaissance vocale. Les sections suivantes décrivent les principales nouvelles fonctionnalités et améliorations.

Nouveaux modèles

Nova 2 Lite

Des modèles améliorés qui offrent un raisonnement avancé avec un support de réflexion étendu, un contrôle d'intensité à trois niveaux et une compréhension multimodale optimisés pour les flux de travail agentiques à l'échelle de la production.

Nova 2 Sonic

Un modèle vocal conversationnel amélioré avec des capacités améliorées de compréhension de la parole, de traitement du langage naturel et de génération vocale dans sept langues.

Intégrations multimodales Nova

Modèle d'intégration multimodal qui traite les entrées de texte, d'image, de document, vidéo et audio et génère des intégrations.

Nouvelles fonctionnalités et capacités

Intégrations multimodales Nova

Nova Multimodal Embeddings prend en charge le texte, les documents, les images, la vidéo et l'audio via un modèle unique, permettant ainsi des applications de récupération intermodales. Nova Multimodal Embeddings mappe chacun de ces types de contenu dans un espace sémantique unifié, ce qui vous permet de réaliser des opérations vectorielles unimodales, intermodales et multimodales, alimentant des applications telles que la génération augmentée par extraction agentique (RAG) et la recherche sémantique multimodale.

Réflexion et raisonnement approfondis

Nova 2 Lite prend en charge la réflexion étendue, ce qui permet aux modèles de passer plus de temps à réfléchir à des problèmes complexes avant de générer des réponses. Cette fonctionnalité

améliore la précision des tâches de raisonnement en plusieurs étapes, notamment les flux de travail agentiques avec plusieurs outils, les mathématiques avancées, la planification complexe et la génération de code.

Pour en savoir plus, accédez à [Réflexion approfondie dans Amazon Nova 2](#).

Outils intégrés

Amazon Nova 2 inclut des outils intégrés qui étendent les capacités du modèle sans nécessiter d'intégrations externes :

- Base sur le Web : accède aux informations en temps réel depuis le Web pour fournir des up-to-date réponses et réduire les hallucinations.
- Interpréteur de code — Exécute le code Python pour effectuer des calculs.

Création d'agents IA

Les modèles Amazon Nova 2 sont optimisés pour créer des agents d'intelligence artificielle. Les modèles permettent une meilleure utilisation des outils, un meilleur raisonnement pour les tâches en plusieurs étapes et une meilleure capacité à maintenir le contexte dans les flux de travail complexes des agents.

Meilleure compréhension des documents

Nova 2 Lite fournit des capacités de traitement de documents améliorées avec une meilleure compréhension des mises en page de documents complexes, des tableaux, des graphiques et des documents de plusieurs pages. Les modèles extraient les informations de manière plus précise à partir de PDFs feuilles de calcul et d'autres formats de documents.

Meilleure compréhension de la vidéo

Nova 2 Lite offre des capacités d'analyse vidéo améliorées, notamment une meilleure perception visuelle, une meilleure compréhension temporelle, une reconnaissance des actions et la capacité de traiter des séquences vidéo plus longues avec une plus grande précision.

Personnalisation de modèles

Nova 2 Lite prend en charge le réglage fin supervisé (SFT) et le réglage fin par renforcement (RFT) sur Amazon Bedrock et SageMaker AI, ce qui vous permet d'adapter Amazon Nova 2 aux besoins spécifiques de votre entreprise.

Amazon Nova Forge

Amazon Nova Forge est un first-of-its-kind service qui offre aux entreprises le moyen le plus simple et le plus rentable de créer leurs propres modèles de pointe à l'aide d'Amazon Nova.

Étapes suivantes

- Pour en savoir plus sur les modèles et les fonctionnalités d'Amazon Nova, consultez [Qu'est-ce qu'Amazon Nova 2 ?](#).
- Pour commencer à utiliser Amazon Nova 2.0, consultez [Commencer à utiliser Amazon Nova 2](#).
- Pour découvrir les principales fonctionnalités d'inférence, voir [Inférence de base](#).

Commencer à utiliser Amazon Nova 2

Vous pouvez invoquer les modèles Amazon Nova 2 à l'aide des méthodes suivantes :

- Opérations d'API Amazon Bedrock [InvokeModel](#) ou [Converse](#)
- [Terrains de jeu](#) Amazon Bedrock pour Nova 2 Lite, Nova 2 Sonic

Important

Si vous utilisez Amazon Bedrock pour la première fois, vous devez d'abord configurer Amazon Bedrock en suivant les étapes décrites dans [Getting started with Amazon Bedrock](#). AWS [Vous pouvez également essayer les modèles Amazon Nova sans accéder à Amazon Bedrock sur nova.amazon.com ou commencer à créer gratuitement sur nova.amazon.com/dev.](#)

Sélectionnez une rubrique pour commencer à utiliser Amazon Nova avec cette méthode.

Premiers pas avec Amazon Nova dans la console Amazon Bedrock

Cette section explique comment utiliser les terrains de jeu de la AWS console pour envoyer un message texte aux modèles Amazon Nova 2 et générer une réponse sous forme de texte ou d'image. Avant d'exécuter les exemples suivants, vous devez vérifier que vous remplissez les conditions préalables suivantes :

Conditions préalables

- Vous avez un AWS compte et êtes autorisé à accéder à un rôle dans ce compte avec les autorisations nécessaires pour Amazon Bedrock. Sinon, veuillez suivre les étapes décrites dans [Premiers pas avec Amazon Bedrock](#).
- Si vous accédez au modèle depuis une région des États-Unis, vous devez utiliser le point de terminaison CRIS américain, ce qui implique d'ajouter le us préfixe à l'identifiant du modèle (tel que `us.amazon.nova-2-lite-v1:0`).
- Si vous accédez au modèle depuis un autre pays que les États-Unis, vous pouvez soit utiliser le point de terminaison CRIS global (tel que `global.amazon.nova-2-lite-v1:0`), soit préfixer la région dans l'ID du modèle (tel que `eu/jp`).

Explorer le terrain de jeu de texte

L'exemple suivant illustre comment utiliser le terrain de jeu de texte :

1. Ouvrez la console Amazon Bedrock à <https://console.aws.amazon.com/bedrock/> l'adresse.
2. Dans le volet de navigation de gauche, choisissez Chat/Texte sous Terrains de jeu.
3. Choisissez Sélectionner le modèle, puis sélectionnez un fournisseur et un modèle. Pour cet exemple, sélectionnez Amazon, puis sélectionnez un modèle Amazon Nova 2 (tel que Nova 2 Lite ou Nova 2 Sonic). Choisissez ensuite Appliquer.
4. Sélectionnez une invite par défaut sous le panneau de texte ou entrez une invite dans le panneau de texte, telle que **Describe the purpose of a "hello world" program in one line.**
5. Pour explorer les capacités de compréhension des images d'Amazon Nova, vous pouvez charger depuis votre ordinateur une image au format JPEG, PNG, GIF ou WEBP dont la taille est inférieure ou égale à 25 Mo. Une fois l'image chargée, vous pouvez interroger Amazon Nova à son sujet.
6. Pour explorer les fonctionnalités d'Amazon Nova en matière de compréhension des documents, vous pouvez télécharger des documents au format CSV, DOC, DOCX, HTML, MD, PDF, TXT, XLS ou XLSX d'une taille inférieure ou égale à 4,5 Mo. Une fois les documents chargés, vous pouvez interroger Amazon Nova à leur sujet.
7. Pour découvrir les fonctionnalités de compréhension vidéo d'Amazon Nova, vous pouvez télécharger une vidéo au MP4 format MKV, MOV ou d'une taille inférieure ou égale à 25 Mo depuis votre ordinateur. Vous pouvez utiliser Amazon S3 pour les vidéos jusqu'à 1 Go. Une fois la vidéo chargée, vous pouvez interroger Amazon Nova à son sujet.
8. Choisissez Exécuter pour exécuter l'inférence sur le modèle. Le texte généré apparaît sous votre invite dans le panneau de texte.

Démarrez avec l'API

Avant de commencer APIs, suivez ces procédures si vous utilisez la AWS CLI AWS ou un AWS SDK pour la première fois ou si vous devez les installer. Si aucune de ces conditions ne s'applique à vous, passez à [Obtention des informations d'identification pour accorder un accès programmatique](#).

Je suis nouveau AWS

Si vous n'avez pas de AWS compte, suivez les étapes ci-dessous pour en créer un. Vous pouvez également commencer à créer gratuitement avec Amazon Nova sur nova.amazon.com/dev.

Pour créer un AWS compte

1. Ouvrez l'<https://portal.aws.amazon.com/billing/inscription>.
2. Suivez les instructions en ligne.

Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique ou un SMS et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

Lorsque vous créez un AWS compte, un AWS utilisateur root est créé. L'utilisateur root a accès à tous les services et ressources AWS du compte. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et utilisez uniquement l'utilisateur root pour effectuer [les tâches nécessitant un accès utilisateur root](#).

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à <https://aws.amazon.com/> et en choisissant Mon compte.

Sécurisez votre AWS compte (utilisateur root)

1. Connectez-vous en [AWS Management Console](#) tant que propriétaire du compte en choisissant Utilisateur root et en saisissant l'adresse e-mail de votre AWS compte. Sur la page suivante, saisissez votre mot de passe.

Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur root, consultez [la section Connexion en tant qu'utilisateur root](#) dans le Guide de l'utilisateur de AWS connexion.

2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

Pour obtenir des instructions, consultez la section [Activer un périphérique MFA virtuel pour l'utilisateur root de votre AWS compte \(console\)](#) dans le guide de l'utilisateur IAM.

J'ai besoin d'installer la AWS CLI ou un AWS SDK

Pour installer la AWS CLI, suivez les étapes décrites dans la [section Installer ou mettre à jour vers la dernière version de la AWS CLI](#).

Pour installer un AWS SDK, sélectionnez l'onglet correspondant au langage de programmation que vous souhaitez utiliser dans [Tools to Build on AWS](#).

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chaque SDK fournit une API, des exemples de code et de la documentation qui permettent aux développeurs de créer plus facilement des applications dans leur langue préférée. SDKs effectuer automatiquement des tâches utiles pour vous, telles que :

- Signature cryptographique des requêtes de service
- Nouvelle tentative de demandes
- Gestion des réponses d'erreur

Obtention des informations d'identification pour accorder un accès programmatique

Les utilisateurs ont besoin d'un accès programmatique s'ils souhaitent interagir AWS en dehors de la console de AWS gestion. La manière d'accorder un accès programmatique dépend du type d'utilisateur qui y accède AWS.

Pour accorder aux utilisateurs un accès programmatique, choisissez l'une des options suivantes.

Quel principal a besoin d'un accès programmatique ?	À	En
Utilisateurs IAM	Limitez la durée des informations d'identification à long terme pour signer les demandes programmatiques adressées à la AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour la AWS CLI, voir Authentification à l'aide des informations d'identification utilisateur IAM dans le Guide de l'utilisateur de l'interface de ligne de commande de AWS. • Pour les outils AWS SDKs et, voir Authentifier à l'aide d'informations

Quel principal a besoin d'un accès programmatique ?	À	En
		d'identification à long terme dans le guide de référence des outils AWS SDKs et. • Pour AWS APIs, voir Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.
Rôles IAM	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées à la AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de la section Utilisation d'informations d'identification temporaires avec AWS des ressources dans le Guide de l'utilisateur IAM.
Utilisateurs gérés dans IAM Identity Center	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées à la AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour la AWS CLI, voir Configuration de la AWS CLI pour utiliser IAM Identity Center dans le Guide de l'utilisateur de l'interface de ligne de AWS commande. • Pour les outils et AWS SDKs AWS APIs, consultez la section Authentification IAM Identity Center dans le guide de référence des outils AWS SDKs et.

Attribution des autorisations Amazon Bedrock à un utilisateur ou à un rôle

Après avoir configuré les informations d'identification pour l'accès par programmation, vous devez configurer les autorisations pour qu'un utilisateur ou un rôle IAM ait accès aux actions liées à Amazon Bedrock. Pour configurer ces autorisations, procédez comme suit :

1. Sur la page d'accueil de la console de AWS gestion, sélectionnez le service IAM ou accédez à la console IAM à l'adresse. <https://console.aws.amazon.com/iam/>
2. Sélectionnez Utilisateurs ou Rôles, puis votre utilisateur ou votre rôle.
3. Dans l'onglet Autorisations, choisissez Ajouter des autorisations, puis sélectionnez Ajouter une politique AWS gérée. Choisissez la stratégie gérée par [AmazonBedrockFullAccess](#) AWS .
4. Pour autoriser l'utilisateur ou le rôle à s'abonner à des modèles, choisissez Créer une politique en ligne, puis spécifiez les autorisations suivantes dans l'éditeur JSON :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MarketplaceBedrock",
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:ViewSubscriptions",
        "aws-marketplace:Unsubscribe",
        "aws-marketplace:Subscribe"
      ],
      "Resource": "*"
    }
  ]
}
```

Générer une réponse à une invite de texte à l'aide d'un modèle Amazon Nova

Une fois que vous avez rempli toutes les conditions requises, vous pouvez tester l'envoi de demandes d'invocation de modèles aux modèles Amazon Nova à l'aide d'une demande [Converse](#).

Pour installer la AWS CLI, suivez les étapes décrites dans la [section Installer ou mettre à jour vers la dernière version de la AWS CLI](#). Vérifiez que vous avez configuré vos informations d'identification

pour utiliser Boto3 en suivant les étapes décrites dans [Obtention des informations d'identification pour accorder un accès programmatique](#).

Pour générer une réponse à une invite de texte dans Nova 2 Lite à l'aide de la AWS CLI, exécutez la commande suivante dans un terminal :

```
aws bedrock-runtime converse \
  --model-id us.amazon.nova-2-lite-v1:0 \
  --messages '[{"role":"user","content":[{"text":"Write a short poem"}]]' \
  --additional-model-request-fields '{"reasoningConfig":
{"type":"enabled","maxReasoningEffort":"low"}}'
```


Inférence de base

L'inférence est le processus qui consiste à envoyer une demande à un modèle Amazon Nova et à recevoir une réponse générée. Les modèles Amazon Nova prennent en charge l'inférence via deux options d'API :

- API Converse ([Converse](#), [ConverseStream](#)) : fournit une interface cohérente entre les différents modèles, ce qui facilite le passage d'un modèle à l'autre ou la création d'applications compatibles avec plusieurs modèles. Recommandé pour la plupart des cas d'utilisation.
- Invoke API ([InvokeModel](#), [InvokeModelWithResponseStream](#)) : les charges utiles des requêtes sont structurées spécifiquement pour le format natif de chaque modèle. Exécute l'inférence à l'aide des paramètres d'invite et d'inférence fournis dans le corps de la demande.

Les deux APIs prennent en charge les mêmes fonctionnalités de base, notamment :

- Conversations complexes
- Entrées multimodales (texte, images, vidéo, audio)
- Utilisation d'outil
- Barrières de protection
- Réponses en streaming

La structure de demande est presque identique entre les deux APIs, ne différant que par la manière dont les données en octets (documents, images, vidéo et audio) sont encodées. Pour les paramètres de demande de modèle propres aux modèles Amazon Nova, tels que `reasoningConfig` et `topK`, ils sont placés dans un `inferenceConfig` objet supplémentaire au sein du `additionalModelRequestFields`. Il s'agit de paramètres de haut niveau pour `InvokeModel` et `InvokeModelWithResponseStream`.

Note

Consultez les exemples de code et le dépannage pour obtenir une liste d'exemples de code relatifs aux modèles Nova 2.

Définissez l'une `modelId` des valeurs suivantes pour utiliser les modèles Amazon Nova :

Modèle	ID du modèle
Nova 2 Lite	• global.amazon.nova-2-lite-v 1:0 • us.amazon.nova-2-lite-v 1:0
Nova 2 Sonic	• global.amazon.nova-2-sonic-v 1:0 • us.amazon.nova-2-sonic-v 1:0
Intégrations multimodales Nova	amazon.nova-2- 1:0 multimodal-embeddings-v

Important : Configuration du délai d'expiration

Important

Le traitement des demandes d'inférence Amazon Nova peut prendre jusqu'à 60 minutes. Configurez les paramètres de délai d'expiration de votre client en conséquence : L'exemple suivant est du code Python. Les utilisateurs peuvent consulter la documentation de leur version linguistique préférée du SDK dans la documentation de l'API de ce SDK.

```
from botocore.config import Config

bedrock = boto3.client(
    'bedrock-runtime',
    region_name='us-east-1',
    config=Config(
        read_timeout=3600 # 60 minutes
    )
)
```

Sujets d'inférence de base

Cette section aborde les sujets suivants :

- Utilisation de l'API converse : une interface cohérente offrant une compatibilité avec la plupart des modèles Bedrock

- Utilisation de l'API d'invocation : une interface unique aux modèles Nova et non portable pour les autres modèles Bedrock
- Réponses en streaming : génération de réponses en temps réel
- Utilisation des intégrations Amazon Nova : fonctionnalités d'intégration de texte
- Inférence à la demande : Pay-per-use modèle d'inférence

Utilisation de l'API Converse

L'API Converse fournit une interface unifiée pour interagir avec les modèles Amazon Nova. Il élimine les détails spécifiques au modèle et fournit une méthode cohérente pour gérer les conversations à plusieurs tours, les instructions du système et les réponses en streaming sur tous les modèles Amazon Nova.

Rubriques

- [Structure de la demande](#)
- [Utilisation des instructions du système](#)
- [Paramètres d'inférence](#)
- [Utiliser le raisonnement](#)

Structure de la demande

- Conversations à plusieurs tours : maintenez le contexte dans plusieurs échanges
- Invites du système : instructions système telles que les personas ou les directives de réponse
- Chat documentaire : interagissez avec des documents ou des collections de documents et interrogez-les
- Vision : traiter et analyser des images et des vidéos
- Utilisation des outils : autorisez les modèles à utiliser des outils externes et APIs
- Garde-corps : appliquez le filtrage du contenu et les contrôles de sécurité
- Raisonnement : réflexion approfondie pour la résolution de problèmes complexes

Une requête d'API Converse de base inclut l'ID du modèle et une liste de messages :

```
import boto3
```

```
bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

response = bedrock.converse(
    modelId='us.amazon.nova-2-lite-v1:0',
    messages=[
        {
            'role': 'user',
            'content': [{'text': 'What is machine learning?'}]
        }
    ]
)

content_list = response["output"]["message"]["content"]
# Extract the first text block
text = next((item["text"] for item in content_list if "text" in item), None)
if text is not None:
    print(text)
```

Utilisation des instructions du système

Les instructions du système fournissent le contexte et les instructions au modèle :

```
import boto3

bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

response = bedrock.converse(
    modelId='us.amazon.nova-2-lite-v1:0',
    system=[
        {'text': 'You are a helpful AI assistant specializing in cloud computing.'}
    ],
    messages=[
        {
            'role': 'user',
            'content': [{'text': 'Explain serverless computing.'}]
        }
    ]
)

# Print the response text
content_list = response["output"]["message"]["content"]
text = next((item["text"] for item in content_list if "text" in item), None)
```

```
if text is not None:
    print(text)
```

Paramètres d'inférence

Contrôlez la sortie du modèle à l'aide de paramètres d'inférence. Les paramètres d'inférence disponibles sont les suivants.

- **MaxTokens** (entier) : nombre maximum de jetons à générer (jusqu'à 65 000). S'il n'est pas spécifié, le modèle utilise une valeur par défaut dynamique basée sur le contexte de la demande.
- **temperature** (float) : contrôle le caractère aléatoire (0,0-1,0, 0,7 par défaut). Des valeurs plus faibles rendent la sortie plus déterministe
- **TopP** (float) : seuil d'échantillonnage du noyau (0-1, 0,9 par défaut). Des valeurs plus faibles rendent la sortie plus ciblée
- **StopSequences** (array) : séquences de caractères qui arrêtent la génération lorsqu'elles sont rencontrées

Exemple :

```
import boto3
bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

response = bedrock.converse(
    modelId='us.amazon.nova-2-lite-v1:0',
    messages=[
        {
            'role': 'user',
            'content': [{'text': 'Write a short story.'}]
        }
    ],
    inferenceConfig={
        'maxTokens': 512,
        'temperature': 0.7,
        'topP': 0.9,
        'stopSequences': ['END']
    }
)

content_list = response["output"]["message"]["content"]
text = next((item["text"] for item in content_list if "text" in item), None)
```

```
if text is not None:
    print(text)
```

Utiliser le raisonnement

Nova 2 Lite permet une réflexion approfondie pour la résolution de problèmes complexes. Permettez de raisonner avec `reasoningConfig`.

Par défaut, le raisonnement est désactivé pour optimiser la rapidité et le coût des requêtes simples. Lorsque vous devez aller au-delà de ces tâches simples, vous pouvez activer le raisonnement. Nova 2 permet de contrôler de manière flexible la profondeur du raisonnement grâce à trois niveaux d'effort :

Faible effort (`maxReasoningEffort: "low"`)

IDÉAL POUR : LES TÂCHES PLUS COMPLEXES NÉCESSITANT UNE RÉFLEXION STRUCTURÉE. Par exemple, vous pouvez l'utiliser pour la révision du code et les suggestions d'amélioration lorsque le modèle doit prendre en compte attentivement la qualité du code existant, pour les tâches d'analyse qui nécessitent une prise en compte approfondie de plusieurs facteurs ou pour les scénarios de résolution de problèmes bénéficiant d'une approche méthodique. Le faible niveau d'effort est idéal pour les tâches complexes où le raisonnement de base améliore la précision sans nécessiter une planification approfondie en plusieurs étapes.

Effort moyen (`maxReasoningEffort: "medium"`)

IDÉAL POUR : LES TÂCHES EN PLUSIEURS ÉTAPES ET LES FLUX DE TRAVAIL DE CODAGE. Par exemple, vous pouvez l'utiliser pour le développement et le débogage de logiciels lorsque le modèle doit comprendre la structure de code existante avant d'implémenter des modifications, pour générer du code nécessitant une coordination entre plusieurs fichiers ou composants, pour des calculs en plusieurs étapes avec interdépendances ou pour planifier des tâches avec plusieurs contraintes. Un effort moyen est optimal pour les flux de travail agentiques qui coordonnent plusieurs outils et nécessitent que le modèle maintienne le contexte de plusieurs opérations séquentielles.

Effort élevé (`maxReasoningEffort: "high"`)

IDÉAL POUR : LE RAISONNEMENT RADICAL ET LA RÉOLUTION AVANCÉE DE PROBLÈMES. Par exemple, vous pouvez l'utiliser pour des problèmes mathématiques avancés et des preuves qui nécessitent une step-by-step vérification rigoureuse, des analyses scientifiques et des tâches de recherche exigeant une investigation approfondie, une conception de systèmes

complexes avec des considérations architecturales à plusieurs dimensions, ou des scénarios de prise de décision critiques ayant des implications importantes. Un effort élevé permet d'obtenir une précision maximale pour les tâches nécessitant un raisonnement sophistiqué, une évaluation minutieuse des alternatives et une validation approfondie des conclusions.

Les exemples suivants montrent différents niveaux d'effort de raisonnement :

Low effort

```
import boto3

bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

response = bedrock.converse(
    modelId='us.amazon.nova-2-lite-v1:0',
    system=[{"text": "You are a highly capable personal assistant"}],
    messages=[{
        "role": "user",
        "content": [{"text": "Provide a meal plan for a gluten free family of 4."}]
    }],
    inferenceConfig={
        "temperature": 0.7,
        "topP": 0.9,
        "maxTokens": 10000
    },
    additionalModelRequestFields={
        "reasoningConfig": {
            "type": "enabled",
            "maxReasoningEffort": "low"
        }
    }
)
```

Paramètres de raisonnement :

Les paramètres de raisonnement sont les suivants :

- type: enabled ou disabled (par défaut :disabled)
- maxReasoningEffort: lowmedium, ouhigh. Cela est nécessaire lorsque le raisonnement est activé.

Note

Temperature, TopP et TopK ne peuvent pas être utilisés lorsque le paramètre est `maxReasoningEffort` réglé sur `high`. Cela provoquera une erreur.

La réponse inclut le contenu du raisonnement :

```
{
  "output": {
    "message": {
      "role": "assistant",
      "content": [
        {
          "reasoningContent": {
            "reasoningText": {
              "text": "[REDACTED]"
            }
          }
        },
        {
          "text": "Based on the premises, we can conclude..."
        }
      ]
    }
  },
  "stopReason": "end_turn"
}
```

Note

Avec Amazon Nova 2, le contenu de raisonnement s'affiche sous la forme `[REDACTED]`. Les jetons de raisonnement vous sont toujours facturés, car ils contribuent à améliorer la qualité de sortie. Nous incluons maintenant ce champ dans la structure de réponse afin de préserver la possibilité d'exposer le contenu du raisonnement à l'avenir. Nous travaillons activement avec nos clients afin de déterminer la meilleure approche pour définir le processus de raisonnement du modèle.

Utilisation de l'API Invoke

L'API Invoke fournit un accès direct aux modèles Amazon Nova et permet de mieux contrôler le format des demandes et des réponses. Contrairement à l'API Converse qui extrait les détails spécifiques au modèle, l'API Invoke vous permet de travailler directement avec les structures de demande et de réponse natives du modèle.

Note

L'API Invoke prend en charge les mêmes fonctionnalités que l'API Converse, à l'exception de la modalité de saisie de documents, qui est spécifique à l'API Converse.

Structure de la demande

Une demande d'API Invoke nécessite l'ID du modèle et un corps de requête JSON :

```
import boto3
import json

bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

request_body = {
    'messages': [
        {
            'role': 'user',
            'content': [{'text': 'What is machine learning?'}]
        }
    ],
    'inferenceConfig': {
        'maxTokens': 512,
        'temperature': 0.7
    }
}

response = bedrock.invoke_model(
    modelId='us.amazon.nova-2-lite-v1:0',
    body=json.dumps(request_body)
)

response_body = json.loads(response['body'].read())
```

```
content_list = response_body["output"]["message"]["content"]
# Extract the first text block
text_block = next((item for item in content_list if "text" in item), None)
if text_block is not None:
    print(text_block["text"])
```

Paramètres de demande

L'API Invoke prend en charge les paramètres clés suivants :

- **messages**: ensemble de messages de conversation avec rôle et contenu
- **system**: demande de contexte et d'instructions facultative du système
- **inferenceConfig**: Paramètres contrôlant la sortie du modèle (temperature, MaxTokens, TopP, TopK, StopSequences, ReasoningConfig)
- **toolConfig**: Spécifications de l'outil et choix de l'outil pour l'appel de fonctions

Réponses en streaming

Le streaming vous permet de recevoir les réponses des modèles de manière incrémentielle au fur et à mesure qu'elles sont générées, offrant ainsi une expérience utilisateur plus interactive. L'API Converse et l'API Invoke prennent toutes deux en charge le streaming.

Streaming avec ConverseStream

ConverseStream à utiliser pour recevoir des réponses sous forme de flux d'événements :

```
import boto3

bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

response = bedrock.converse_stream(
    modelId='us.amazon.nova-2-lite-v1:0',
    messages=[
        {
            'role': 'user',
            'content': [{'text': 'Write a short story about AI.'}]
        }
    ]
)
```

```
for event in response['stream']:
    if 'contentBlockDelta' in event:
        delta = event['contentBlockDelta']['delta']
        if 'text' in delta:
            print(delta['text'], end='', flush=True)
```

Streaming avec InvokeModelWithResponseStream

InvokeModelWithResponseStream À utiliser pour le streaming avec l'API Invoke :

```
import boto3
import json

bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

request_body = {
    'messages': [
        {
            'role': 'user',
            'content': [{'text': 'Explain quantum computing.'}]
        }
    ]
}

response = bedrock.invoke_model_with_response_stream(
    modelId='us.amazon.nova-2-lite-v1:0',
    body=json.dumps(request_body)
)

for event in response['body']:
    chunk = json.loads(event['chunk']['bytes'])
    if 'contentBlockDelta' in chunk:
        delta = chunk['contentBlockDelta']['delta']
        if 'text' in delta:
            print(delta['text'], end='', flush=True)
```

Types d'événements en streaming

Les réponses de streaming incluent plusieurs types d'événements :

- `messageStart`: indique le début d'un message
- `contentBlockStart`: indique le début d'un bloc de contenu

- `contentBlockDelta`: contient du texte ou des données incrémentiels
- `contentBlockStop`: indique la fin d'un bloc de contenu
- `messageStop`: indique la fin du message avec la raison de l'arrêt
- `metadata`: contient des informations d'utilisation (nombre de jetons)

Utilisation des intégrations Amazon Nova

Amazon Nova Multimodal Embeddings est un modèle d'intégration multimodale pour les applications AGENTIC RAG et de recherche sémantique. Il prend en charge le texte, les documents, les images, la vidéo et l'audio via un modèle unique, permettant une récupération intermodale. Nova Multimodal Embeddings mappe chacun de ces types de contenu dans un espace sémantique unifié, ce qui vous permet de réaliser des opérations vectorielles unimodales, intermodales et multimodales.

Lorsqu'un contenu est transmis via les vectorisations Nova, le modèle convertit ce contenu en un format numérique universel, appelé vecteur. Un vecteur est un ensemble de valeurs numériques qui peuvent être utilisées pour différentes fonctionnalités de recherche. Les contenus similaires se voient attribuer un vecteur plus proche que les contenus moins similaires.

Applications :

- Récupération de contenu sémantique et recommandation : générez des intégrations pour votre contenu, puis utilisez-les pour rechercher des éléments similaires ou fournir des recommandations personnalisées.
- Recherche multimodale : combinez les intégrations de différents types de contenu pour activer les fonctionnalités de recherche intermodale.
- RAG : générez des vectorisations à partir de contenu multimodal, tel que des documents contenant du texte et des images entrelacés, afin d'optimiser votre flux de travail de recherche pour les applications GenAI.

Fonctions principales

Les principales caractéristiques de Nova Embeddings sont les suivantes :

- Support du texte, de l'image, de l'image du document, de la vidéo et de l'audio dans un espace sémantique unifié. La longueur maximale du contexte est de 8 000 jetons ou 30 s de vidéo et 30 s d'audio.

- Synchrones et asynchrones APIs : l'API prend en charge l'utilisation synchrone et asynchrone.
- Segmentation de fichiers volumineux : l'API asynchrone facilite le traitement de données d'entrée volumineuses en fournissant une segmentation créée à l'API pour les textes, vidéos et fichiers audio longs, contrôlée par des paramètres définis par l'utilisateur. Le modèle générera une seule vectorisation pour chaque segment.
- Vidéo avec audio : traitez simultanément la vidéo et l'audio. Spécifiez si vous souhaitez une intégration unique représentant les deux modalités ou deux intégrations distinctes.
- Objectif de l'intégration : optimisez vos intégrations en fonction de l'application en aval prévue (classification/retrieval/RAG/Search, clustering).
- Tailles des dimensions : 4 tailles de dimensions pour trouver un compromis entre la précision de la vectorisation et le coût de stockage des vecteurs : 3072 ; 1024 ; 384 ; 256.
- Méthodes de saisie : transmettez le contenu à intégrer en spécifiant un URI S3 ou en ligne sous forme de codage base64.

Génération d'intégrations

Procédez comme suit pour générer des intégrations.

Intégration synchrone pour le texte :

```
import boto3
import json

# Create the Bedrock Runtime client.
bedrock_runtime = boto3.client(
    service_name='bedrock-runtime',
    region_name='us-east-1',
)

# Define the request body.
request_body = {
    'taskType': 'SINGLE_EMBEDDING',
    'singleEmbeddingParams': {
        'embeddingPurpose': 'GENERIC_INDEX',
        'embeddingDimension': 3072,
        'text': {'truncationMode': 'END', 'value': 'Hello, World!'},
    },
}
```

```

try:
    # Invoke the Nova Embeddings model.
    response = bedrock_runtime.invoke_model(
        body=json.dumps(request_body, indent=2),
        modelId='amazon.nova-2-multimodal-embeddings-v1:0',
        accept='application/json',
        contentType='application/json',
    )

except Exception as e:
    # Add your own exception handling here.
    print(e)

# Print the request ID.
print('Request ID:', response.get('ResponseMetadata').get('RequestId'))

# Print the response body.
response_body = json.loads(response.get('body').read())
print(json.dumps(response_body, indent=2))

```

essayez :

```

response = bedrock_runtime.invoke_model(
#####body=json.dumps(request_body, indent=2),
#####modelId='amazon.nova-2-multimodal-embeddings-v1:0',
#####accept='application/json',
#####contentType='application/json',
####)
except Exception as e:
####print(e)

print('Request ID:', response.get('ResponseMetadata').get('RequestId'))
response_body = json.loads(response.get('body').read())
print(json.dumps(response_body, indent=2))

```

Intégration asynchrone pour les vidéos :

```

import boto3

bedrock_runtime = boto3.client(
####service_name='bedrock-runtime',
####region_name='us-east-1',
)

```

```

model_input = {
    ####'taskType': 'SEGMENTED_EMBEDDING',
    ####'segmentedEmbeddingParams': {
    #####'embeddingPurpose': 'GENERIC_INDEX',
    #####'embeddingDimension': 3072,
    #####'video': {
    #####'format': 'mp4',
    #####'embeddingMode': 'AUDIO_VIDEO_COMBINED',
    #####'source': {
    #####'s3Location': {'uri': 's3://my-bucket/path/to/video.mp4'}
    #####},
    #####'segmentationConfig': {
    #####'durationSeconds': 15### Segment into 15 second chunks
    #####},
    #####},
    ####},
}

```

essayez :

```

response = bedrock_runtime.start_async_invoke(
    #####modelId='amazon.nova-2-multimodal-embeddings-v1:0',
    #####modelInput=model_input,
    #####outputDataConfig={
    #####'s3OutputDataConfig': {
    #####'s3Uri': 's3://my-bucket'
    #####}
    #####},
    ####)
except Exception as e:
    ####print(e)

print('Request ID:', response.get('ResponseMetadata').get('RequestId'))
print('Invocation ARN:', response.get('invocationArn'))

```

Inférence par lots

L'inférence par lots vous permet de soumettre plusieurs demandes et de générer des intégrations de manière asynchrone. L'inférence par lots vous permet de traiter efficacement de nombreuses demandes en envoyant une seule demande et en générant des réponses dans un compartiment Amazon S3.

- Après avoir défini les entrées du modèle dans les fichiers que vous créez, vous chargez les fichiers dans un compartiment S3.
- Vous soumettez ensuite une demande d'inférence par lots et spécifiez le compartiment S3.
- Une fois la tâche terminée, vous pouvez extraire les fichiers de sortie depuis S3.
- Vous pouvez utiliser l'inférence par lots pour améliorer les performances de l'inférence des modèles sur les jeux de données volumineux.

Formatez et téléchargez vos données d'inférence par lots

Vous devez ajouter vos données d'inférence par lots à un emplacement S3 que vous choisirez ou spécifierez lors de la soumission d'un modèle de tâche d'invocation. L'emplacement S3 doit contenir les éléments suivants :

Au moins un fichier JSONL qui définit les entrées du modèle. Un fichier JSONL contient des lignes d'objets JSON. Votre format JSONL doit se terminer par l'extension `.jsonl` et être au format suivant :

```
{
  "recordId": "record001",
  "modelInput": {
    "taskType": "SINGLE_EMBEDDING",
    "singleEmbeddingParams": {
      "embeddingPurpose": "GENERIC_INDEX",
      "embeddingDimension": 3072,
      "text": {
        "source": {
          "s3Location": {
            "uri": "s3://batch-inference-input-bucket/text_001.txt",
            "bucketOwner": "111122223333"
          }
        },
        "truncationMode": "END"
      }
    }
  }
}
```

Types de fichiers d'entrée pris en charge

- Formats d'image : PNG, JPEG, WEBP, GIF
- Formats audio : WAV MP3, OGG

- Formats vidéo : MOV MP4, MKV, WEBM, FLV, MPEG, MPG, WMV, 3GP

Inférence à la demande

L'inférence à la demande fournit un accès sans serveur aux modèles Amazon Nova sans nécessiter de capacité provisionnée. Ce mode s'adapte automatiquement à votre charge de travail et à vos frais en fonction de l'utilisation.

Avantages

L'inférence à la demande présente plusieurs avantages :

- Aucune planification des capacités : s'adapte automatiquement à la demande
- Paiement à l'utilisation : facturé uniquement pour les jetons traités
- Disponibilité instantanée : aucun temps de mise en service ou de préchauffage requis
- Rentable : idéal pour les charges de travail variables ou imprévisibles

Utilisation de l'inférence à la demande

L'inférence à la demande est le mode par défaut pour les modèles Amazon Nova. Spécifiez simplement l'ID du modèle lorsque vous effectuez des appels d'API :

```
import boto3

bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

response = bedrock.converse(
    modelId='us.amazon.nova-2-lite-v1:0',
    messages=[
        {
            'role': 'user',
            'content': [{'text': 'Hello, Nova!'}]
        }
    ]
)

# Print the response text
content_list = response["output"]["message"]["content"]
text = next((item["text"] for item in content_list if "text" in item), None)
```

```
if text is not None:  
    print(text)
```

Tarification

L'inférence à la demande est facturée en fonction du nombre de jetons d'entrée et de sortie traités. Pour en savoir plus sur les tarifs actuels, consultez les [tarifs d'Amazon Bedrock](#).

Quotas et limites

L'inférence à la demande comporte des quotas par défaut qui varient en fonction du modèle et de la région. Pour demander des augmentations de quotas, utilisez la [console Service Quotas](#).

Utilisation des fonctionnalités d'Amazon Nova 2

Ce guide fournit des recettes basées sur les tâches et des exemples pratiques pour vous aider à intégrer rapidement et efficacement les fonctionnalités de Nova dans vos applications. Il est destiné aux développeurs et aux ingénieurs qui souhaitent :

- Intégrez les modèles Nova dans leurs applications via l'API
- Comprendre les exigences de prétraitement et les stratégies d'optimisation
- Mettre en œuvre des workflows de compréhension multimodaux

Chaque section fournit :

- Informations techniques clés sur les formats pris en charge, les limites de taille et l'estimation des jetons
- Exemples de code illustrant comment utiliser l'API avec le téléchargement direct et Amazon S3
- Cas d'utilisation illustrant des applications pratiques
- Limites à prendre en compte lors de la conception de votre application

Conditions préalables

Avant de commencer, assurez-vous de disposer des éléments suivants :

- Et Compte AWS avec accès à Amazon Bedrock
- Autorisations IAM appropriées pour invoquer les modèles Bedrock
- AWS SDK installé (exemples de Boto3 pour Python)
- Connaissance de base des AWS services, tels qu'Amazon S3 (pour la gestion de fichiers volumineux)

Capacités de raisonnement

Amazon Nova 2 Lite prend en charge la réflexion étendue, qui est désactivée par défaut. Lorsqu'il est activé, le modèle génère des jetons de raisonnement internes qui améliorent la qualité des réponses. Dans Amazon Nova 2 Lite, le contenu du raisonnement est expurgé dans la sortie et s'affiche comme [REDACTED] si ces jetons vous étaient toujours facturés.

Le champ de raisonnement est inclus dans la structure de réponse afin de préserver la possibilité d'exposer ce contenu dans les futures versions. Pour plus d'informations sur la pensée étendue, voir [the section called “Pensée étendue”](#) et [the section called “Bibliothèque de codes”](#).

Compréhension multimodale

Amazon Nova 2 Lite peut comprendre plusieurs modalités de saisie. Ce modèle est doté de fonctionnalités de vision qui lui permettent de comprendre et d'analyser des images, des documents, des vidéos et des discours afin de déduire des questions et d'y répondre en fonction du contenu fourni.

Cette section décrit les directives relatives à l'utilisation d'images, de documents et de vidéos dans Amazon Nova, notamment les stratégies de prétraitement utilisées, les exemples de code et les limites pertinentes à prendre en compte.

Types de contenu pris en charge par modalité

Les informations suivantes détaillent les formats de fichier pris en charge par chaque type de fichier multimédia et la méthode de saisie acceptée.

Type de fichier multimédia	Formats de fichiers pris en charge	Méthode de saisie	Limites de taille	Nombre d'objets
Image	PNG, JPEG, GIF, WebP	Intégrer des données dans la demande	25 Mo	5
	Remarque : Si vous utilisez un fichier GIF ou WebP animé, seule la première image sera utilisée.	Si vous utilisez l'API Converse, codez les données sous forme d'octets. Si vous utilisez l'API Invoke, codez les données sous forme de chaîne Base64.	URI Amazon S3	2 Go au total 1 000

Type de fichier multimédia	Formats de fichiers pris en charge	Méthode de saisie	Limites de taille	Nombre d'objets
Vidéo	MP4, MOV, MKV, WebM, FLV, MPEG, MPG, WMV, 3GP	Intégrer des données dans la demande	25 Mo	1
		Si vous utilisez l'API Converse, codez les données sous forme d'octets. Si vous utilisez l'API Invoke, codez les données sous forme de chaîne Base64.		
		URI Amazon S3	1 Go	1

Rubriques

- [Compréhension des images](#)
- [Compréhension des vidéos](#)
- [Compréhension des documents](#)

Compréhension des images

La compréhension des images fait référence à la capacité d'Amazon Nova à traiter une image et à effectuer diverses tâches de vision par ordinateur, telles que :

- Réalisation de la détection d'objets
- Répondre aux questions sur les images grâce à la fonction de réponse visuelle aux questions (VQA)
- Classification et synthèse des images
- Réalisation de la détection des boîtes de délimitation
- Reconnaissance optique de caractères (OCR)
- Comptage d'objets

Les images peuvent être incluses sous forme d'invite transmise à l'API sous forme de tableaux d'octets ou via l'URI S3.

Informations techniques clés

Voici les principales informations techniques à prendre en compte lorsque vous utilisez cette fonctionnalité.

Dimensionnement et redimensionnement de l'image

Amazon Nova redimensionne automatiquement les images pour optimiser la qualité et les performances :

- Détermine le rapport hauteur/largeur le plus proche (tel que 1:1, 1:2, 2:3, etc.)
- Redimensionne de manière à ce qu'un côté ≥ 896 pixels ou corresponde au côté le plus court de l'image d'origine, selon le plus grand des deux
- Maintient le rapport hauteur/largeur
- Supporte une résolution allant jusqu'à $8\,000 \times 8\,000$ pixels

Coordonnées du cadre de délimitation :

- Utile pour des tâches telles que l'identification d'éléments dans les captures d'écran ou la mise à la base d'images
- Les coordonnées peuvent être redimensionnées pour correspondre aux dimensions d'origine de l'image lors du post-traitement
- Renvoie les cases de délimitation sur une échelle de $[0, 1000]$.

Estimation du jeton d'image

Amazon Nova convertit chaque image en jetons à des fins de traitement. Le nombre de jetons dépend de la résolution et du rapport hauteur/largeur de l'image.

Voici des exemples de nombres approximatifs de jetons basés sur la résolution de l'image :

Résolution de l'image	Tokens estimés
900 x 450	515

Résolution de l'image	Tokens estimés
900 x 900	~1 035
1400 x 900	~1 600
1 800 x 900	~2 060
1300 x 1300	~2 155

Exemples de compréhension d'images

Pour savoir comment intégrer des données d'image directement dans la demande, reportez-vous à l'exemple de saisie multimodale utilisant un actif intégré - API Converse (non diffusé) dans le.

[Bibliothèque de codes](#)

Pour charger des fichiers image volumineux ou plusieurs fichiers image, dont la charge utile globale est supérieure à 25 Mo, utilisez Amazon S3. Pour un exemple complet de l'utilisation des références d'URI Amazon S3 pour la saisie d'images, reportez-vous à l'exemple de saisie multimodale à l'aide de l'URI S3 - API Converse (non diffusé) dans le. [Bibliothèque de codes](#)

Note

Lorsque vous utilisez S3, assurez-vous que le service Amazon Bedrock est autorisé à accéder au bucket et à l'objet.

Principales limites

La liste suivante décrit les limites actuelles des modèles de compréhension des images :

- Compréhension multilingue des images : les modèles ont une compréhension limitée des images et des images vidéo multilingues et peuvent avoir des difficultés ou des hallucinations lors de tâches simples.
- Identification des personnes : les modèles Amazon Nova 2 ne permettent pas d'identifier ou de nommer des personnes sur des images, des documents ou des vidéos.

- **Raisonnement spatial** : les modèles Amazon Nova 2 ont des capacités de raisonnement spatial limitées. Ils peuvent rencontrer des difficultés avec les tâches qui nécessitent une localisation précise ou une analyse de la disposition.
- **Petit texte dans les images et les vidéos** : si le texte de l'image ou de la vidéo est trop petit, envisagez d'augmenter la taille relative du texte de l'image en le recadrant dans la section appropriée tout en préservant le contexte nécessaire.

Compréhension des vidéos

La compréhension vidéo fait référence à la capacité d'Amazon Nova à traiter les entrées vidéo et à effectuer diverses tâches de compréhension vidéo, telles que :

- Analyse des images clés et synthèse du contenu vidéo
- Répondre aux questions concernant les segments vidéo (réponse aux questions vidéo ou assurance qualité vidéo)
- Détection et suivi d'objets sur plusieurs images
- Identification des actions, des scènes et des événements
- Réalisation d'une segmentation temporelle pour localiser des moments spécifiques
- Génération de légendes descriptives ou de résumés de séquences vidéo

Informations techniques clés

Voici les principales informations techniques à prendre en compte lorsque vous utilisez cette fonctionnalité.

Informations sur la taille de la vidéo

Les fonctionnalités de compréhension vidéo d'Amazon Nova prennent en charge le format multiformat. Toutes les vidéos sont redimensionnées avec distorsion (vers le haut ou vers le bas, en fonction du rapport hauteur/largeur d'origine) à 672 × 672 carrés avant d'être saisies dans le modèle.

Le modèle utilise une stratégie d'échantillonnage dynamique basée sur la durée de la vidéo. Pour les vidéos d'une durée de 16 minutes ou moins, Amazon Nova 2 Lite échantillonne 1 image par seconde (FPS). Pour les vidéos de plus de 16 minutes, le taux d'échantillonnage diminue pour maintenir un échantillonnage constant de 960 images, le taux d'échantillonnage variant en conséquence. Cette approche est conçue pour fournir une compréhension plus précise de la vidéo au niveau de la scène pour les vidéos plus courtes par rapport aux contenus vidéo plus longs.

Nous vous recommandons de limiter la durée de la vidéo à moins d'une heure pour les vidéos en faible intensité et à moins de 16 minutes pour les vidéos en mouvement élevé.

Il ne devrait y avoir aucune différence entre l'analyse d'une version 4k d'une vidéo et celle d'une version Full HD. De même, comme le taux d'échantillonnage est de 1 images par seconde, une vidéo à 60 images par seconde devrait fonctionner aussi bien qu'une vidéo à 30 images par seconde. L'utilisation d'une résolution et d'un nombre d'images par seconde supérieurs à ce qui est requis n'est pas avantageuse en raison de la limite de 1 Go de taille vidéo. Cela limitera la longueur de la vidéo correspondant à cette limite de taille. Vous souhaitez peut-être prétraiter des vidéos de plus de 1 Go.

Jetons vidéo

La durée de la vidéo est le principal facteur influant sur le nombre de jetons générés. Pour calculer le coût approximatif, multipliez le nombre estimé de jetons vidéo par le prix par jeton pour le modèle spécifique utilisé.

Le tableau suivant fournit quelques approximations de l'échantillonnage d'images et de l'utilisation des jetons par durée de vidéo pour Amazon Nova 2 Lite :

Durée de la vidéo	Cadres à échantillonner	Fréquence d'échantillonnage par seconde	Jetons approximatifs
10 secondes	10	1	2 880
30 secondes	30	1	8 640
16 minutes	960	1	276 480
20 minutes	1200	1	345 600
30 minutes	1800	1	518 400
45 minutes	2700	1	777 600

Exemples de compréhension de vidéos

Pour voir comment intégrer des données vidéo directement dans la demande, reportez-vous à l'exemple d'entrée multimodale utilisant un actif intégré - API Converse (non diffusé) dans le.

[Bibliothèque de codes](#)

Pour un exemple d'utilisation des références d'URI S3 dans l'entrée vidéo, reportez-vous à l'exemple d'entrée multimodale utilisant l'URI S3 - API Converse (non diffusé) dans le [Bibliothèque de codes](#)

Principales limites

Voici les principales limites du modèle, pour lesquelles la précision et les performances du modèle peuvent ne pas être garanties :

- **Aucun support audio** : les modèles Amazon Nova sont actuellement formés pour traiter et comprendre le contenu vidéo uniquement sur la base des images visuelles. Les pistes audio des vidéos ne sont ni traitées ni analysées.
- **Compréhension multilingue des images** : les modèles Amazon Nova ont une compréhension limitée des images et des images vidéo multilingues. Ils peuvent avoir des difficultés ou avoir des hallucinations lors de tâches simples.
- **Identification des personnes** : les modèles Amazon Nova ne prennent pas en charge la capacité d'identifier ou de nommer des personnes dans des images, des documents ou des vidéos. Les modèles ne fourniront pas les noms ou les identités des personnes dans le contenu visuel.
- **Petit texte dans les vidéos** : si le texte de l'image ou de la vidéo est trop petit, pensez à augmenter la taille relative du texte de la vidéo.
- **Raisonnement spatial** : les modèles Amazon Nova 2 ont des capacités de raisonnement spatial limitées. Ils peuvent avoir du mal à effectuer des tâches qui nécessitent une compréhension précise de la position des objets, des distances ou des relations spatiales dans les vidéos.
- **Contenu inapproprié** : les modèles Amazon Nova ne traitent pas les images inappropriées ou explicites qui enfreignent la politique d'utilisation acceptable.
- **Applications médicales** : en raison de la nature sensible de ces artefacts, même si les modèles Amazon Nova peuvent fournir une analyse générale de certaines images ou vidéos de santé, nous ne recommandons pas leur utilisation pour interpréter des images médicales sensibles telles que des scans diagnostiques complexes. La réponse des modèles Amazon Nova ne doit jamais être considérée comme un substitut à un avis médical professionnel.

Compréhension des documents

La fonctionnalité de compréhension des documents d'Amazon Nova vous permet d'inclure des documents PDFs complets (fichiers Word, feuilles de calcul, etc.) dans le cadre de votre demande. Cela permet au modèle d'analyser, de résumer, d'extraire des informations ou de répondre à des questions concernant le contenu du document.

Amazon Nova 2 Lite peut interpréter à la fois le texte et les éléments visuels (tels que des graphiques ou des tableaux) contenus dans ces documents. Cela permet des cas d'utilisation tels que la réponse à des questions, la synthèse et l'analyse de longs rapports ou de documents numérisés.

Les principales fonctionnalités de compréhension des documents incluent une très grande fenêtre contextuelle (1 million de jetons) pour les documents longs et la possibilité de gérer plusieurs documents en une seule requête.

Modalités et formats de documents pris en charge

Amazon Nova fait la distinction entre deux types de saisie de documents :

- Les documents textuels, tels que les fichiers TXT, CSV, HTML, Markdown ou DOC, sont traités principalement pour leur contenu textuel. Amazon Nova comprend le texte de ces documents et en extrait les informations.
- Les documents multimédias, tels que les fichiers PDF ou DOCX, peuvent contenir des mises en page complexes, des images, des graphiques ou des graphiques intégrés. Pour les documents multimédias, Amazon Nova utilise une compréhension basée sur la vision pour interpréter le contenu visuel, tel que les graphiques, les tableaux, les diagrammes ou les captures d'écran, parallèlement au texte du document.

Les formats de fichiers pris en charge incluent les types de documents courants tels que :

- Fichiers texte brut et texte structuré : CSV, TXT
- Feuilles de calcul : XLS, XLSX, HTML, Markdown
- Formats d'image standard (pour les images contenues dans des documents) : PNG, JPG, GIF, WebP
- Formats de document : DOC, DOCX, PDF
- PDFs qui contiennent des encodages d'image, tels que CMYK ou SVG, ne sont pas pris en charge.

Limites de taille des documents et directives d'utilisation

Contrainte	Limite
Nombre maximum de documents	Jusqu'à 5 documents par demande (s'applique à la fois au téléchargement direct et à Amazon S3)
Taille de document basée sur le texte	Chaque document texte doit être égal ou inférieur à 4,5 Mo
Taille du document basée sur le support	Pour les fichiers PDF et DOCX, il n'existe aucune limite de taille de fichier individuelle. Lorsque vous utilisez le téléchargement direct, la taille combinée de tous les documents multimédia doit être inférieure ou égale à 25 Mo. Lorsque vous utilisez Amazon S3, la taille combinée de tous les documents multimédia doit être inférieure ou égale à 2 Go.
Contenu PDF non pris en charge	PDFs qui contiennent des profils de couleurs CMJN ou des images SVG ne sont pas pris en charge.

Tarification

Amazon Nova utilise une tarification basée sur les jetons : vous payez pour les jetons d'entrée (tout ce que vous envoyez, y compris les documents joints) et les jetons de sortie (la réponse du modèle).

Estimation des jetons pour PDFs : Pour la planification, supposons une page PDF standard de 8,5 x 11 pouces $\approx$ 2 560 jetons d'entrée (cette estimation couvre à la fois le texte et les éléments visuels d'une page type).

Exemples : utilisation de la compréhension des documents de Nova via l'API et S3

Pour un exemple de son utilisation via une API, reportez-vous à l'exemple de saisie multimodale utilisant un actif intégré - API Converse (sans streaming) dans le [Bibliothèque de codes](#)

Pour un exemple de son utilisation via S3, reportez-vous à l'exemple d'entrée multimodale à l'aide de l'URI S3 - API Converse (sans streaming) dans le [Bibliothèque de codes](#)

Speech-to-Speech (Amazon Nova 2 Sonic)

Amazon Nova 2 Sonic permet une intelligence artificielle conversationnelle en temps réel avec entrée et sortie vocales. La section suivante couvre les fonctionnalités avancées de création d'assistants vocaux interactifs, d'automatisation du service client et d'applications conversationnelles.

Fonctions principales

Amazon Nova 2 Sonic fournit les fonctionnalités suivantes :

- State-of-the-art compréhension vocale en continu grâce à une API de diffusion bidirectionnelle qui permet des conversations multitours en temps réel et à faible latence.
- Support multilingue avec détection et changement automatiques de langue. Des voix expressives sont proposées, y compris des voix masculines et féminines, dans les langues suivantes :
 - Anglais (États-Unis, Royaume-Uni, Inde, Australie)
 - Français
 - Italien
 - Allemand
 - Espagnol
 - Portugais
 - Hindi
- Des voix polyglottes pouvant parler n'importe laquelle des langues prises en charge afin de garantir une expérience utilisateur cohérente même lorsque l'utilisateur change de langue au cours de la même session.
- Robustesse au bruit de fond pour les scénarios de déploiement réels.
- Robustesse aux différents accents pour les langues prises en charge.
- Des expériences d'intelligence artificielle conversationnelle naturelles, semblables à celles de l'homme, avec une richesse contextuelle dans toutes les langues prises en charge.
- Réponse vocale adaptative qui ajuste dynamiquement la restitution en fonction de la prosodie de la parole entrée.
- Prise de tour intelligente qui détecte le moment où l'utilisateur a fini de parler et le moment où l'assistant doit répondre, créant ainsi un rythme de dialogue naturel.
- Gestion élégante des interruptions de l'utilisateur sans perte du contexte conversationnel.

- Ancrage des connaissances dans les données d'entreprise à l'aide de la génération à enrichissement contextuel (RAG).
- Appel de fonctions et prise en charge du flux de travail agentique pour créer des applications IA complexes.
- Gestion asynchrone des outils qui exécute les appels aux outils tout en maintenant le flux de conversation, ce qui permet à l'assistant de continuer à parler pendant que les outils traitent en arrière-plan.
- Prise en charge de la saisie intermodale pour les entrées audio et textuelles au sein d'une même conversation, permettant des modèles d'interaction flexibles.
- Limite de connexion de 8 minutes, avec un modèle de renouvellement de connexion et de poursuite de session disponible dans des exemples de code.

Commencer avec speech-to-speech

Les sections suivantes fournissent un exemple et step-by-step expliquent comment implémenter une application de streaming audio simple en temps réel à l'aide d'Amazon Nova 2 Sonic. Cette version simplifiée présente les fonctionnalités de base nécessaires pour créer une conversation audio avec le modèle Amazon Nova 2 Sonic.

Vous pouvez accéder à l'exemple suivant dans notre [GitHubdépôt d'échantillons Nova](#).

La limite de connexion est de 8 minutes, le renouvellement de la connexion et le schéma de poursuite de conversation étant disponibles sur [GitHub](#).

Définir les importations et la configuration

Cette section importe les bibliothèques nécessaires et définit les paramètres de configuration audio :

- `asyncio` : pour la programmation asynchrone
- `base64` : pour l'encodage et le décodage des données audio
- `pyaudio` : pour la capture et la lecture audio
- Composants du kit SDK Amazon Bedrock pour le streaming
- Les constantes audio définissent le format de capture audio (fréquence d'échantillonnage de 16 kHz, canal mono)

```
import os
```

```
import asyncio
import base64
import json
import uuid
import pyaudio
from aws_sdk_bedrock_runtime.client import BedrockRuntimeClient,
    InvokeModelWithBidirectionalStreamOperationInput
from aws_sdk_bedrock_runtime.models import
    InvokeModelWithBidirectionalStreamInputChunk, BidirectionalInputPayloadPart
from aws_sdk_bedrock_runtime.config import Config, HTTPAuthSchemeResolver,
    SigV4AuthScheme
from smithy_aws_core.identity import EnvironmentCredentialsResolver

# Audio configuration
INPUT_SAMPLE_RATE = 16000
OUTPUT_SAMPLE_RATE = 24000
CHANNELS = 1
FORMAT = pyaudio.paInt16
CHUNK_SIZE = 1024
```

Définir la classe **SimpleNovaSonic**

La classe `SimpleNovaSonic` est la classe principale qui gère l'interaction avec Amazon Nova Sonic :

- `model_id` : l'ID du modèle Amazon Nova Sonic (`amazon.nova-2-sonic-v1:0`)
- `region`: AWS Région, la valeur par défaut est `us-east-1`
- Unique IDs pour le suivi rapide et le suivi du contenu
- Une file d'attente asynchrone pour la lecture audio

```
class SimpleNovaSonic:
    def __init__(self, model_id='amazon.nova-2-sonic-v1:0', region='us-east-1'):
        self.model_id = model_id
        self.region = region
        self.client = None
        self.stream = None
        self.response = None
        self.is_active = False
        self.prompt_name = str(uuid.uuid4())
        self.content_name = str(uuid.uuid4())
        self.audio_content_name = str(uuid.uuid4())
```



```
self.audio_queue = asyncio.Queue()
self.display_assistant_text = False
```

Initialiser le client

Cette méthode configure le client Amazon Bedrock avec les éléments suivants :

- Le point de terminaison approprié pour la région spécifiée
- Informations d'authentification utilisant des variables d'environnement pour les informations d'identification AWS
- Le schéma d'authentification SigV4 pour les appels AWS d'API

```
def _initialize_client(self):
    """Initialize the Bedrock client."""
    config = Config(
        endpoint_uri=f"https://bedrock-runtime.{self.region}.amazonaws.com",
        region=self.region,
        aws_credentials_identity_resolver=EnvironmentCredentialsResolver(),
        auth_scheme_resolver=HTTPAuthSchemeResolver(),
        auth_schemes={"aws.auth#sigv4": SigV4AuthScheme(service="bedrock")}
    )
    self.client = BedrockRuntimeClient(config=config)
```

Gérer les événements

Cette méthode d'aide envoie des événements JSON au flux bidirectionnel, qui est utilisé pour toutes les communications avec le modèle Amazon Nova Sonic :

```
async def send_event(self, event_json):
    """Send an event to the stream."""
    event = InvokeModelWithBidirectionalStreamInputChunk(
        value=BidirectionalInputPayloadPart(bytes_=event_json.encode('utf-8'))
    )
    await self.stream.input_stream.send(event)
```

Démarrer la séance

Cette méthode lance la session et configure les événements restants pour démarrer la diffusion audio. Ces événements doivent être envoyés dans le même ordre.

```
async def start_session(self):
    """Start a new session with Nova Sonic."""
    if not self.client:
        self._initialize_client()

    # Initialize the stream
    self.stream = await self.client.invoke_model_with_bidirectional_stream(
        InvokeModelWithBidirectionalStreamOperationInput(model_id=self.model_id)
    )
    self.is_active = True

    # Send session start event
    session_start = '''
    {
        "event": {
            "sessionStart": {
                "inferenceConfiguration": {
                    "maxTokens": 1024,
                    "topP": 0.9,
                    "temperature": 0.7
                },
                "turnDetectionConfiguration": {
                    "endpointingSensitivity": "HIGH"
                }
            }
        }
    }
    '''
    await self.send_event(session_start)

    # Send prompt start event
    prompt_start = f'''
    {{
        "event": {{
            "promptStart": {{
                "promptName": "{self.prompt_name}",
                "textOutputConfiguration": {{
                    "mediaType": "text/plain"
                }},
            }},
            "audioOutputConfiguration": {{
                "mediaType": "audio/lpcm",
                "sampleRateHertz": 24000,
                "sampleSizeBits": 16,
```

```

        "channelCount": 1,
        "voiceId": "matthew",
        "encoding": "base64",
        "audioType": "SPEECH"
    }}
}}
}}
...
await self.send_event(prompt_start)

# Send system prompt
text_content_start = f'''
{{
    "event": {{
        "contentStart": {{
            "promptName": "{self.prompt_name}",
            "contentName": "{self.content_name}",
            "type": "TEXT",
            "interactive": true,
            "role": "SYSTEM",
            "textInputConfiguration": {{
                "mediaType": "text/plain"
            }}
        }}
    }}
}}
...
await self.send_event(text_content_start)

system_prompt = "You are a friendly assistant. The user and you will engage in
a spoken dialog " \
    "exchanging the transcripts of a natural real-time conversation. Keep your
responses short, " \
    "generally two or three sentences for chatty scenarios."

text_input = f'''
{{
    "event": {{
        "textInput": {{
            "promptName": "{self.prompt_name}",
            "contentName": "{self.content_name}",
            "content": "{system_prompt}"

```

```

        }}
    }}
}}
'''
await self.send_event(text_input)

text_content_end = f'''
{{
    "event": {{
        "contentEnd": {{
            "promptName": "{self.prompt_name}",
            "contentName": "{self.content_name}"
        }}
    }}
}}
'''
await self.send_event(text_content_end)

# Start processing responses
self.response = asyncio.create_task(self._process_responses())

```

Gérer l'entrée audio

Ces méthodes gèrent le cycle de vie de l'entrée audio :

- `start_audio_input` : configure et démarre le flux d'entrée audio
- `send_audio_chunk` : encode et envoie des segments audio au modèle
- `end_audio_input` : ferme correctement le flux d'entrée audio

```

async def start_audio_input(self):
    """Start audio input stream."""
    audio_content_start = f'''
    {{
        "event": {{
            "contentStart": {{
                "promptName": "{self.prompt_name}",
                "contentName": "{self.audio_content_name}",
                "type": "AUDIO",
                "interactive": true,
                "role": "USER",
                "audioInputConfiguration": {{

```

```

        "mediaType": "audio/lpcm",
        "sampleRateHertz": 16000,
        "sampleSizeBits": 16,
        "channelCount": 1,
        "audioType": "SPEECH",
        "encoding": "base64"
    }}
    }}
    }}
    ...
    await self.send_event(audio_content_start)

async def send_audio_chunk(self, audio_bytes):
    """Send an audio chunk to the stream."""
    if not self.is_active:
        return

    blob = base64.b64encode(audio_bytes)
    audio_event = f'''
    {{
        "event": {{
            "audioInput": {{
                "promptName": "{self.prompt_name}",
                "contentName": "{self.audio_content_name}",
                "content": "{blob.decode('utf-8')}"
            }}
        }}
    }}
    '''
    await self.send_event(audio_event)

async def end_audio_input(self):
    """End audio input stream."""
    audio_content_end = f'''
    {{
        "event": {{
            "contentEnd": {{
                "promptName": "{self.prompt_name}",
                "contentName": "{self.audio_content_name}"
            }}
        }}
    }}
    '''
    ...

```

```
await self.send_event(audio_content_end)
```

Terminer la session

Cette méthode ferme correctement la session en :

- Envoi d'un événement `promptEnd`
- Envoi d'un événement `sessionEnd`
- Fermeture du flux d'entrée

```
async def end_session(self):
    """End the session."""
    if not self.is_active:
        return

    prompt_end = f'''
    {{
      "event": {{
        "promptEnd": {{
          "promptName": "{self.prompt_name}"
        }}
      }}
    }}
    '''
    await self.send_event(prompt_end)

    session_end = '''
    {
      "event": {
        "sessionEnd": {}
      }
    }
    '''
    await self.send_event(session_end)
    # close the stream
    await self.stream.input_stream.close()
```

Gérer les réponses

Cette méthode traite en continu les réponses du modèle et effectue les opérations suivantes :

- Attend la sortie du flux.
- Analyse la réponse JSON.
- Gère la sortie texte en l'imprimant sur la console avec la reconnaissance vocale automatique et la transcription.
- Gère la sortie audio en la décodant et en la mettant en file d'attente pour la lecture.

```

async def _process_responses(self):
    """Process responses from the stream."""
    try:
        while self.is_active:
            output = await self.stream.await_output()
            result = await output[1].receive()

            if result.value and result.value.bytes_:
                response_data = result.value.bytes_.decode('utf-8')
                json_data = json.loads(response_data)

                if 'event' in json_data:
                    # Handle content start event
                    if 'contentStart' in json_data['event']:
                        content_start = json_data['event']['contentStart']
                        # set role
                        self.role = content_start['role']
                        # Check for speculative content
                        if 'additionalModelFields' in content_start:
                            additional_fields =
                                json.loads(content_start['additionalModelFields'])
                            if additional_fields.get('generationStage') ==
                                'SPECULATIVE':
                                self.display_assistant_text = True
                            else:
                                self.display_assistant_text = False

                    # Handle text output event
                    elif 'textOutput' in json_data['event']:
                        text = json_data['event']['textOutput']['content']

                        if (self.role == "ASSISTANT" and
                            self.display_assistant_text):
                            print(f"Assistant: {text}")
                        elif self.role == "USER":

```

```

        print(f"User: {text}")

        # Handle audio output
        elif 'audioOutput' in json_data['event']:
            audio_content = json_data['event']['audioOutput']
['content']

            audio_bytes = base64.b64decode(audio_content)
            await self.audio_queue.put(audio_bytes)
    except Exception as e:
        print(f"Error processing responses: {e}")

```

Lire l'audio

Cette méthode effectue les tâches suivantes :

- Initialise un flux d'entrée PyAudio
- Récupère en continu les données audio de la file d'attente
- Lit l'audio via les haut-parleurs
- Nettoie correctement les ressources une fois terminé

```

async def play_audio(self):
    """Play audio responses."""
    p = pyaudio.PyAudio()
    stream = p.open(
        format=FORMAT,
        channels=CHANNELS,
        rate=OUTPUT_SAMPLE_RATE,
        output=True
    )

```

essayez :

```

        while self.is_active:
            audio_data = await self.audio_queue.get()
            stream.write(audio_data)
    except Exception as e:
        print(f"Error playing audio: {e}")
    finally:

```



```
stream.stop_stream()
stream.close()
p.terminate()
```

Capturer l'audio

Cette méthode effectue les tâches suivantes :

- Initialise un flux PyAudio de sortie
- Démarre la session d'entrée audio
- Capture en continu des segments audio à partir du microphone
- Envoie chaque segment au modèle Amazon Nova Sonic
- Nettoie correctement les ressources une fois terminé

```
async def capture_audio(self):
    """Capture audio from microphone and send to Nova Sonic."""
    p = pyaudio.PyAudio()
    stream = p.open(
        format=FORMAT,
        channels=CHANNELS,
        rate=INPUT_SAMPLE_RATE,
        input=True,
        frames_per_buffer=CHUNK_SIZE
    )

    print("Starting audio capture. Speak into your microphone...")
    print("Press Enter to stop...")

    await self.start_audio_input()
```

essayez :

```
while self.is_active:
    audio_data = stream.read(CHUNK_SIZE, exception_on_overflow=False)
    await self.send_audio_chunk(audio_data)
    await asyncio.sleep(0.01)
except Exception as e:
    print(f"Error capturing audio: {e}")
```

```
finally:
    stream.stop_stream()
    stream.close()
    p.terminate()
    print("Audio capture stopped.")
    await self.end_audio_input()
```

Exécution de la fonction principale

La fonction principale orchestre l'ensemble du processus en effectuant les opérations suivantes :

- Crée un client Amazon Nova 2 Sonic
- Démarre la session
- Crée des tâches de simultanéité pour la lecture et la capture audio
- Attend que l'utilisateur appuie pour **Enter** arrêter
- Termine correctement la session et nettoie les tâches

```
async def main():
    # Create Nova Sonic client
    nova_client = SimpleNovaSonic()

    # Start session
    await nova_client.start_session()

    # Start audio playback task
    playback_task = asyncio.create_task(nova_client.play_audio())

    # Start audio capture task
    capture_task = asyncio.create_task(nova_client.capture_audio())

    # Wait for user to press Enter to stop
    await asyncio.get_event_loop().run_in_executor(None, input)

    # End session
    nova_client.is_active = False

    # First cancel the tasks
    tasks = []
    if not playback_task.done():
        tasks.append(playback_task)
```

```
if not capture_task.done():
    tasks.append(capture_task)
for task in tasks:
    task.cancel()
if tasks:
    await asyncio.gather(*tasks, return_exceptions=True)

# cancel the response task
if nova_client.response and not nova_client.response.done():
    nova_client.response.cancel()

await nova_client.end_session()
print("Session ended")

if __name__ == "__main__":
    # Set AWS credentials if not using environment variables
    # os.environ['AWS_ACCESS_KEY_ID'] = "your-access-key"
    # os.environ['AWS_SECRET_ACCESS_KEY'] = "your-secret-key"
    # os.environ['AWS_DEFAULT_REGION'] = "us-east-1"

    asyncio.run(main())
```

Exemples de code

Ces exemples de code vous aident à démarrer rapidement avec Amazon Nova 2 Sonic. Vous pouvez accéder à la liste complète des exemples [sur la page des GitHub exemples d'Amazon Nova Sonic](#).

Exemples pour débuter

Pour des exemples simples conçus pour vous aider à utiliser Amazon Nova 2 Sonic, reportez-vous aux implémentations suivantes :

- Implémentation de [base d'Amazon Nova 2 Sonic \(Python\)](#) : implémentation de base qui montre comment les événements sont structurés dans l'API de diffusion bidirectionnelle. Cette version ne prend pas en charge la fonctionnalité barge-in (interruption de l'assistant pendant qu'il parle) et n'implémente pas de véritable communication bidirectionnelle.
- [Implémentation complète d'Amazon Nova 2 Sonic \(Python\)](#) : l'implémentation complète avec une véritable communication bidirectionnelle et un support intégré. Cela permet des conversations plus naturelles où les utilisateurs peuvent interrompre l'assistant pendant qu'il parle, comme dans le cas des conversations humaines.

- [Amazon Nova 2 Sonic avec utilisation d'outils \(Python\)](#) : implémentation avancée qui étend les capacités de communication bidirectionnelle avec des exemples d'utilisation d'outils. Cette version montre comment Amazon Nova 2 Sonic peut interagir avec des outils externes et APIs fournir des fonctionnalités améliorées.
- [Nova Sonic avec texte et saisie mixte \(Python\)](#) : exemple d'implémentation pour montrer comment Amazon Nova 2 Sonic peut avoir du texte en entrée.
- [WebSocket Implémentation de Java \(Java\)](#) : cet exemple implémente une application de streaming WebSocket audio bidirectionnelle qui s'intègre à Amazon Nova 2 Sonic pour des speech-to-speech conversations en temps réel à l'aide de Java.
- [Implémentation de NodeJS Websocket \(NodeJS\) : cet exemple implémente WebSocket une application](#) de streaming audio bidirectionnelle qui s'intègre à Amazon Nova 2 Sonic pour des conversations en temps réel à l'aide de NodeJS. speech-to-speech
- [Implémentation de NodeJS Websocket \(C#\)](#) : cet exemple implémente une application de streaming audio WebSocket bidirectionnelle qui s'intègre à Amazon Nova 2 Sonic pour des conversations en temps réel via .NET. speech-to-speech

Cas d'utilisation avancés

Pour des exemples avancés illustrant des cas d'utilisation plus complexes, veuillez vous référer aux implémentations suivantes :

- [Implémentation de la base de connaissances Amazon Bedrock \(NodeJS\)](#) : cet exemple montre comment créer une application conversationnelle intelligente en intégrant Amazon Nova 2 Sonic à la base de connaissances Amazon Bedrock à l'aide de NodeJS.
- [Gestion de l'historique des discussions \(Python\)](#) : cet exemple inclut un système d'enregistrement de l'historique des discussions qui capture et préserve toutes les interactions entre l'utilisateur et Amazon Nova 2 Sonic à l'aide de Python.
- [Annulation de réservation d'hôtel \(NodeJS\)](#) : cet exemple illustre un cas d'utilisation pratique du service client pour Amazon Nova 2 Sonic, mettant en œuvre un système d'annulation de réservation d'hôtel à l'aide de NodeJS.
- [LangChain Intégration à la base de connaissances \(Python\)](#) : cette implémentation montre comment intégrer les speech-to-speech fonctionnalités d'Amazon Nova 2 Sonic à une base de connaissances LangChain optimisée pour des expériences conversationnelles améliorées à l'aide de Python.

- Reprise de [conversation \(NodeJS\)](#) : cet exemple montre comment implémenter des fonctionnalités de reprise de conversation avec Amazon Nova 2 Sonic. En utilisant un scénario d'annulation de réservation d'hôtel comme contexte, l'application montre comment maintenir l'état de la conversation entre les sessions, permettant aux utilisateurs de poursuivre de manière fluide les interactions précédemment interrompues à l'aide de NodeJS.
- [Nova 2 Sonic Speaks First \(NodeJS\)](#) : cet exemple montre comment Amazon Nova 2 Sonic peut lancer des conversations de manière proactive.
- [Continuation de session \(Python\)](#) : cet exemple montre comment activer une durée de conversation illimitée avec Amazon Nova 2 Sonic en implémentant des transitions de session fluides. L'application crée et bascule automatiquement vers de nouvelles sessions en arrière-plan, ce qui permet aux conversations de se poursuivre indéfiniment sans interruption ni perte de contexte.

Atelier pratique

Un atelier pratique est disponible pour vous aider à créer une application de chat vocal à l'aide d'Amazon Nova 2 Sonic avec une interface de streaming bidirectionnelle. Vous pouvez [accéder à l'atelier](#) et trouver les [exemples de code complets](#).

Invites de conversation vocale

Nova 2 introduit les invites vocales, une fonctionnalité d'invite spécialisée conçue pour contrôler le formatage de transcription spécifique à la parole en hindi. Les instructions vocales fonctionnent parallèlement à l'invite de votre système mais ont un objectif distinct :

- Invite du système : contrôle le comportement, la personnalité et le style de réponse de votre assistant
- Demande vocale : contrôle le formatage de la transcription pour le changement de code en hindi (scripts) Latin/Devanagari/mixed

Directives importantes

Les instructions vocales sont préconfigurées et doivent être utilisées exactement comme indiqué. Ils sont conçus pour répondre à des besoins spécifiques de formatage de transcription et ne doivent pas être modifiés ou personnalisés, car les modifications peuvent entraîner un comportement inattendu.

Quand utiliser les invites vocales :

- Vous devez contrôler la sortie du script pour le changement de code en hindi () Latin/Devanagari/mixed

Quand ne pas utiliser les invites vocales :

- Pour les instructions générales ou le comportement de l'assistant (utilisez plutôt l'invite du système)
- Si vous n'utilisez pas le formatage de transcription en hindi
- Si le besoin de formatage spécifique ne s'applique pas à votre cas d'utilisation

Bonne pratique : n'incluez une invite vocale que si vous avez spécifiquement besoin d'un formatage de transcription en hindi. Toutes les autres instructions, y compris les préférences linguistiques, le style de réponse, la verbosité et le raisonnement, doivent figurer dans l'invite de votre système.

Important : les instructions vocales doivent être envoyées au modèle une fois que le système l'a demandé.

Système de référence recommandé : invite vocale

You are a warm, professional, and helpful AI assistant. Give accurate answers that sound natural, direct, and human. Start by answering the user's question clearly in 1-2 sentences. Then, expand only enough to make the answer understandable, staying within 3-5 short sentences total. Avoid sounding like a lecture or essay.

Configuration de l'invite vocale

Changement de code

Remarque : Cette fonctionnalité ne s'applique actuellement qu'à la langue hindi.

Choisissez l'une des instructions suivantes en fonction du script de sortie souhaité :

Pour la sortie en script latin (hindi romanisé) :

If the input audio/speech contains hindi, then the transcription and response should be in All Latin script (romanized Hindi).

Pour la sortie du script Devanagari :

If the input audio/speech contains hindi, then the transcription and response should be in All Devanagari script (Hindi).

Pour une sortie de script mixte (commutation de code naturelle) :

If the input audio/speech contains hindi, then the transcription and response can mix Latin and Devanagari scripts naturally for code-switching.

Configuration rapide du système

Contrôle de la verbosité des réponses

Réponses concises et conversationnelles :

You are a warm, professional, and helpful AI assistant. Give accurate answers that sound natural, direct, and human. Start by answering the user's question clearly in 1-2 sentences. Then, expand only enough to make the answer understandable, staying within 3-5 short sentences total. Avoid sounding like a lecture or essay.

Réponses détaillées et complètes :

You are a warm, professional, and helpful AI assistant. Give accurate, complete answers that sound warm, direct, and human. Answer the question directly in the first 1-2 sentences. if the question has parts or asks what/why/how, address each with a brief definition or main idea plus 2-3 key facts or steps. Offer practical, actionable advice. Keep a confident, kind, conversational tone; never robotic or theatrical. Be thorough; add examples or context only when helpful. Prefer accuracy and safety over speculation; if unsure, say so and suggest what to check.

Mise en miroir du langage

Nova peut reconnaître et répondre dans la langue parlée par l'utilisateur. Utilisez cette invite pour garantir la cohérence linguistique :

CRITICAL LANGUAGE MIRRORING RULES:

- Always reply in the language spoken. DO NOT mix with English. However, if the user talks in English, reply in English.

- Please respond in the language the user is talking to you in, If you have a question or suggestion, ask it in the language the user is talking in. I want to ensure that our communication remains in the same language as the user.

Accord sur le genre pour les langues sexospécifiques

Certaines langues exigent un accord entre les sexes dans les verbes, les adjectifs ou les pronoms lorsque l'assistant se décrit lui-même. Pour ces langues, spécifiez le sexe de l'assistant dans l'invite de votre système afin qu'il corresponde à la voix que vous avez sélectionnée.

Langues concernées : hindi, portugais, français, italien, espagnol, russe, polonais

Lorsque l'accord entre les sexes est important :

- Hindi : toujours nécessaire - les verbes se conjuguent en fonction du sexe du locuteur à la première personne
- Portugais/français : Nécessaire pour utiliser des participes ou des adjectifs antérieurs (tels que « Je suis fatigué » - « Estou cansada/cansado »)
- Italien/espagnol : Nécessaire lorsque vous utilisez des adjectifs pour vous décrire (tels que « Je suis heureux » - « Sono contenta/contento »)

Mise en œuvre :

Incluez l'identifiant de genre approprié au début de l'invite de votre système en fonction de votre sélection vocale :

Pour les voix féminines (Kiara, Carolina, Ambre, Beatrice, Lupe, Tiffany) :

You are a warm, professional, and helpful female AI assistant.

Pour les voix masculines (Arjun, Leo, Florian, Lorenzo, Carlos, Matthew) :

You are a warm, professional, and helpful male AI assistant.

Exemples :

Hindi avec voix féminine (kiara) :

You are a warm, professional, and helpful female AI assistant.

Résultat : « » (achchhi hoon principal) contre « » (achchha hoon principal) contre « » (achchha hoon principal)

Italien avec voix masculine (Lorenzo) :

You are a warm, professional, and helpful male AI assistant.

Résultat : « Sono contento » contre « Sono contenta »

Chaîne de pensée pour le discours : raisonnement constitutionnel

Utilisez cette invite lorsque vous souhaitez que le modèle montre son raisonnement pour résoudre des problèmes complexes :

You are a friendly assistant. The user will give you a problem. Explain your reasoning following the guidelines given in CONSTITUTION - REASONING, and summarize your decision at the end of your response, in one sentence.

CONSTITUTION - REASONING

1. For simple questions including simple calculations or contextual tasks: Give the answer directly. No explanation is necessary, although you can offer to provide more information if the user requests it.
2. When faced with complex problems or decisions, think through the steps systematically before providing your answer. Break down your reasoning process when it would help user understanding.
3. For subjective matters or comparisons: explain your thought process step-by-step.

Remarque : Si vous ne souhaitez pas que le modèle soit soumis à un raisonnement pour chaque demande, vous pouvez ajouter quelques exemples de plans à l'invite (voir les exemples ci-dessous).

You are a warm, professional, and helpful AI assistant. You converse in fluid and conversational English. Give accurate, complete answers that sound warm, direct, and human. Answer the question directly in the first 1-2 sentences. Keep a confident, kind, conversational tone; never robotic or theatrical. Avoid formatted lists or numbering and keep your output as a spoken transcript. Be concise but thorough; add examples or context only when helpful. Prefer accuracy and safety over speculation; if unsure, say so and suggest what to check. The user will give you a problem. Explain your reasoning following the guidelines given in CONSTITUTION - REASONING, and summarize your decision at the end of your response in one sentence.

CONSTITUTION - REASONING

1. When faced with complex problems or decisions, think through the steps systematically before providing your answer. Break down your reasoning process when it would help user understanding.
2. For subjective matters or comparisons: explain your thought process step-by-step.
3. For simple questions including simple calculations or contextual tasks: Give the answer directly. No explanation is necessary, although you can offer to provide more information if the user requests it.

EXAMPLES

User: What is $7 + 5$?

Assistant: 12.

User: What is the capital of India?

Assistant: Delhi is the capital of India.

User: I have a \$1,000 budget for a trip. Here are my costs... Can I afford it? Please explain your reasoning.

Assistant: (step-by-step breakdown + one-sentence conclusion)

Utilisation abusive de phrases suggérées

Nova Sonic 2 est plus sensible aux suggestions de phrases que Sonic 1. Cette sensibilité accrue n'est pas intrinsèquement bonne ou mauvaise, cela dépend de votre cas d'utilisation. Si vous voulez un phrasé cohérent et prévisible, cela peut être bénéfique. Toutefois, si vous souhaitez une variation plus naturelle, les listes de phrases explicites peuvent entraîner une surutilisation.

Si vous incluez des instructions contenant des listes explicites de phrases, le modèle les utilisera très fréquemment :

Exemple 1 - Phrases soulignées :

Instead of using bold or italics, emphasize important information by using phrases like "The key thing to remember is," "What's really important here is," or "I want to highlight that."

Exemple 2 - Remplisseurs conversationnels :

Include natural speech elements like "Well," "You know," "Actually," "I mean," or "By the way" at appropriate moments to create an authentic, casual conversation flow.

Recommandation :

- Si vous voulez un phrasé cohérent : les listes de phrases explicites fonctionnent bien dans Sonic 2 pour créer des réponses prévisibles et adaptées à la marque.
- Si vous souhaitez une variation naturelle : évitez de fournir des listes de phrases explicites. Utilisez plutôt des conseils généraux tels que « un son naturel et conversationnel » ou fournissez des exemples ponctuels.

Meilleure approche - Utilisez des exemples uniques :

Au lieu d'énumérer des phrases, fournissez un ou deux exemples illustrant le ton et le style souhaités :

Tonalité naturelle et utile

You are a warm, professional, and helpful AI assistant. Sound natural and conversational in your responses.

Example:

User: How do I reset my password?

Assistant: You can reset your password by clicking the "Forgot Password" link on the login page. You'll get an email with instructions to create a new one. The whole process usually takes just a couple of minutes.

Concis et direct

You are a helpful AI assistant. Provide clear, direct answers without unnecessary elaboration.

Example:

User: What's the weather like today?

Assistant: It's 72 degrees and sunny with a light breeze. Perfect day to be outside.

Professionnel empathique

You are a professional and empathetic AI assistant. Acknowledge the user's situation while providing practical solutions.

Example:

User: I'm frustrated because my order hasn't arrived yet.

Assistant: I understand how frustrating that must be, especially when you're waiting for something important. Let me check your order status right now. Can you provide your order number?

Technique mais accessible

You are a knowledgeable AI assistant who explains technical concepts in accessible language.

Example:

User: What is machine learning?

Assistant: Machine learning is when computers learn from examples rather than following strict rules. Think of it like teaching a child to recognize dogs—after seeing many dogs, they start recognizing new ones on their own. The computer does something similar with data.

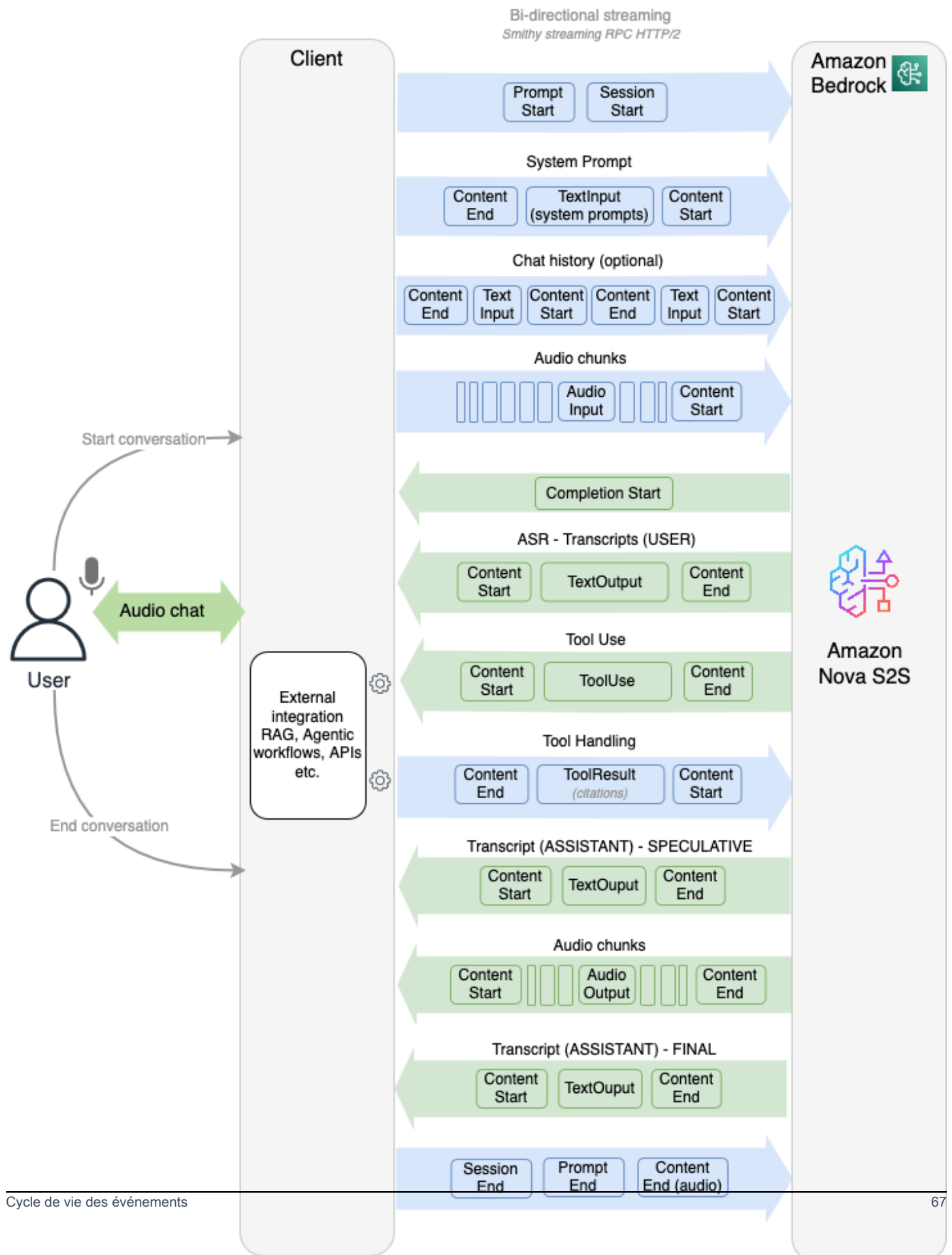
Cette approche montre au modèle le comportement souhaité sans déclencher de phrases répétitives, tout en conservant le contrôle du ton et du style.

Concepts de base

Amazon Nova 2 Sonic utilise une architecture de streaming bidirectionnelle avec des événements structurés pour une IA conversationnelle en temps réel. La compréhension de ces concepts fondamentaux est essentielle pour créer des applications vocales efficaces.

Cycle de vie des événements

Le schéma suivant illustre le cycle de vie complet des événements de streaming bidirectionnel :



Le cycle de vie des événements de streaming bidirectionnel suit un schéma structuré, de l'initialisation de la session à la fin de la conversation. Chaque conversation implique des événements d'entrée (depuis votre application) et des événements de sortie (depuis Amazon Nova 2 Sonic) qui fonctionnent ensemble pour créer des interactions vocales naturelles.

Séquence du flux d'événements

Une conversation typique suit la séquence d'événements suivante :

1. Début de session : initialise la session de conversation
2. System Prompt - Envoyer les instructions du système
3. Historique des discussions (facultatif) - Fournissez le contexte de la conversation
4. Audio Chunks - Diffusez l'entrée audio de l'utilisateur
5. Début de l'achèvement : l'IA commence le traitement
6. Transcriptions ASR (USER) - Transcription vocale de l'utilisateur
7. Utilisation de l'outil (facultatif) - L'IA demande l'exécution de l'outil
8. Manipulation des outils (facultatif) - Traiter et renvoyer les résultats de l'outil
9. Transcription (ASSISTANT) - SPÉCULATIF - Réponse préliminaire de l'IA
10. Audio Chunks - Diffusez une sortie audio AI
11. Transcription (ASSISTANT) - FINALE - Transcription finale de la réponse AI
12. Fin du contenu audio : marque la fin du contenu audio
13. Fin de l'invite : indique la fin du traitement de l'invite
14. Fin de session - Clôture de la conversation

Gestion des événements d'entrée avec l'API bidirectionnelle

L'API Stream bidirectionnelle utilise une architecture événementielle avec des événements d'entrée et de sortie structurés. Il est essentiel de comprendre l'ordre correct des événements pour implémenter des applications conversationnelles efficaces et maintenir un état de conversation approprié tout au long des interactions.

Présentation de

La conversation Nova Sonic suit une séquence d'événements structurée. Vous commencez par envoyer un événement `sessionStart` qui contient les paramètres de configuration de l'inférence,

tels que la température et les limites de jetons. Ensuite, vous envoyez `promptStart` pour définir le format de sortie audio et les configurations des outils, en attribuant un identifiant `promptName` unique qui doit être inclus dans tous les événements suivants.

Pour chaque type d'interaction (invite système, audio, etc.), vous suivez un schéma en trois parties : utilisez `contentStart` pour définir le type de contenu et le rôle du contenu (`SYSTEM`, `USER`, `ASSISTANT`, `TOOL`, `SYSTEM_SPEECH`), puis indiquez l'événement de contenu réel et terminez par `contentEnd` pour fermer ce segment. L'événement `contentStart` spécifie si vous envoyez les résultats d'un outil, un flux audio ou une invite système. L'événement `contentStart` comprend un identifiant `contentName` unique.

Historique des conversations

Un historique de conversation ne peut être inclus qu'une seule fois, après l'invite système et avant le début du flux audio. Il suit le même modèle `contentStart/textInput/contentEnd`. Les rôles `USER` et `ASSISTANT` doivent être définis dans l'événement `contentStart` pour chaque message historique. Cela fournit un contexte essentiel pour la conversation en cours, mais doit être effectué avant que toute nouvelle entrée utilisateur ne commence.

Diffusion audio

Le streaming audio fonctionne avec un échantillonnage continu du microphone. Après l'envoi d'un `contentStart` initial, les trames audio (environ 32 ms chacune) sont capturées directement à partir du microphone et immédiatement envoyées en tant qu'événements `audioInput` en utilisant le même `contentName`. Ces échantillons audio doivent être diffusés en temps réel au fur et à mesure qu'ils sont capturés, en conservant la cadence naturelle d'échantillonnage du microphone tout au long de la conversation. Toutes les trames audio partagent un seul conteneur de contenu jusqu'à la fin de la conversation et sa fermeture explicite.

Clôture de la session

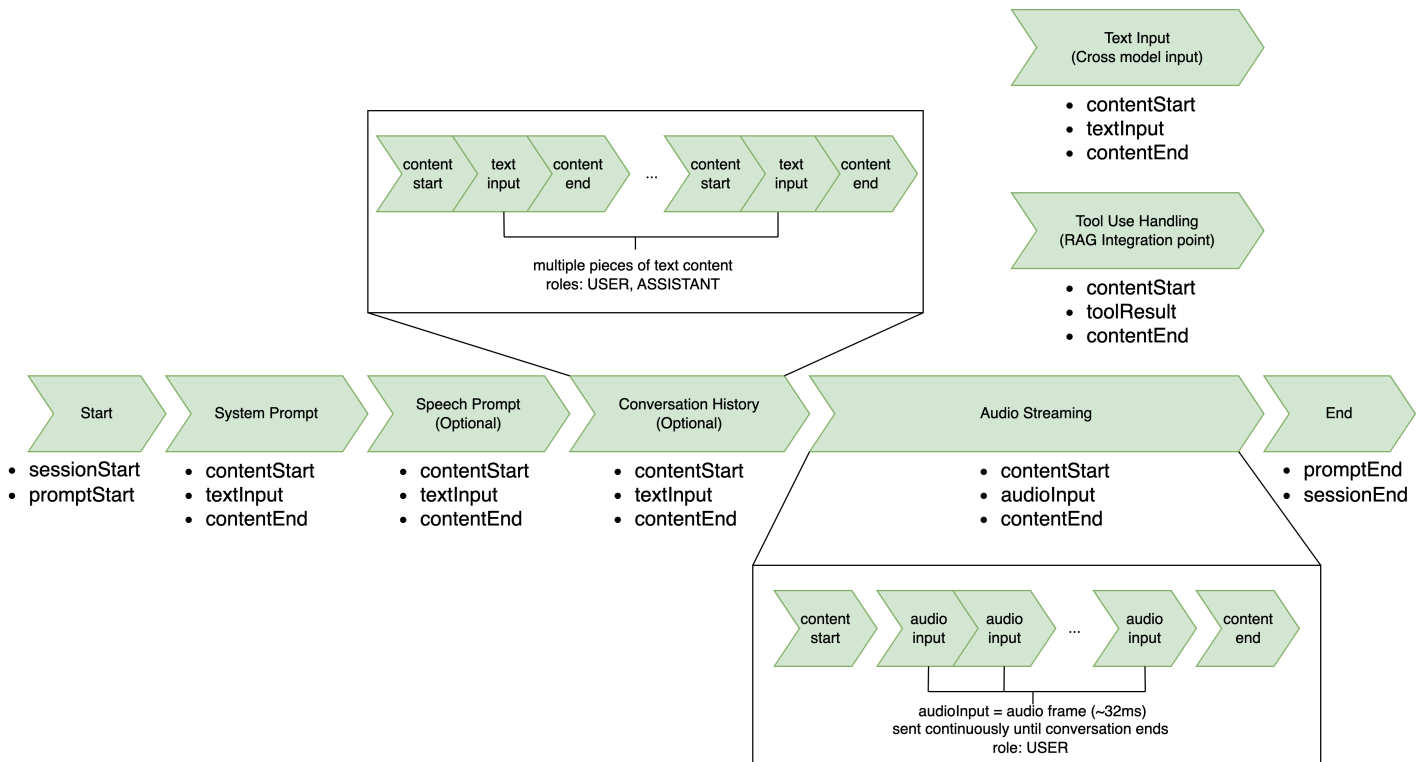
Une fois la conversation terminée ou lorsqu'il est nécessaire d'y mettre fin, il est essentiel de fermer correctement tous les flux ouverts et de terminer la session dans le bon ordre. Pour terminer correctement une session et éviter les fuites de ressources, vous devez suivre une séquence de fermeture spécifique :

- Fermez tous les flux audio ouverts avec l'événement `contentEnd`.
- Envoyez un événement `promptEnd` qui fait référence au `promptName` d'origine.

- Envoyez l'événement `sessionEnd`.

Le fait d'ignorer l'un de ces événements de fermeture peut entraîner des conversations incomplètes ou des ressources orphelines.

Ces identifiants créent une structure hiérarchique : le `promptName` relie tous les événements de conversation, tandis que chaque `contentName` marque les limites de blocs de contenu spécifiques. Cette hiérarchie garantit que le modèle conserve un contexte approprié tout au long de l'interaction.



Flux d'événements d'entrée

La structure du flux d'événements d'entrée est présentée dans cette section.

1. RequestStartEvent (Début de session)

L'événement de démarrage de session initialise la conversation avec les paramètres de configuration d'inférence et de détection des virages.

Configuration d'inférence :

- `maxTokens`: nombre maximum de jetons à générer dans la réponse
- `topP`: paramètre d'échantillonnage du noyau (0,0 à 1,0) pour contrôler le caractère aléatoire

- **temperature**: Contrôle le caractère aléatoire de la génération (0,0 à 1,0)

Configuration de la détection des virages : le `endpointingSensitivity` paramètre contrôle la rapidité avec laquelle Nova Sonic détecte lorsqu'un utilisateur a fini de parler :

- **HIGH**: détecte rapidement les pauses, permettant des réponses plus rapides mais peut interrompre les haut-parleurs plus lents
- **MEDIUM**: sensibilité équilibrée pour la plupart des scénarios de conversation (valeur par défaut recommandée)
- **LOW**: attend plus longtemps avant de détecter la fin du discours, ce qui est préférable pour les orateurs attentionnés ou hésitants

```
{
  "event": {
    "sessionStart": {
      "inferenceConfiguration": {
        "maxTokens": "int",
        "topP": "float",
        "temperature": "float"
      },
      "turnDetectionConfiguration": {
        "endpointingSensitivity": "HIGH" | "MEDIUM" | "LOW"
      }
    }
  }
}
```

Exemple :

```
{
  "event": {
    "sessionStart": {
      "inferenceConfiguration": {
        "maxTokens": 2048,
        "topP": 0.9,
        "temperature": 0.7
      },
      "turnDetectionConfiguration": {
        "endpointingSensitivity": "MEDIUM"
      }
    }
  }
}
```

```

    }
  }
}

```

2. PromptStartEvent

L'événement de démarrage rapide définit la configuration de la conversation, notamment les formats de sortie, la sélection vocale et les outils disponibles.

Pour obtenir la liste des voix disponibles IDs, reportez-vous à la section Prise en [charge linguistique et fonctionnalités multilingues](#)

```

{
  "event": {
    "promptStart": {
      "promptName": "string", // unique identifier same across all events i.e.
      UUID
      "textOutputConfiguration": {
        "mediaType": "text/plain"
      },
      "audioOutputConfiguration": {
        "mediaType": "audio/lpcm",
        "sampleRateHertz": 8000 | 16000 | 24000,
        "sampleSizeBits": 16,
        "channelCount": 1,
        "voiceId": "matthew" | "tiffany" | "amy" | "olivia" | "lupe" | "carlos"
        | "ambre" | "florian" | "lennart" | "beatrice" | "lorenzo" |
          "tina" | "carolina" | "leo" | "kiara" | "arjun",
        "encoding": "base64",
        "audioType": "SPEECH"
      },
      "toolUseOutputConfiguration": {
        "mediaType": "application/json"
      },
      "toolConfiguration": {
        "tools": [
          {
            "toolSpec": {
              "name": "string",
              "description": "string",
              "inputSchema": {
                "json": "{}"
              }
            }
          }
        ]
      }
    }
  }
}

```

```
}
}
}
}
}
}
```

3. InputContentStartEvent

Texte

L'événement de démarrage du contenu textuel est utilisé pour les instructions du système, l'historique des conversations et la saisie de texte intermodale.

Paramètre interactif :

- `true`: permet la saisie intermodale, autorisant les messages texte pendant une session vocale active
- `false`: saisie de texte standard pour les instructions du système et l'historique des conversations

Types de rôles :

- SYSTEM: instructions et instructions du système
- USER: messages utilisateur dans l'historique des conversations ou saisie intermodale
- ASSISTANT: réponses de l'assistant dans l'historique des conversations
- SYSTEM_SPEECH: Contrôle le formatage de la transcription pour le changement de code en hindi (scripts) Latin/Devanagari/mixed

```
{
  "event": {
    "contentStart": {
      "promptName": "string", // same unique identifier from promptStart event
      "contentName": "string", // unique identifier for the content block
      "type": "TEXT",
      "interactive": "boolean", // true for cross-modal input
      "role": "SYSTEM" | "USER" | "ASSISTANT" | "TOOL" | "SYSTEM_SPEECH",
      "textInputConfiguration": {
        "mediaType": "text/plain"
      }
    }
  }
}
```

```

    }
  }
}

```

Exemple - Invite système :

```

{
  "event": {
    "contentStart": {
      "promptName": "conv-12345",
      "contentName": "system-prompt-1",
      "type": "TEXT",
      "interactive": false,
      "role": "SYSTEM",
      "textInputConfiguration": {
        "mediaType": "text/plain"
      }
    }
  }
}

```

Exemple - Entrée intermodale :

```

{
  "event": {
    "contentStart": {
      "promptName": "conv-12345",
      "contentName": "user-text-1",
      "type": "TEXT",
      "interactive": true,
      "role": "USER",
      "textInputConfiguration": {
        "mediaType": "text/plain"
      }
    }
  }
}

```

Audio

```

{

```

```

"event": {
  "contentStart": {
    "promptName": "string", // same unique identifier from promptStart event
    "contentName": "string", // unique identifier for the content block
    "type": "AUDIO",
    "interactive": true,
    "role": "USER",
    "audioInputConfiguration": {
      "mediaType": "audio/lpcm",
      "sampleRateHertz": 8000 | 16000 | 24000,
      "sampleSizeBits": 16,
      "channelCount": 1,
      "audioType": "SPEECH",
      "encoding": "base64"
    }
  }
}

```

Outil

```

{
  "event": {
    "contentStart": {
      "promptName": "string", // same unique identifier from promptStart event
      "contentName": "string", // unique identifier for the content block
      "interactive": false,
      "type": "TOOL",
      "role": "TOOL",
      "toolResultInputConfiguration": {
        "toolUseId": "string", // existing tool use id
        "type": "TEXT",
        "textInputConfiguration": {
          "mediaType": "text/plain"
        }
      }
    }
  }
}

```

4. TextInputContent

```

{

```

```
"event": {
  "textInput": {
    "promptName": "string", // same unique identifier from promptStart event
    "contentName": "string", // unique identifier for the content block
    "content": "string"
  }
}
```

5. AudioInputContent

```
{
  "event": {
    "audioInput": {
      "promptName": "string", // same unique identifier from promptStart event
      "contentName": "string", // same unique identifier from its contentStart
      "content": "base64EncodedAudioData"
    }
  }
}
```

6. ToolResultContentEvent

```
"event": {
  "toolResult": {
    "promptName": "string", // same unique identifier from promptStart event
    "contentName": "string", // same unique identifier from its contentStart
    "content": "{\"key\": \"value\"}" // stringified JSON object as a tool result
  }
}
```

7. InputContentEndEvent

```
{
  "event": {
    "contentEnd": {
      "promptName": "string", // same unique identifier from promptStart event
      "contentName": "string" // same unique identifier from its contentStart
    }
  }
}
```

8. PromptEndEvent

```
{
  "event": {
    "promptEnd": {
      "promptName": "string" // same unique identifier from promptStart event
    }
  }
}
```

9. RequestEndEvent

```
{
  "event": {
    "sessionEnd": {}
  }
}
```

Gestion des événements de sortie avec l'API bidirectionnelle

Lorsque le modèle Amazon Nova Sonic répond, il suit une séquence d'événements structurée. Le flux commence par un événement `completionStart` qui contient des identifiants uniques tels que `sessionId`, `promptName` et `completionId`. Ces identifiants sont cohérents tout au long du cycle de réponse et relient tous les événements de réponse suivants.

Présentation de

Chaque type de réponse suit un modèle cohérent en trois parties : `contentStart` définit le type et le format du contenu, l'événement de contenu réel, et `contentEnd` clôt ce segment. La réponse comprend généralement plusieurs blocs de contenu en séquence : transcription par reconnaissance vocale automatique (ASR) (ce que l'utilisateur a dit), utilisation facultative d'un outil (lorsque des informations externes sont nécessaires), réponse textuelle (ce que le modèle prévoit de dire) et réponse audio (la sortie vocale).

Types de contenu de réponse

Transcription ASR

La transcription ASR apparaît en premier, fournissant la compréhension du modèle de la parole de l'utilisateur avec `role: "USER"` et `"additionalModelFields": "{ \"generationStage\": \"FINAL\" }"` dans le `contentStart`.

Utilisation de l'outil

Lorsque le modèle a besoin de données externes, il envoie des événements liés à des outils avec des noms et des paramètres d'outils spécifiques.

Réponse textuelle

La réponse textuelle fournit un aperçu de la parole prévue avec `role`: "ASSISTANT" et `"additionalModelFields"`: "{\\"generationStage\\":\\"SPECULATIVE\\"}".

Réponse audio

La réponse audio fournit ensuite des segments de parole encodés en base64 partageant le même `contentId` tout au long du flux.

Support Barge-In

Pendant la génération audio, Amazon Nova Sonic prend en charge le flux naturel de la conversation grâce à sa fonctionnalité d'interruption. Lorsqu'un utilisateur interrompt Amazon Nova Sonic pendant qu'il parle, Nova Sonic arrête immédiatement de générer la parole, passe en mode écoute et envoie une notification de contenu indiquant que l'interruption s'est produite. Comme Nova Sonic fonctionne plus rapidement qu'en temps réel, certains éléments audio peuvent avoir déjà été transmis mais ne pas avoir encore été lus. La notification d'interruption permet à l'application cliente d'effacer sa file d'attente audio et d'arrêter immédiatement la lecture, créant ainsi une expérience conversationnelle réactive.

Transcription finale

Une fois la génération audio terminée (ou interrompue via l'interruption), Amazon Nova Sonic fournit une réponse textuelle supplémentaire qui contient une transcription au niveau de la phrase de ce qui a été réellement dit. Cette réponse textuelle comprend un événement `contentStart` avec `role`: "ASSISTANT" et `"additionalModelFields"`: "{\\"generationStage\\":\\"FINAL\\"}".

Suivi de l'utilisation

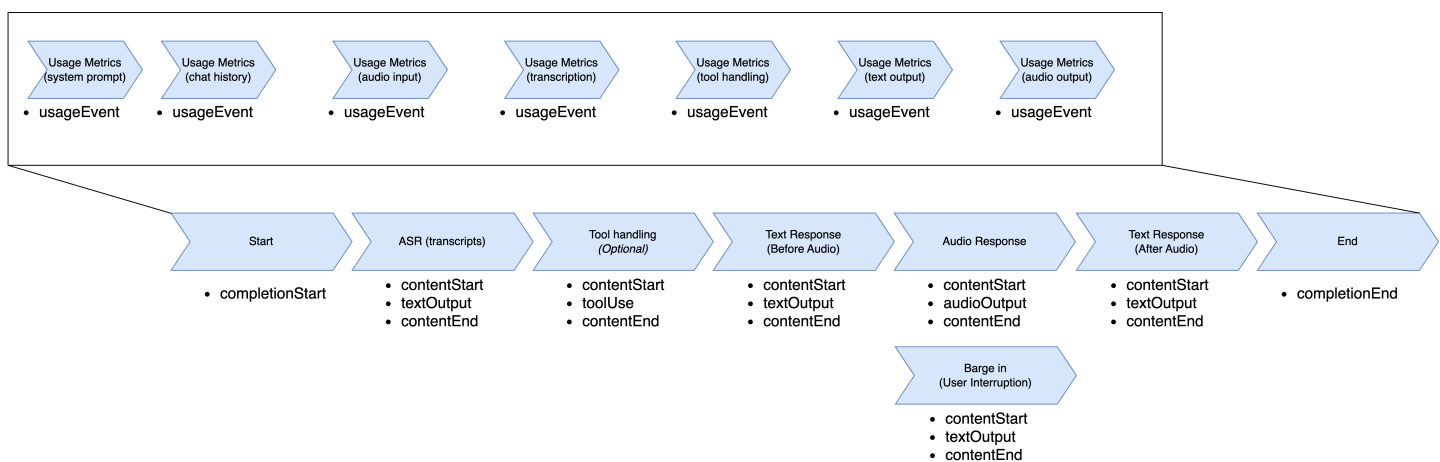
Tout au long du traitement de la réponse, des événements `usageEvent` sont envoyés pour suivre la consommation de jetons. Ces événements contiennent des métriques détaillées sur les jetons d'entrée et de sortie (voix et texte), ainsi que leurs totaux cumulés. Chaque `usageEvent` conserve les mêmes `sessionId`, `promptName` et `completionId` que les autres événements du flux de conversation. La section des détails fournit à la fois les changements incrémentiels (`delta`) et les

totaux cumulés de l'utilisation des jetons, ce qui permet un suivi précis de l'utilisation pendant la conversation.

Achèvement

Le modèle envoie un événement `completionEnd` avec les identifiants d'origine et un `stopReason` qui indique comment la conversation s'est terminée. Cette hiérarchie d'événements garantit que votre application peut suivre les parties de la réponse qui vont ensemble et les traiter en conséquence, en conservant le contexte de la conversation tout au long des multiples tours.

Le flux d'événements de sortie commence par entrer dans la phase de génération de réponse. Il commence par la reconnaissance vocale automatique, sélectionne un outil à utiliser, transcrit la parole, génère l'audio, finalise la transcription et termine la session.



Flux d'événements de sortie

La structure du flux d'événements de sortie est décrite dans cette section.

1. UsageEvent

```

"event": {
  "usageEvent": {
    "completionId": "string", // unique identifier for completion
    "details": {
      "delta": { // incremental changes since last event
        "input": {
          "speechTokens": number, // input speech tokens
          "textTokens": number // input text tokens
        },
        "output": {
          "speechTokens": number, // speech tokens generated
        }
      }
    }
  }
}
  
```

```

        "textTokens": number // text tokens generated
    },
    "total": { // cumulative counts
        "input": {
            "speechTokens": number, // total speech tokens processed
            "textTokens": number // total text tokens processed
        },
        "output": {
            "speechTokens": number, // total speech tokens generated
            "textTokens": number // total text tokens generated
        }
    },
    "promptName": "string", // same unique identifier from promptStart event
    "sessionId": "string", // unique identifier
    "totalInputTokens": number, // cumulative input tokens
    "totalOutputTokens": number, // cumulative output tokens
    "totalTokens": number // total tokens in the session
}
}

```

2. CompleteStartEvent

```

"event": {
    "completionStart": {
        "sessionId": "string", // unique identifier
        "promptName": "string", // same unique identifier from promptStart event
        "completionId": "string", // unique identifier
    }
}

```

3. TextOutputContent

ContentStart

```

"event": {
    "contentStart": {
        "additionalModelFields": "{\\"generationStage\\":\\"FINAL\\"}" |
        "{\\"generationStage\\":\\"SPECULATIVE\\"}",
        "sessionId": "string", // unique identifier
        "promptName": "string", // same unique identifier from promptStart event
        "completionId": "string", // unique identifier
    }
}

```

```

        "contentId": "string", // unique identifier for the content block
        "type": "TEXT",
        "role": "USER" | "ASSISTANT",
        "textOutputConfiguration": {
            "mediaType": "text/plain"
        }
    }
}

```

TextOutput

```

"event": {
    "textOutput": {
        "sessionId": "string", // unique identifier
        "promptName": "string", // same unique identifier from promptStart event
        "completionId": "string", // unique identifier
        "contentId": "string", // same unique identifier from its contentStart
        "content": "string" // User transcribe or Text Response
    }
}

```

ContentEnd

```

"event": {
    "contentEnd": {
        "sessionId": "string", // unique identifier
        "promptName": "string", // same unique identifier from promptStart event
        "completionId": "string", // unique identifier
        "contentId": "string", // same unique identifier from its contentStart
        "stopReason": "PARTIAL_TURN" | "END_TURN" | "INTERRUPTED",
        "type": "TEXT"
    }
}

```

4. ToolUse

ContentStart

```

"event": {
    "contentStart": {
        "sessionId": "string", // unique identifier
        "promptName": "string", // same unique identifier from promptStart event
    }
}

```

```

    "completionId": "string", // unique identifier
    "contentId": "string", // unique identifier for the content block
    "type": "T00L",
    "role": "T00L",
    "toolUseOutputConfiguration": {
        "mediaType": "application/json"
    }
  }
}

```

ToolUse

```

"event": {
  "toolUse": {
    "sessionId": "string", // unique identifier
    "promptName": "string", // same unique identifier from promptStart event
    "completionId": "string", // unique identifier
    "contentId": "string", // same unique identifier from its contentStart
    "content": "json",
    "toolName": "string",
    "toolUseId": "string"
  }
}

```

ContentEnd

```

"event": {
  "contentEnd": {
    "sessionId": "string", // unique identifier
    "promptName": "string", // same unique identifier from promptStart event
    "completionId": "string", // unique identifier
    "contentId": "string", // same unique identifier from its contentStart
    "stopReason": "T00L_USE",
    "type": "T00L"
  }
}

```

5. AudioOutputContent

ContentStart

```

"event": {

```

```

    "contentStart": {
      "sessionId": "string", // unique identifier
      "promptName": "string", // same unique identifier from promptStart event
      "completionId": "string", // unique identifier
      "contentId": "string", // unique identifier for the content block
      "type": "AUDIO",
      "role": "ASSISTANT",
      "audioOutputConfiguration": {
        "mediaType": "audio/lpcm",
        "sampleRateHertz": 8000 | 16000 | 24000,
        "sampleSizeBits": 16,
        "encoding": "base64",
        "channelCount": 1
      }
    }
  }
}

```

AudioOutput

```

"event": {
  "audioOutput": {
    "sessionId": "string", // unique identifier
    "promptName": "string", // same unique identifier from promptStart event
    "completionId": "string", // unique identifier
    "contentId": "string", // same unique identifier from its contentStart
    "content": "base64EncodedAudioData", // Audio
  }
}

```

ContentEnd

```

"event": {
  "contentEnd": {
    "sessionId": "string", // unique identifier
    "promptName": "string", // same unique identifier from promptStart event
    "completionId": "string", // unique identifier
    "contentId": "string", // same unique identifier from its contentStart
    "stopReason": "PARTIAL_TURN" | "END_TURN",
    "type": "AUDIO"
  }
}

```

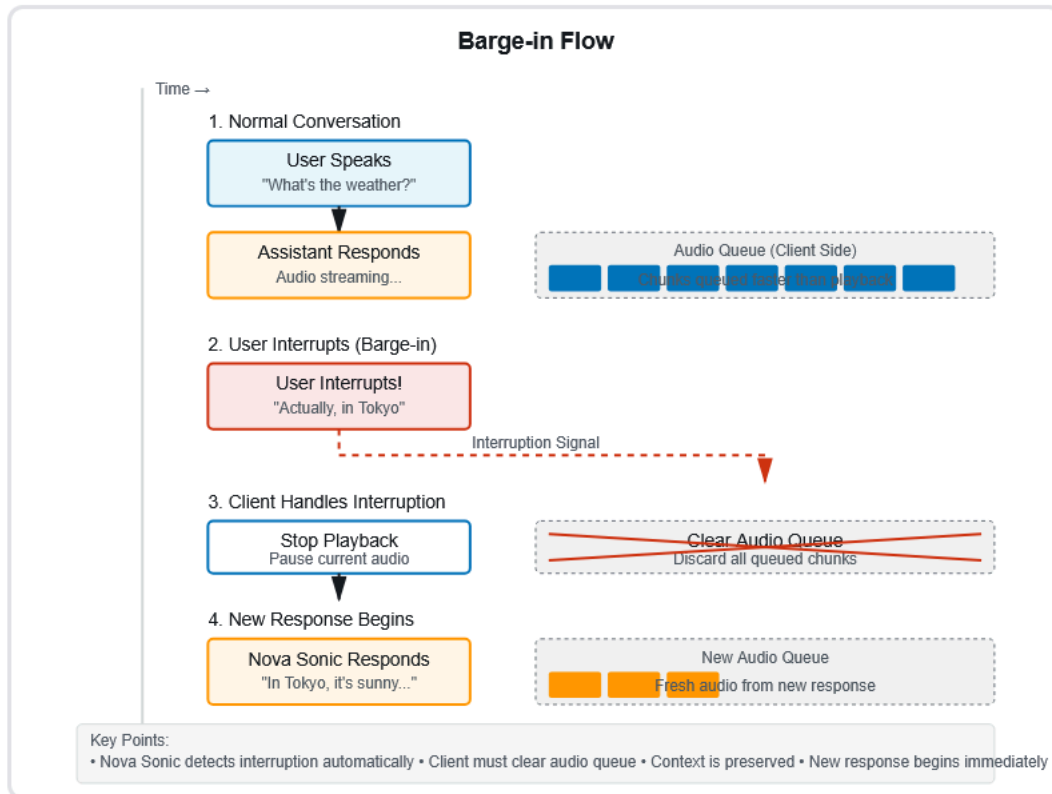
6. CompletionEndEvent

```
"event": {
  "completionEnd": {
    "sessionId": "string", // unique identifier
    "promptName": "string", // same unique identifier from promptStart event
    "completionId": "string", // unique identifier
    "stopReason": "END_TURN"
  }
}
```

Barge-in

Barge-in permet aux utilisateurs d'interrompre l'assistant AI pendant qu'il parle, comme dans les conversations humaines naturelles. Au lieu d'attendre que l'assistant ait terminé, les utilisateurs peuvent introduire de nouvelles informations, corriger ou clarifier leur déclaration précédente, rediriger la conversation vers un autre sujet ou simplement arrêter l'assistant lorsqu'ils en ont assez entendu. Cela crée une expérience conversationnelle plus naturelle et plus réactive.

Le schéma suivant illustre le flux complet des conversations intégrées :



Comment Amazon Nova 2 Sonic gère le barge-in

Amazon Nova 2 Sonic est conçu pour gérer les interruptions avec élégance. Lorsque l'utilisateur commence à parler pendant une réponse, le système arrête immédiatement de générer la réponse en cours, maintient le contexte conversationnel complet, envoie un signal d'interruption au client et commence à traiter la nouvelle entrée utilisateur.

Préservation du contexte : Même en cas d'interruption, Nova Sonic se souvient de ce qui a été dit avant l'interruption, du sujet discuté, de l'historique des conversations et de tout contexte pertinent des tours précédents. Cela garantit que la conversation reste cohérente et naturelle.

Exigences de mise en œuvre côté client

Alors qu'Amazon Nova 2 Sonic gère le barge-in côté serveur, vous devez implémenter une logique côté client pour une expérience complète.

Le défi de la file d'attente audio : la génération audio est plus rapide que la vitesse de lecture. Autrement dit :

- Nova Sonic génère rapidement des morceaux audio
- Votre client reçoit et met en file d'attente ces morceaux
- Le client les rejoue à une vitesse de parole normale
- Lorsqu'une intrusion se produit, le son est déjà en file d'attente pour la lecture

Logique requise côté client : votre application doit gérer quatre étapes clés :

1. **Détectez le signal d'interruption :** écoutez l'événement d'interruption de Nova Sonic et réagissez immédiatement lorsqu'il est reçu.
2. **Arrêter la lecture en cours :** mettez en pause le son en cours de lecture et arrêtez tout son en cours de lecture.
3. **Effacez la file d'attente audio :** supprimez tous les morceaux audio en file d'attente et supprimez tout son mis en mémoire tampon de la réponse interrompue.
4. **Démarrer un nouvel audio :** commencez à lire le nouveau fichier audio reçu et reprenez le flux de lecture normal.

Contrôlabilité à tour de rôle

La prise de tour est un aspect fondamental de la conversation naturelle. Amazon Nova 2 Sonic permet de contrôler avec précision le moment où l'IA passe à son tour pour définir le `turnDetectionConfiguration` paramètre. Cela vous permet d'optimiser le flux de conversation pour différents cas d'utilisation, en équilibrant réactivité et précision. Le paramètre `EndpointingSensitivity` contrôle la rapidité avec laquelle Amazon Nova 2 Sonic détecte la fin du tour d'un utilisateur et commence à réagir. Ce paramètre affecte à la fois la latence des réponses et la probabilité d'interrompre les utilisateurs qui parlent encore.

Configuration de l'API

Configurez la sensibilité de détection des virages dans l'événement `SessionStart` :

```
{
  "event": {
    "sessionStart": {
      "inferenceConfiguration": {
        "maxTokens": 1000,
        "topP": 0.9,
        "temperature": 0.7
      },
      "turnDetectionConfiguration": {
        "endpointingSensitivity": "HIGH" | "MEDIUM" | "LOW"
      }
    }
  }
}
```

Niveaux de sensibilité

Le paramètre `EndpointingSensitivity` accepte trois valeurs : `HIGH`, `MEDIUM` et `LOW`. Chaque niveau équilibre la vitesse de réponse et le risque d'interrompre les utilisateurs qui parlent encore.

ÉLEVÉ

Temps de réponse le plus rapide, optimisé pour la latence. Nova Sonic répond le plus rapidement possible après avoir détecté la fin du discours. Durée de la pause : 1,5 seconde. Idéal pour les questions et réponses rapides, command-and-control les applications et les interactions urgentes.

MOYEN

Approche équilibrée avec temps de réponse modéré. Réduit les faux positifs tout en maintenant la réactivité. Durée de la pause : 1,75 seconde. Idéal pour les conversations générales, le service client avec des requêtes complexes et les discussions à plusieurs tours.

BAS

Temps de réponse le plus lent avec un maximum de patience. Nova Sonic attend le plus longtemps avant de répondre, minimisant ainsi les interruptions des utilisateurs qui font une pause pendant qu'ils réfléchissent. Durée de la pause : 2 secondes. Idéal pour les conversations réfléchies, les personnes âgées ou ayant des troubles de la parole et la résolution de problèmes complexes.

Référence de durée de pause

Niveau de sensibilité	Durée de la pause (secondes)
Haut (rapide)	1.5
Moyenne	1,75
Faible (lent)	~2,0

Entrée intermodale

Amazon Nova 2 Sonic prend désormais en charge la saisie intermodale, ce qui vous permet d'envoyer des messages texte en plus de la saisie vocale lors d'une session de conversation. Bien que la parole reste le principal mode d'interaction, la saisie de texte offre de la flexibilité pour les scénarios où la saisie est plus pratique ou appropriée.

Streaming continu requis : l'entrée intermodale nécessite une session de streaming active pour fonctionner correctement. La session doit maintenir un streaming continu comme une session vocale normale, sinon des délais d'expiration de session standard seront appliqués et la connexion sera interrompue.

Les niveaux de sensibilité lors de la saisie de texte intermodale sont utiles pour des scénarios tels que :

- Intégration des applications côté client (Web et mobile) : permet aux utilisateurs d'interagir avec l'application à la fois par le texte et par la voix, garantissant ainsi des expériences multimodales fluides.
- Cas d'utilisation odel-start-first « M » : un message texte peut être envoyé immédiatement après le début de la session pour inviter le modèle à commencer à parler.
- Guide du modèle lors des appels d'outils asynchrones : lorsqu'un événement ToolUse est déclenché et que le système commence à traiter les appels d'outils, le client peut envoyer un message texte à Sonic pour fournir une réponse naturelle en attendant, par exemple : « Attendez une seconde pendant que je traite vos informations. En attendant, y a-t-il autre chose que je puisse faire pour vous aider ? »
- Intégration de la téléphonie DTMF : le client utilise le clavier du téléphone pour saisir des informations sensibles (telles que les numéros de carte de crédit). Remarque : Amazon Nova Sonic ne traite pas les tonalités DTMF de manière native. Pour prendre en charge l'entrée DTMF, votre système doit détecter les sons, les convertir en texte (comme « 1234 ») et les envoyer à Nova 2 Sonic.

Comment ça marche

L'entrée intermodale utilise une séquence de trois événements similaire à l'entrée audio :

1. Événement de démarrage du contenu : indique le début de la saisie de texte
2. Événement de saisie de texte : contient le message texte réel
3. Événement de fin de contenu : signale la fin de la saisie de texte

Les trois événements doivent utiliser les mêmes PromptName et ContentName pour conserver la séquence. Un nouvel UUID doit être généré pour ContentName chaque fois que vous envoyez du texte afin de garantir un suivi correct des conversations à plusieurs tours.

Structure d'évènements

1. Événement de lancement de contenu

Lance la séquence de saisie de texte avec les détails de configuration :

```
{
  "event": {
    "contentStart": {
```

```
    "promptName": "<prompt_name>",
    "contentName": "<new_content_name>",
    "role": "USER",
    "type": "TEXT",
    "interactive": true,
    "textInputConfiguration": {
      "mediaType": "text/plain"
    }
  }
}
```

Paramètres clés:

- **promptName**: Le nom de votre invite de conversation (identique tout au long de la session)
- **contentName**: identifiant unique pour cette saisie de texte (générez un nouvel UUID pour chaque message)
- **role**: défini sur "USER" pour indiquer la saisie par l'utilisateur
- **type**: défini sur "TEXT" pour la saisie de texte
- **interactive**: défini sur true pour activer le mode interactif
- **mediaType**: défini sur "text/plain" pour le contenu en texte brut

2. Événement de saisie de texte

Contient le contenu réel du message texte :

```
{
  "event": {
    "textInput": {
      "promptName": "<prompt_name>",
      "contentName": "<new_content_name>",
      "content": "<your_text_message>"
    }
  }
}
```

Paramètres clés:

- **promptName**: doit correspondre à la valeur de l'événement de démarrage du contenu

- `contentName`: doit correspondre à la valeur de l'événement de démarrage du contenu
- `role`: chaîne de votre message texte

3. Événement de fin de contenu

Signale la fin de la saisie de texte :

```
{
  "event": {
    "contentEnd": {
      "promptName": "<prompt_name>",
      "contentName": "<new_content_name>"
    }
  }
}
```

Paramètres clés:

- `promptName`: doit correspondre à la valeur des événements précédents
- `contentName`: doit correspondre à la valeur des événements précédents

Support linguistique et fonctionnalités multilingues

Amazon Nova 2 Sonic propose une sélection variée de voix dans plusieurs langues, ce qui vous permet de créer des applications d'intelligence artificielle conversationnelles qui soient naturelles et culturellement adaptées à vos utilisateurs. Chaque langue propose des options vocales à consonance féminine et à consonance masculine.

Le tableau suivant répertorie toutes les voix disponibles et les langues locales correspondantes :

Language	Paramètre régional	Identification vocale à consonance féminine	Identification vocale à consonance masculine
Anglais (États-Unis)	en-US	tiffany	matthew
Anglais (Royaume-Uni)	en-GB	amy	-

Language	Paramètre régional	Identification vocale à consonance féminine	Identification vocale à consonance masculine
Anglais (Australie)	en-AU	olivia	-
Anglais (Inde)	en-IN	kiara	arjun
Français	fr-FR	ambre	florian
Italien	it-IT	béatrice	lorenzo
Allemand	de-DE	tina	lennart
Espagnol (États-Unis)	es-US	lupe	carlos
Portugais	pt-BR	caroline	leo
Hindi	hi-IN	kiara	arjun

Structure de l'événement à l'aide de voix dans votre application

Vous pouvez spécifier l'identifiant vocal dans le `audioOutputConfiguration` lorsque vous lancez une invite dans l'`promptStart` événement :

```
"event": {
  "promptStart": {
    "promptName": "string",
    "audioOutputConfiguration": {
      "mediaType": "audio/lpcm",
      "sampleRateHertz": 16000,
      "sampleSizeBits": 16,
      "channelCount": 1,
      "voiceId": "tiffany",
      "encoding": "base64",
      "audioType": "SPEECH"
    }
  }
}
```

Support multilingue

Amazon Nova 2 Sonic fournit de puissantes fonctionnalités multilingues qui permettent des conversations naturelles dans plusieurs langues. Le service prend en charge à la fois les voix polyglottes (parlant plusieurs langues) et le changement de code (mélange des langues au sein d'une même phrase), ce qui vous permet de créer des applications conversationnelles véritablement mondiales.

TIFFANY (en-US, femme) et MATTHEW (en-US, homme) sont des voix polyglottes uniques qui peuvent parler toutes les langues prises en charge :

1. Anglais
2. Français
3. Italien
4. Allemand
5. Espagnol
6. Portugais
7. Hindi

Tiffany et Matthew sont donc idéaux pour les applications qui doivent passer facilement d'une langue à l'autre.

Gestion de l'historique des discussions

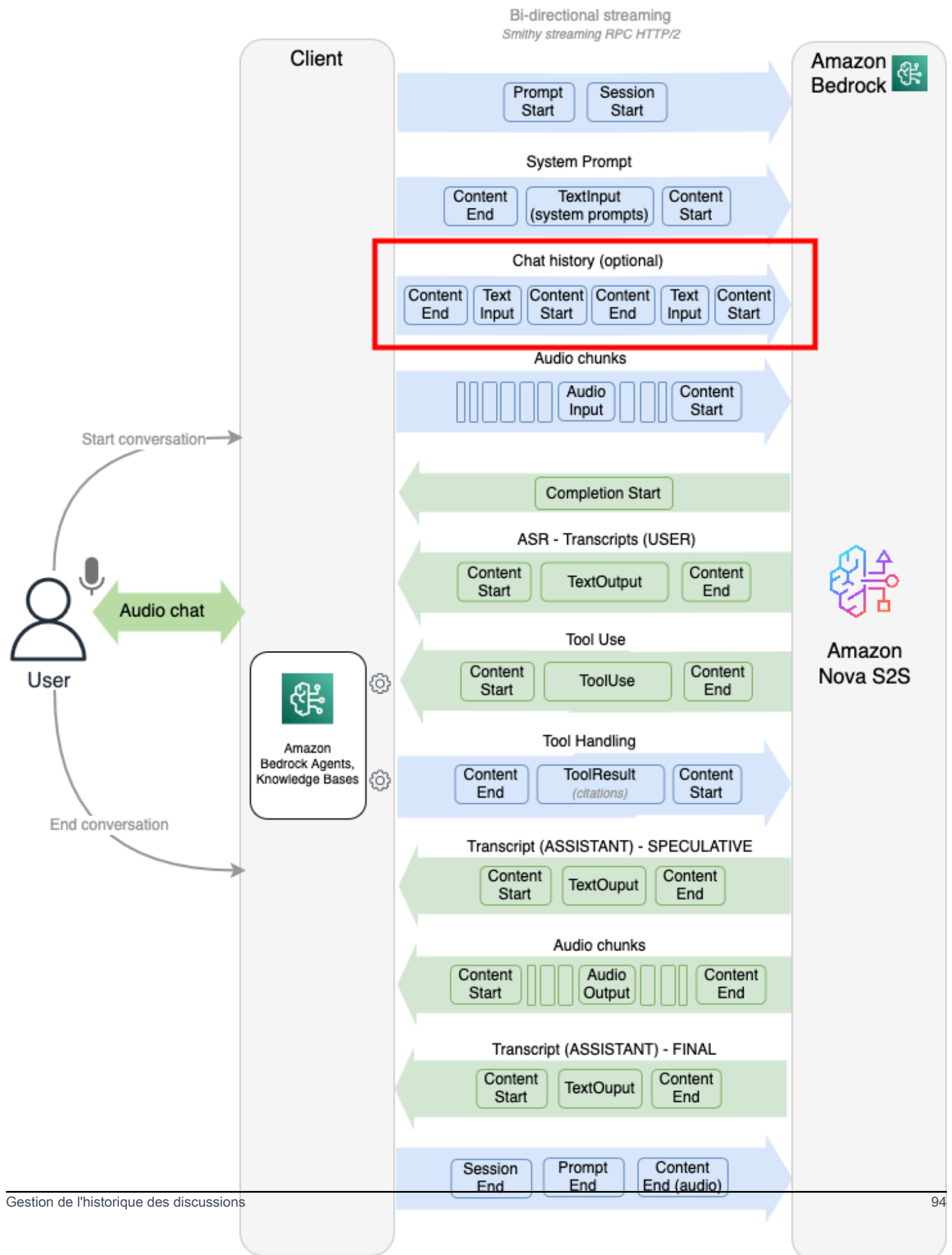
Les réponses Amazon Nova 2 Sonic incluent des transcriptions ASR (reconnaissance vocale automatique) pour les voix des utilisateurs et des assistants. Le stockage de l'historique des discussions est une bonne pratique, non seulement à des fins de journalisation, mais également pour reprendre les sessions lorsque la connexion est interrompue de manière inattendue. Cela permet au client de renvoyer le contexte à Nova Sonic pour poursuivre la conversation de manière fluide.

Consultez les ressources suivantes pour plus d'informations sur la gestion de l'historique des discussions :

1. [Enregistrement de l'historique des discussions](#)
2. [Reprise des conversations](#)

Envoi de l'historique des discussions

L'historique des conversations ne peut être inclus qu'une seule fois, après l'invite système/vocale et avant le début de la diffusion audio. L'historique global des discussions ne peut pas dépasser 40 Ko. Le schéma suivant montre à quel moment l'historique des discussions est transmis au cours du cycle de vie de l'événement :



Chaque message historique nécessite trois événements : `contentStart`, `textInput` et `contentEnd`.

Schéma d'événement par message :

- `contentStart`- Définit le rôle et la configuration du message

```
{
  "event": {
    "contentStart": {
      "promptName": "<prompt-id>",
      "contentName": "<content-id>",
      "type": "TEXT",
      "interactive": true,
      "role": "ASSISTANT",
      "textInputConfiguration": {
        "mediaType": "text/plain"
      }
    }
  }
}
```

- `textInput`- Contient le contenu réel du message. Un `TextInput` ne peut pas dépasser 1 Ko. Si tel est le cas, divisez-le en plusieurs entrées de texte dans le même bloc de contenu. Si la conversation est supérieure à 40 Ko, réduisez l'historique global des discussions.

```
{
  "event": {
    "textInput": {
      "promptName": "<prompt-id>",
      "contentName": "<content-id>",
      "content": "Take your time, Don. I'll be here when you're ready."
    }
  }
}
```

- `contentEnd`- Marque la fin du message

```
{
  "event": {
    "contentEnd": {
      "promptName": "<prompt-id>",
      "contentName": "<content-id>"
    }
  }
}
```

```
}  
}  
}
```

Répétez ces trois événements pour chaque message de l'historique de vos discussions, en alternant les rôles UTILISATEUR et ASSISTANT.

Considérations importantes :

- L'historique des discussions ne peut être inclus qu'une seule fois par session
- L'historique des discussions doit être envoyé après l'invite du système et avant le début de la diffusion audio
- Tous les messages historiques doivent être envoyés avant de démarrer le streaming audio
- Chaque message doit spécifier le rôle UTILISATEUR ou ASSISTANT
- Utiliser le contenu de transcription stocké à partir des événements TextOutput comme valeur de contenu dans TextInput

Réception de transcriptions ASR

Au cours d'une conversation, Amazon Nova 2 Sonic envoie des transcriptions ASR via des événements de sortie. Chaque transcription est fournie sous la forme d'une séquence de trois événements : ContentStart, TextOutput et ContentEnd.

Exemple : transcription vocale d'un utilisateur :

1. ContentStart : indique le début d'une transcription :

```
{  
  "event": {  
    "contentStart": {  
      "additionalModelFields": "{\"generationStage\":\"FINAL\"}",  
      "completionId": "<completion-id>",  
      "contentId": "<content-id>",  
      "promptName": "<prompt-id>",  
      "role": "USER",  
      "sessionId": "<session-id>",  
      "textOutputConfiguration": {  
        "mediaType": "text/plain"  
      },  
    },  
  },  
}
```

```
    "type": "TEXT"
  }
}
```

2. TextOutput : contient le contenu réel de la transcription :

```
{
  "event": {
    "textOutput": {
      "completionId": "<completion-id>",
      "content": "hello how are you",
      "contentId": "<content-id>",
      "promptName": "<prompt-id>",
      "role": "USER",
      "sessionId": "<session-id>"
    }
  }
}
```

3. ContentEnd : marque la fin de la transcription :

```
{
  "event": {
    "contentEnd": {
      "completionId": "<completion-id>",
      "contentId": "<content-id>",
      "promptName": "<prompt-id>",
      "sessionId": "<session-id>",
      "stopReason": "PARTIAL_TURN",
      "type": "TEXT"
    }
  }
}
```

Le même schéma à trois événements s'applique aux rôles UTILISATEUR et ASSISTANT. Extrayez le content champ de l'eventTextOutput et le role champ de l'eventContentStart pour créer l'historique de vos discussions.

Bonnes pratiques

Enregistrez toujours l'historique des discussions pour activer :

- Reprises de session sur différents appareils
- Enregistrement et audit des conversations
- Préservation du contexte pour les interactions de suivi

Important : lors de la sauvegarde de l'historique des discussions, utilisez des sorties de texte en fonction de leur `GenerationStage` :

- Speculative - Un aperçu de ce que Nova 2 Sonic prévoit de dire, généré avant le début de la synthèse audio
- Finale - La transcription réelle au niveau de la phrase de ce qui a été dit dans la réponse audio

Enregistrez toujours le texte final dans votre historique de discussion, car il représente l'enregistrement précis de la conversation.

Exemple de sortie FINALE (enregistrez-le dans l'historique des discussions) :

```
ContentStart event: {
  "additionalModelFields": "{\"generationStage\":\"FINAL\"}",
  "completionId": "<completion-id>",
  "contentId": "<content-id>",
  "role": "ASSISTANT",
  "sessionId": "<session-id>",
  "type": "TEXT"
}
```

Exemple de sortie SPECULATIVE (aperçu facultatif, pas pour l'historique) :

```
ContentStart event: {
  "additionalModelFields": "{\"generationStage\":\"SPECULATIVE\"}",
  "completionId": "<completion-id>",
  "contentId": "<content-id>",
  "role": "ASSISTANT",
  "sessionId": "<session-id>",
  "type": "TEXT"
}
```

Configuration de l'outil

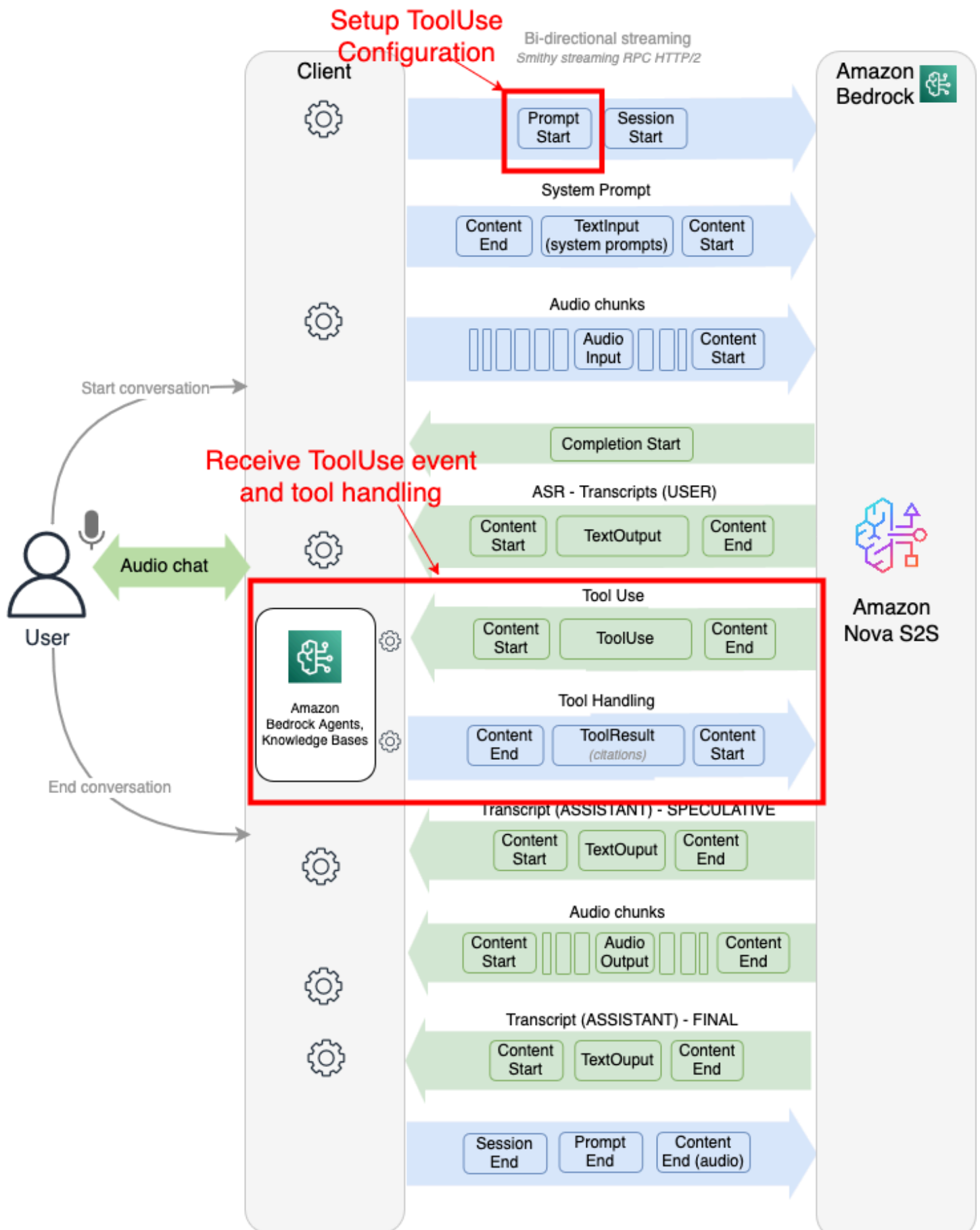
Amazon Nova 2 Sonic prend en charge l'utilisation d'outils (également appelés appels de fonctions), ce qui permet au modèle de demander des informations ou des actions externes lors de conversations, telles que des appels d'API, des requêtes de base de données ou des fonctions de code personnalisées. Cela permet à votre assistant vocal d'effectuer des actions, de récupérer des informations et de s'intégrer à des services externes en fonction des demandes des utilisateurs.

Nova 2 Sonic propose des appels d'outils asynchrones, permettant à l'IA de continuer à converser naturellement pendant que les outils s'exécutent en arrière-plan, créant ainsi une expérience utilisateur plus fluide et réactive.

Voici des étapes simplifiées relatives à l'utilisation des outils :

1. Définition des outils : spécifiez les outils disponibles avec leurs paramètres dans l'événement `PromptStart`
2. L'utilisateur parle : l'utilisateur fait une demande qui nécessite un outil (tel que « Quel temps fait-il à Seattle ? »)
3. Invocation d'outil : Nova 2 Sonic reconnaît le besoin et envoie un événement `ToolUse`
4. Exécuter un outil : votre application exécute l'outil et renvoie les résultats
5. Génération de réponses : Nova 2 Sonic intègre les résultats dans sa réponse vocale

Le schéma suivant illustre le fonctionnement de l'outil :



Outils de définition

Les outils sont définis à l'aide d'un schéma JSON qui décrit leur objectif, leurs paramètres et les entrées attendues.

Les composants et les explications de la définition de l'outil sont les suivants :

- Nom : identifiant unique de l'outil (utilisez snake_case)
- Description : Explication claire de ce que fait l'outil ; aide l'IA à décider quand l'utiliser
- InputSchema: schéma JSON définissant les paramètres acceptés par l'outil
- Propriétés : paramètres individuels avec types et descriptions
- Obligatoire : tableau de noms de paramètres à fournir

Exemple de définition d'outil

Voici une définition simple de l'outil météo

```
{
  "toolSpec": {
    "name": "get_weather",
    "description": "Get current weather information for a specific location",
    "inputSchema": {
      "json": {
        "type": "object",
        "properties": {
          "location": {
            "type": "string",
            "description": "City name or zip code"
          },
          "units": {
            "type": "string",
            "enum": ["celsius", "fahrenheit"],
            "description": "Temperature units"
          }
        },
        "required": ["location"]
      }
    }
  }
}
```

Configuration des outils dans PromptStart

La configuration de l'outil est transmise à Nova 2 Sonic lors de l'promptStart événement, ainsi que les paramètres de sortie audio et texte :

```
{
  "event": {
    "promptStart": {
      "promptName": "<prompt-id>",
      "textOutputConfiguration": {
        "mediaType": "text/plain"
      },
      "audioOutputConfiguration": {
        "mediaType": "audio/lpcm",
        "sampleRateHertz": 16000,
        "sampleSizeBits": 16,
        "channelCount": 1,
        "voiceId": "matthew",
        "encoding": "base64",
        "audioType": "SPEECH"
      },
      "toolUseOutputConfiguration": {
        "mediaType": "application/json"
      },
      "toolConfiguration": {
        "tools": [
          {
            "toolSpec": {
              "name": "get_weather",
              "description": "Get current weather information for a
specific location",
              "inputSchema": {
                "json": {
                  "type": "object",
                  "properties": {
                    "location": {
                      "type": "string",
                      "description": "City name or zip code"
                    },
                    "units": {
                      "type": "string",
                      "enum": ["celsius", "fahrenheit"],
                      "description": "Temperature units"
                    }
                  }
                }
              }
            }
          ]
        }
      }
    }
  }
}
```



```
}
    },
    "required": ["location"]
}
}
}
}
},
    "toolChoice": {
        "auto": {}
    }
}
}
}
```

Paramètres de choix de l'outil

Nova 2 Sonic prend en charge trois paramètres de choix d'outils pour contrôler quand et quels outils sont utilisés. Spécifiez le paramètre ToolChoice dans la configuration de votre outil :

- Auto (par défaut) : le modèle décide si des outils sont nécessaires et peut appeler plusieurs outils si nécessaire. Offre une flexibilité maximale.
- N'importe lequel : garantit qu'au moins un des outils disponibles est appelé au début de la réponse, le modèle sélectionnant le plus approprié. Utile lorsque vous disposez de plusieurs bases de connaissances ou outils et que vous souhaitez garantir qu'ils sont utilisés.
- Outil : force un outil nommé spécifique à être appelé exactement une fois au début de la réponse. Par exemple, si vous spécifiez un outil de base de connaissances, le modèle l'interrogera avant de répondre, qu'il pense que l'outil est nécessaire ou non.

Exemples de choix d'outils

Auto (par défaut)

```
"toolChoice": {
  # "auto": {}
}
```

N'importe lequel :

```
"toolChoice": {
```

```
"any": {}  
}
```

Outil spécifique :

```
"toolChoice": {  
  "tool": {  
    "name": "get_weather"  
  }  
}
```

Événements d'utilisation des outils de réception et de traitement

Lorsqu'Amazon Nova 2 Sonic détermine qu'un outil est nécessaire, il envoie un `toolUse` événement contenant :

1. `toolUseID`: identifiant unique pour l'invocation de cet outil
2. `ToolName`: le nom de l'outil à exécuter
3. `Contenu` : chaîne JSON contenant les paramètres extraits de la demande de l'utilisateur
4. `SessionId` : identifiant de session en cours
5. `Rôle` : défini sur « TOOL » pour les événements liés à l'utilisation de l'outil

Exemple d'événement d'utilisation d'un outil

```
{  
  "event": {  
    "toolUse": {  
      "completionId": "<completion-id>",  
      "content": "{\n\"location\": \"Seattle\", \n\"units\": \"fahrenheit\"}",  
      "contentId": "<content-id>",  
      "promptName": "<prompt-id>",  
      "role": "TOOL",  
      "sessionId": "<session-id>",  
      "toolName": "get_weather",  
      "toolUseId": "<tool-use-id>"  
    }  
  }  
}
```

Étapes de traitement

1. Recevez l'événement ToolUse de Nova 2 Sonic
2. Extraire le nom et les paramètres de l'outil à partir de l'événement
3. Exécutez la logique de votre outil (appel d'API, requête de base de données, etc.)
4. Renvoie le résultat à l'aide d'un événement ToolResult

Exemple d' ToolResult événement

```
{
  "event": {
    "toolResult": {
      "promptName": "<prompt-id>",
      "contentName": "<content-id>",
      "content": "{ \"temperature\": 72, \"condition\": \"sunny\", \"humidity\": 45 }"
    }
  }
}
```

Bonnes pratiques

- Descriptions claires : Rédigez des descriptions détaillées des outils pour aider Nova 2 Sonic à comprendre quand utiliser chaque outil.
- Valider les paramètres : validez toujours les paramètres de l'outil avant l'exécution pour éviter les erreurs. Définissez les paramètres de l'outil à l'aide d'un schéma JSON approprié avec des types de données structurés (tels que des énumérations, des nombres ou des booléens) plutôt que des chaînes ouvertes dans la mesure du possible.
- Gestion des erreurs : renvoie des messages d'erreur significatifs dans les événements ToolResult en cas de défaillance des outils.
- Exécution asynchrone : profitez des appels d'outils asynchrones pour maintenir le flux de conversation.
- Dénomination des outils : utilisez des noms descriptifs orientés vers l'action (tels que get_weather, search_database, send_email).

Appel d'outils asynchrone

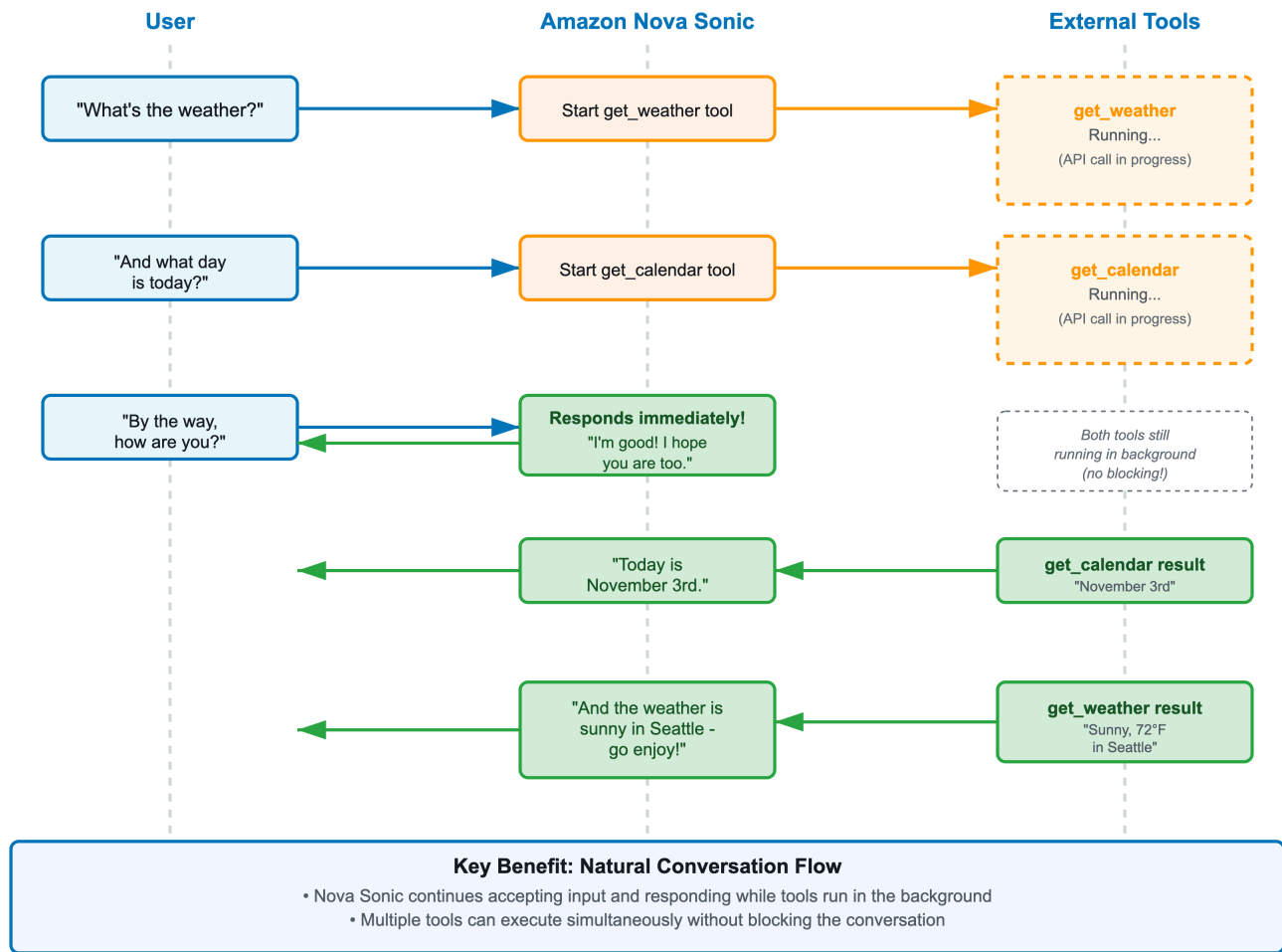
Contrairement aux appels d'outils synchrones traditionnels où l'IA attend silencieusement les résultats de l'outil, l'approche asynchrone d'Amazon Nova 2 Sonic lui permet de :

- Continuer à accepter les entrées de l'utilisateur pendant que les outils sont en cours d'exécution
- Répondez aux nouvelles questions sans attendre les résultats de l'outil en attente
- Gérez plusieurs appels d'outils simultanément
- Maintenez un flux de conversation naturel sans pauses gênantes
- Aucune configuration supplémentaire n'est requise. L'appel d'outils asynchrone fonctionne immédiatement.

Comment ça marche

Lorsque Nova 2 Sonic lance un appel d'outil, la conversation n'est pas interrompue. Au lieu de cela, il continue à écouter et à répondre naturellement jusqu'à ce que l'outil arrive.

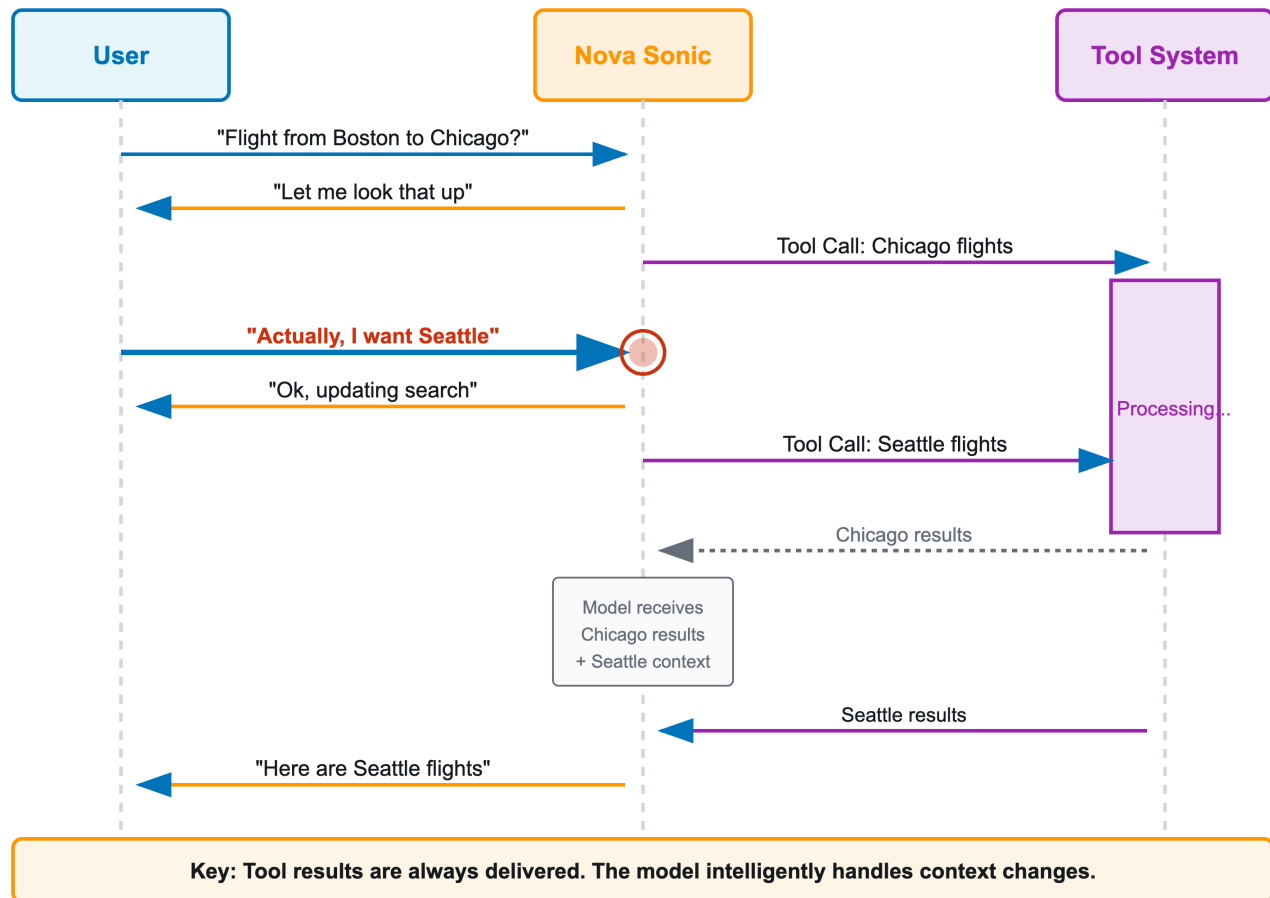
Asynchronous Tool Calling Flow



Gestion des interruptions chez les utilisateurs

Si un utilisateur modifie sa demande pendant l'exécution d'un outil, Nova 2 Sonic la gère intelligemment sans annuler les appels d'outils en attente.

Async Tool Calling with User Interruption



Exemple de scénario

```

User: "Can I book a flight from Boston to Chicago?"
Agent: "Sure, let me look that up for you."
Agent: [initiates tool call for Chicago flights]
User: "Actually, I want to go to Seattle"
Agent: "Ok let me update that search"
Agent: [initiates tool call for Seattle flights]
[First tool returns with Chicago flight results]
Agent: [receives Chicago results and processes them contextually]
  
```

Comment ça marche

Les résultats de l'outil sont toujours fournis : lorsqu'un appel d'outil est terminé, son résultat est toujours envoyé au modèle, même si l'utilisateur a modifié sa demande. Le modèle utilise ses capacités de raisonnement pour déterminer comment traiter les informations.

Traitement sensible au contexte : le modèle comprend le contexte de la conversation et peut gérer de manière appropriée les résultats d'outils obsolètes. Par exemple :

- Si l'utilisateur dit « merci » après avoir changé d'avis, le modèle a toujours besoin des résultats originaux pour le contexte
- Si l'utilisateur modifie sa demande, le modèle peut accuser réception des résultats d'origine tout en se concentrant sur la nouvelle demande

Aucune annulation automatique : le système n'annule ni n'ignore automatiquement les appels aux outils en fonction des nouvelles saisies par l'utilisateur. Cela garantit que le modèle dispose d'informations complètes pour prendre des décisions intelligentes sur la manière de réagir.

Intégrations

Amazon Nova 2 Sonic peut être intégré à différents frameworks et plateformes pour créer des applications d'IA conversationnelle. Ces intégrations fournissent des composants prédéfinis et simplifiés APIs pour les cas d'utilisation courants.

Agents à mèches

Strands Agents est un SDK simple mais puissant qui adopte une approche basée sur un modèle pour créer et exécuter des agents d'IA. Des simples assistants conversationnels aux flux de travail autonomes complexes, du développement local au déploiement en production, Strands Agents s'adapte à vos besoins.

Pour une documentation complète sur le framework Strands, consultez la [documentation officielle de Strands](#).

The Strands BidiAgent fournit une interaction audio et textuelle en temps réel grâce à des connexions de streaming persistantes. Contrairement aux modèles traditionnels de demande-réponse, cet agent entretient des conversations de longue durée en prenant en charge les interruptions, le traitement simultané et les réponses audio continues.

Prérequis :

- Python 3.8 ou version ultérieure installé
- Informations d'identification pour les AWS personnes configurées avec accès à Amazon Bedrock
- Connaissance de base de la syntaxe Python async/await

Exemple de code :

Exemple de code

Installation:

Installez les packages requis :

```
pip install strands-agents strands-agents-tools
```

Exécutez cet exemple :

```
import asyncio
from strands.experimental.bidi.agent import BidiAgent
from strands.experimental.bidi.io.audio import BidiAudioIO
from strands.experimental.bidi.io.text import BidiTextIO
from strands.experimental.bidi.models.novasonic import BidiNovaSonicModel
from strands_tools import calculator

async def main():
    """Test the BidirectionalAgent API."""
    # Audio and Text input/output utility
    audio_io = BidiAudioIO(audio_config={})
    text_io = BidiTextIO()

    # Nova Sonic model
    model = BidiNovaSonicModel(region="us-east-1")

    async with BidiAgent(model=model, tools=[calculator]) as agent:
        print("New BidiAgent Experience")
        print("Try asking: 'What is 25 times 8?' or 'Calculate the square root of 144'")

        await agent.run(
            inputs=[audio_io.input()],
```



```

        outputs=[audio_io.output(), text_io.output()]
    )

if __name__ == "__main__":
    try:
        asyncio.run(main())
    except KeyboardInterrupt:
        print("\nConversation ended by user")
    except Exception as e:
        print(f"Error: {e}")
        import traceback
        traceback.print_exc()

```

1. Importation : modèles requis

```

from strands.experimental.bidi.agent import BidiAgent
from strands.experimental.bidi.io.audio import BidiAudioIO
from strands.experimental.bidi.io.text import BidiTextIO
from strands.experimental.bidi.models.novasonic import BidiNovaSonicModel
from strands_tools import calculator

```

- **BidiAgent**: la principale classe d'agent qui orchestre les conversations bidirectionnelles
- **BidiAudioIO** : gère les entrées et sorties audio pour les interactions vocales
- **BidiTextIO** : fournit une sortie de texte pour les transcriptions et les réponses
- **BidiNovaSonicModel** L'emballage du modèle Nova 2 Sonic
- **Calculateur** : un outil prédéfini pour les opérations mathématiques

2. Configuration des E/S audio et texte

```

audio_io = BidiAudioIO(audio_config={})
text_io = BidiTextIO()

```

L' **BidiAudioIO** gère l'entrée microphone et la sortie haut-parleur, tandis que l' **BidiTextIO** affiche les transcriptions de texte et les réponses dans la console.

3. Initialiser le modèle

```

model = BidiNovaSonicModel(region="us-east-1")

```

Créez une instance de modèle Nova Sonic. Le paramètre `region` indique la AWS région dans laquelle le modèle est déployé.

4. Création et exécution de l'agent

```
async with BidiAgent(model=model, tools=[calculator]) as agent:
    await agent.run(
        inputs=[audio_io.input()],
        outputs=[audio_io.output(), text_io.output()]
    )
```

L'agent est créé avec :

- Modèle : Le modèle Nova 2 Sonic à utiliser
- Outils : liste des outils que l'agent peut appeler (comme une calculatrice)
- Entrées : entrée audio depuis le microphone
- Sorties : sortie audio vers les haut-parleurs et sortie texte vers la console

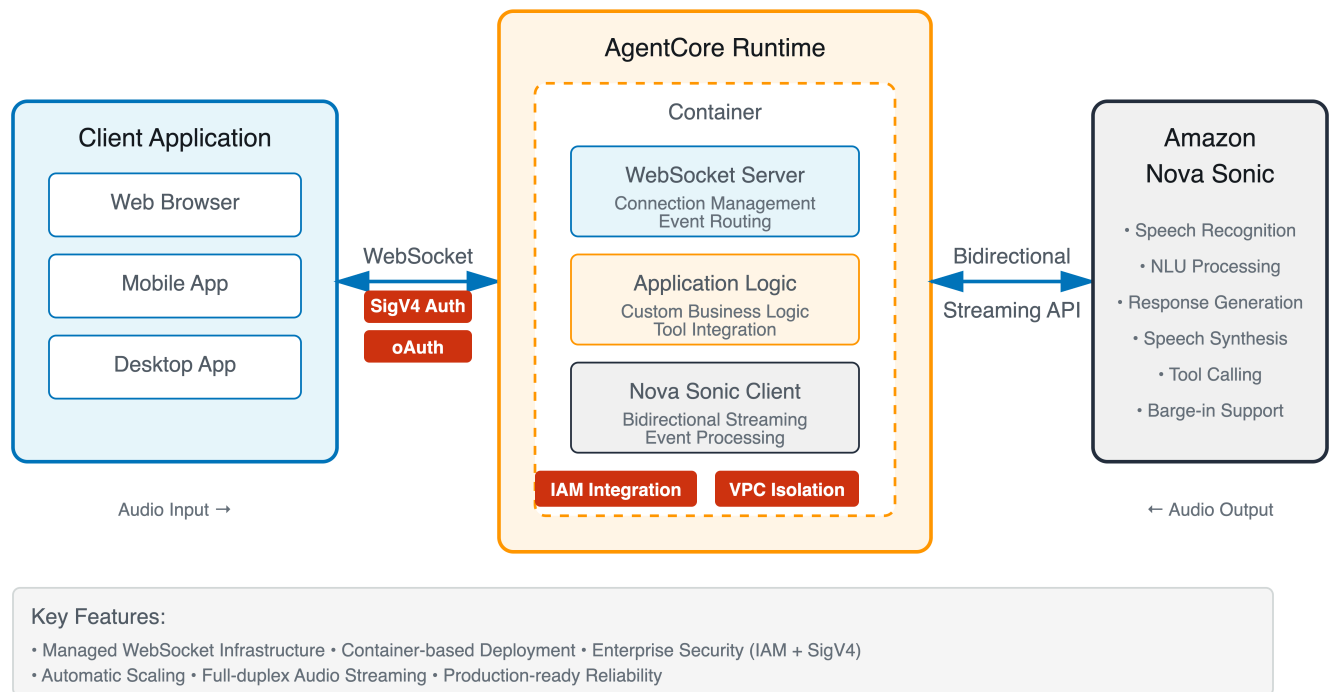
Intégrations du framework

Amazon Nova 2 Sonic peut être intégré à différents frameworks et plateformes pour créer des applications vocales sophistiquées. Les exemples suivants illustrent les modèles d'intégration avec les frameworks les plus courants.

Amazon Bedrock AgentCore

Amazon Bedrock AgentCore fournit un environnement d'exécution géré pour déployer les applications Nova 2 Sonic avec une sécurité et une évolutivité de niveau professionnel. AgentCore simplifie le déploiement d'applications d'IA vocale en temps réel en gérant l'infrastructure, l'authentification et WebSocket la connectivité.

AgentCore Architecture



Caractéristiques principales :

- Streaming bidirectionnel - Support natif de l'interface de streaming en duplex intégral de Nova Sonic avec traitement des événements en temps réel et communication à faible latence.
- WebSocket infrastructure : WebSocket serveurs prêts pour la production avec mise à l'échelle automatique, gestion des connexions et récupération des erreurs.
- Déploiement de conteneurs - Déployez les applications Nova Sonic sous forme de conteneurs sur une infrastructure gérée avec une mise à l'échelle horizontale et un versionnement indépendant.
- Sécurité d'entreprise : authentification précise via IAM et SigV4, isolation VPC et journalisation complète des audits.

L'architecture montre comment les applications clientes se connectent à AgentCore Runtime via WebSocket l'authentification SigV4. L'environnement conteneurisé inclut votre WebSocket serveur, la logique de l'application et le client Nova Sonic, qui communiquent tous avec Nova Sonic via l'API de diffusion bidirectionnelle.

Avantages :

- Opérations simplifiées : concentrez-vous sur la logique des applications tout en AgentCore gérant l'infrastructure, l'évolutivité et la fiabilité.
- Sécurité d'entreprise : fonctionnalités intégrées d'authentification, d'autorisation et de conformité pour les déploiements de production.
- Rentabilité : payez uniquement pour ce que vous utilisez grâce à la mise à l'échelle automatique et à l'optimisation des ressources.
- Productivité des développeurs : réduisez les délais de production grâce à une WebSocket infrastructure gérée et au déploiement de conteneurs.

Cas d'utilisation

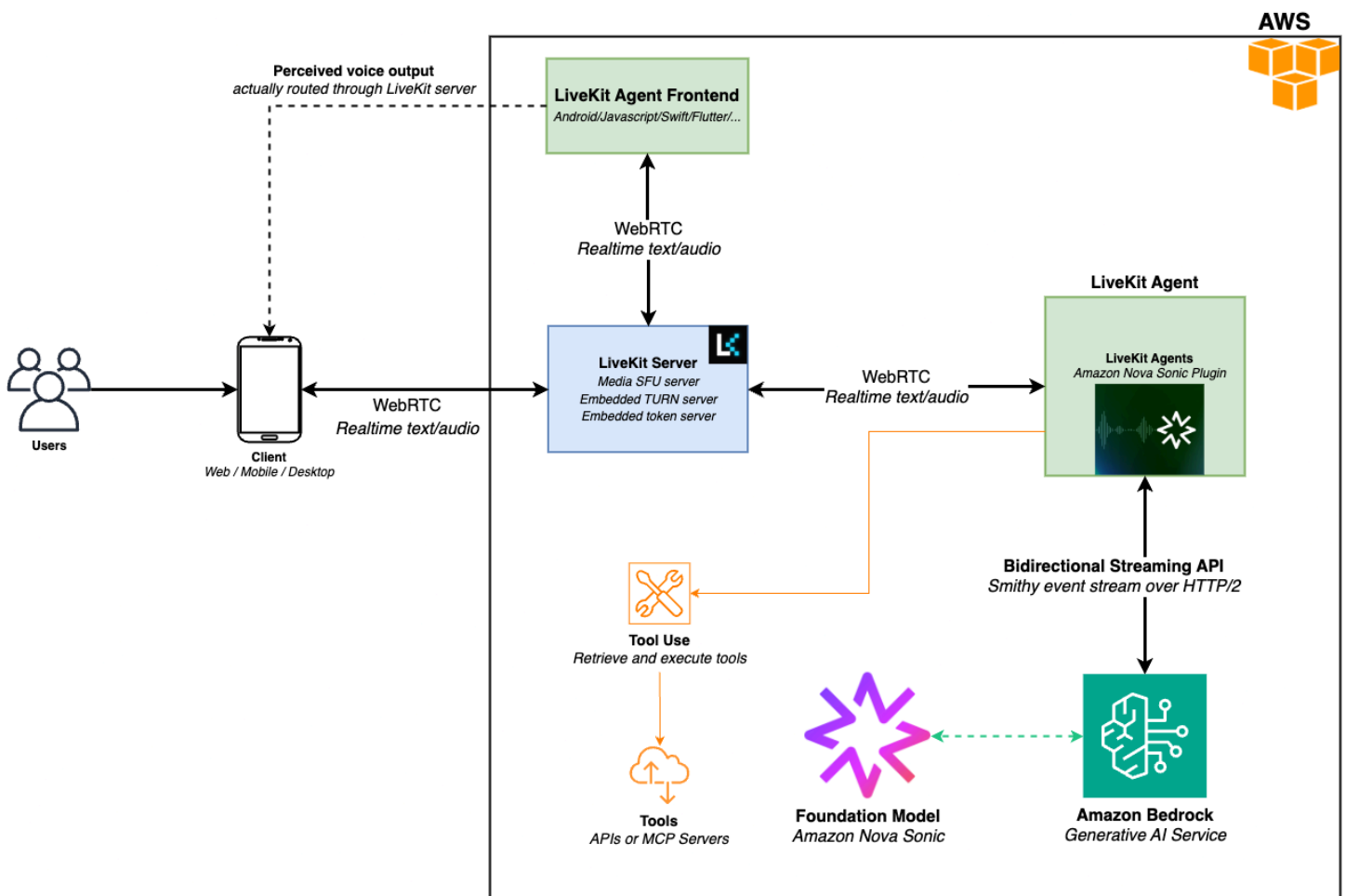
- Assistants vocaux du service client avec authentification sécurisée
- Applications vocales d'entreprise nécessitant une intégration IAM
- Plateformes vocales multi-locataires avec déploiements isolés
- Applications à commande vocale nécessitant des pistes de conformité et d'audit

Pour obtenir une documentation détaillée sur le déploiement de Nova Sonic avec AgentCore, consultez la [AgentCore documentation Amazon Bedrock](#).

LiveKit

LiveKit est une plateforme open source permettant de créer des applications audio et vidéo en temps réel. L'intégration avec Amazon Nova 2 Sonic permet aux développeurs de créer des interfaces vocales conversationnelles sans gérer de pipelines audio ou de protocoles de signalisation complexes.

Pour des exemples d'implémentation détaillés et des exemples de code, consultez la [documentation LiveKit AWS d'intégration](#).



Comment cela fonctionne :

- **Couche client** : les applications Web, mobiles ou de bureau se connectent à l'aide LiveKit du client SDKs, qui gère la capture audio, le streaming WebRTC et la lecture.
- **LiveKit Serveur** : agit en tant que centre de communication en temps réel, gérant les connexions WebRTC, acheminant les flux audio et gérant l'état des sessions avec une optimisation à faible latence.
- **LiveKit Agent** : agent basé sur Python qui reçoit le son du serveur, le traite via le plugin Nova Sonic et renvoie les réponses en continu. Comprend des fonctionnalités intégrées telles que la détection de l'activité vocale et la gestion des virages.
- **Amazon Nova 2 Sonic** : traite le flux audio via une API de diffusion bidirectionnelle, effectue la reconnaissance vocale, la compréhension du langage naturel et génère des réponses conversationnelles à l'aide de la synthèse vocale.

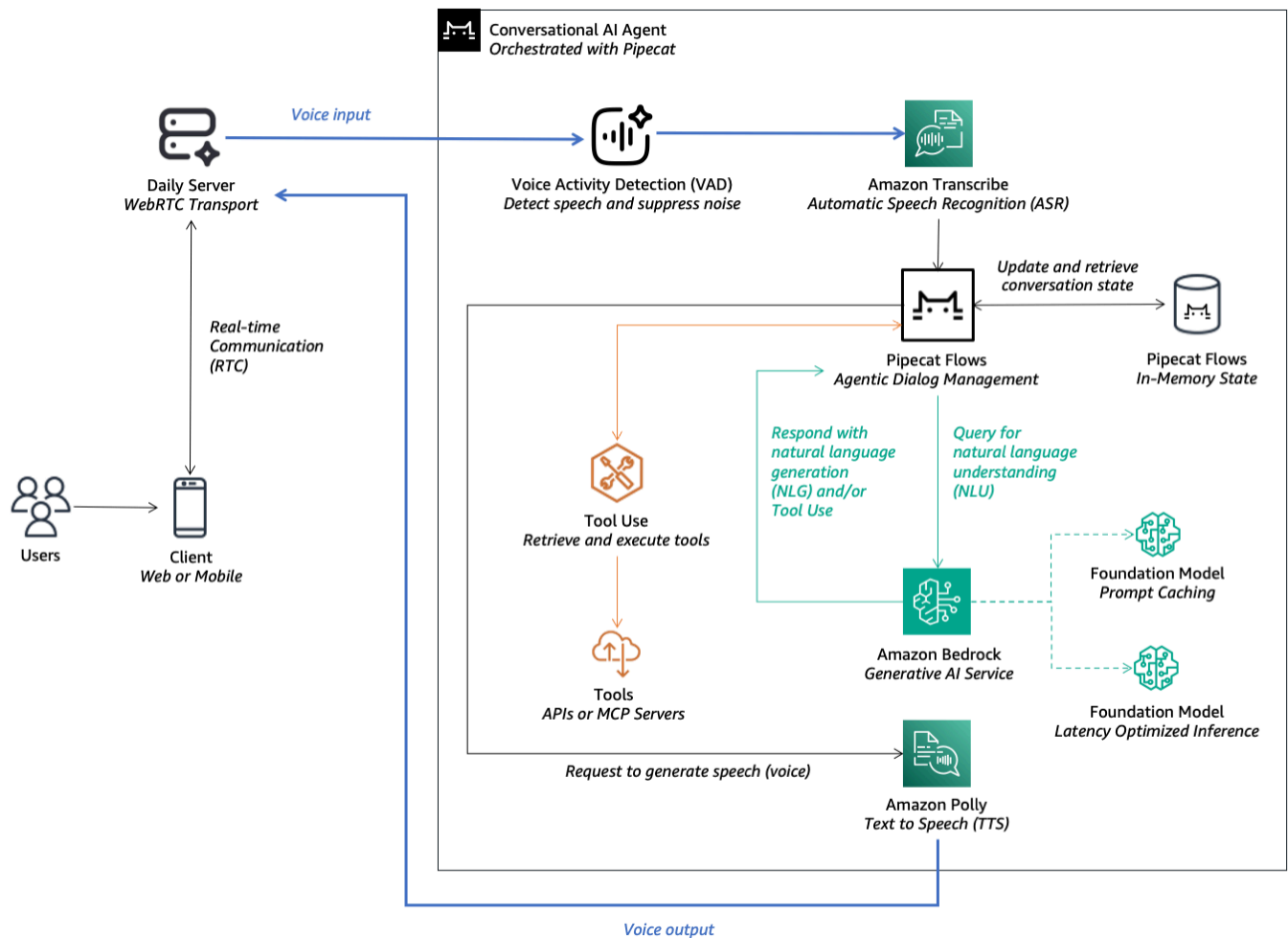
Pipecat

Pipecat est un framework permettant de créer des applications vocales et d'intelligence artificielle conversationnelle multimodale. Il fournit une architecture modulaire basée sur un pipeline qui orchestre plusieurs composants pour créer des applications vocales intelligentes avec Amazon Nova Sonic et d'autres services. AWS

Pour des exemples d'implémentation détaillés et des exemples de code, consultez la [documentation PipeCat AWS d'intégration](#).

Caractéristiques principales :

- Architecture de pipeline : framework modulaire basé sur Python pour composer des composants d'IA vocale, notamment ASR, NLU, TTS, etc.
- Pipecat flows : cadre de gestion des états pour la création d'une logique conversationnelle complexe et l'exécution d'outils.
- Support WebRTC : intégration intégrée avec Daily et d'autres fournisseurs WebRTC pour le streaming audio en temps réel.
- AWS Intégration : support natif pour Amazon Bedrock, Amazon Transcribe et Amazon Polly.



L'architecture inclut :

- Transport WebRTC : diffusion audio en temps réel entre les appareils clients et le serveur d'applications.
- Détection d'activité vocale (VAD) : Silero VAD avec détection vocale et suppression du bruit configurables.
- Reconnaissance vocale : Amazon Transcribe pour une conversion précise en temps réel speech-to-text.
- Compréhension du langage naturel : Amazon Nova Pro sur Bedrock avec inférence optimisée pour la latence.
- Exécution de l'outil : Pipecat Flows pour l'intégration des API et les appels de service principal.
- Génération de réponses : Amazon Nova Pro pour des réponses cohérentes et contextuelles.
- Text-to-speech : Amazon Polly avec des voix génératives pour une sortie vocale réaliste.

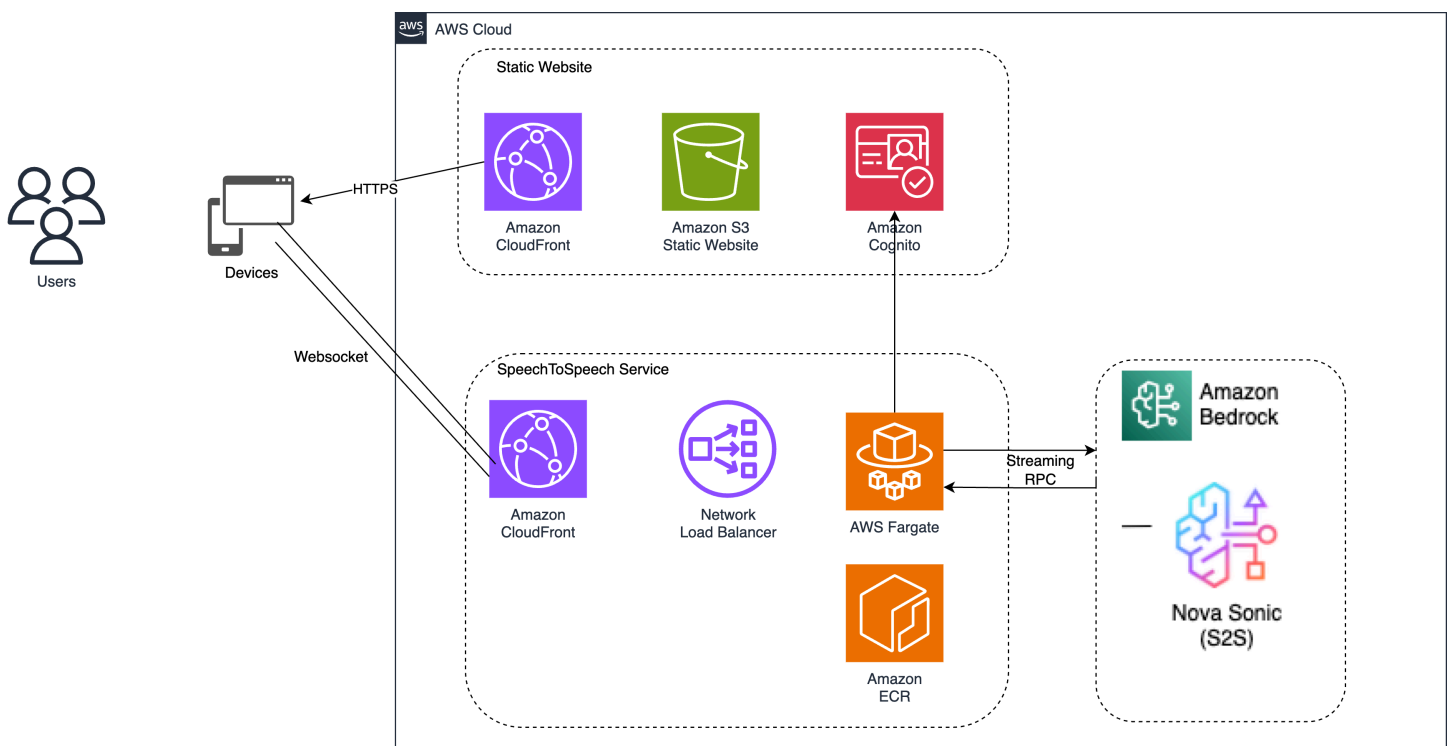
Déployez sur AWS

Déployez vos applications Nova Sonic en AWS utilisant l'infrastructure sous forme de code avec le AWS CDK (Cloud Development Kit). Cette approche fournit des déploiements reproductibles et contrôlés par version avec les meilleures pratiques intégrées.

Options de déploiement

- Amazon ECS (Elastic Container Service) : orchestration de conteneurs entièrement gérée avec intégration d'Application Load Balancer, auto-scaling et exécution Fargate sans serveur.
- Amazon EKS (Elastic Kubernetes Services) : Kubernetes gérés pour une orchestration complexe, une mise en réseau avancée, des déploiements multirégionaux et un écosystème d'outils étendu.
- AWS CDK : AWS CDK vous permet de définir une infrastructure cloud à l'aide de langages de programmation familiers.

Pour un exemple complet et prêt pour la production de déploiement de Nova Sonic avec AWS CDK, consultez l'exemple de [Speech-to-Speech CDK](#) sur GitHub. Ce modèle illustre :



- Configuration complète de l'infrastructure CDK avec TypeScript
- WebSocket implémentation du serveur pour une communication en temps réel
- Déploiement de conteneurs avec ECS et Fargate

- Configuration de l'Application Load Balancer à des fins d'assistance WebSocket
- Configuration du réseau VPC et du groupe de sécurité
- CloudWatch surveillance et journalisation
- Bonnes pratiques pour les déploiements en production

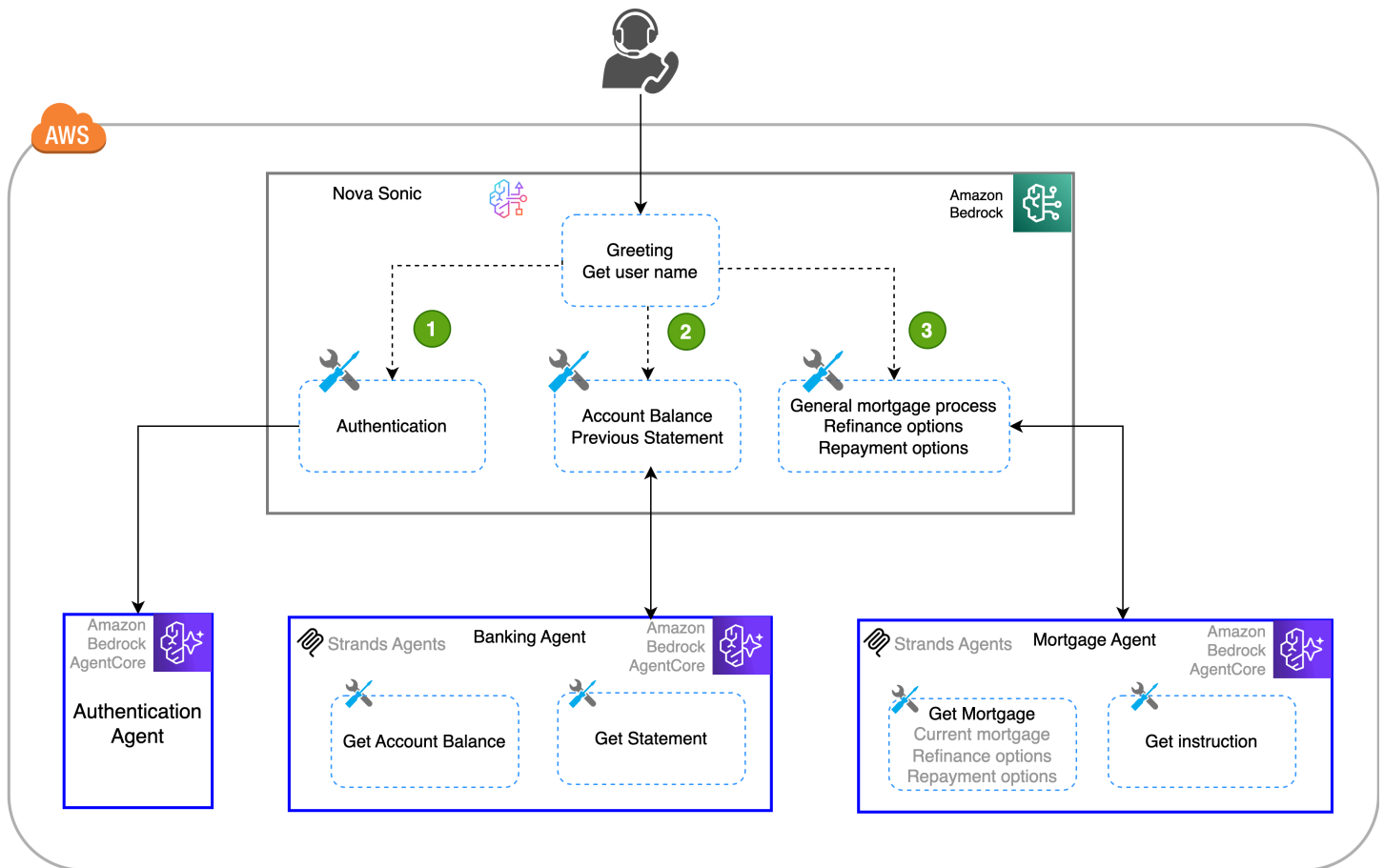
Systèmes multi-agentiques

L'architecture multi-agents est un modèle largement utilisé pour concevoir des assistants IA qui gèrent des tâches complexes. Dans un assistant vocal alimenté par Nova 2 Sonic, cette architecture coordonne plusieurs agents spécialisés, où chaque agent fonctionne indépendamment pour permettre un traitement parallèle, une conception modulaire et des solutions évolutives.

Nova Sonic joue le rôle d'orchestrateur dans un système multi-agents, remplissant deux fonctions clés :

Gestion du flux de conversation : veille à ce que toutes les informations nécessaires soient collectées avant de passer à l'étape suivante de la conversation.

Classification des intentions : analyse les demandes des utilisateurs et les achemine vers le sous-agent spécialisé approprié.



Le schéma ci-dessus montre un assistant vocal bancaire utilisant une architecture multi-agents. Le flux de conversation commence par un message d'accueil et la collecte du nom de l'utilisateur, puis traite les demandes liées aux opérations bancaires ou aux prêts hypothécaires par l'intermédiaire de sous-agents spécialisés.

Exemple de flux de conversation :

1. L'utilisateur se connecte à l'assistant vocal.
2. Nova 2 Sonic : « Bonjour ! Quel est ton nom ? »
3. Utilisateur : « Je m'appelle John »
4. Nova 2 Sonic : « Bonjour John, en quoi puis-je t'aider aujourd'hui ? »
5. Utilisateur : « Je souhaite vérifier le solde de mon compte »
6. Nova 2 Sonic : [Routes vers l'agent d'authentification]
7. Agent d'authentification : « Veuillez fournir votre identifiant de compte »
8. Utilisateur : « 12345 »

9. Agent d'authentification : [Vérifie l'identité]

10Nova 2 Sonic : [Itinéraires vers l'agent bancaire]

11Agent bancaire : « Votre solde actuel est de 5 431 110\$ »

Bien que cet exemple montre des sous-agents utilisant le framework Strands Agents déployé sur Amazon Bedrock AgentCore, l'architecture est flexible. Vous pouvez choisir :

- Votre framework d'agent préféré
- Tout fournisseur de LLM
- Options d'hébergement personnalisées
- Différents modèles d'orchestration

Avantages :

- Modularité : chaque agent se concentre sur un domaine spécifique, ce qui facilite la maintenance et la mise à jour du système.
- Évolutivité : ajoutez de nouveaux agents sans modifier les agents existants, ce qui permet à votre système de s'adapter à vos besoins.
- Traitement parallèle : plusieurs agents peuvent travailler simultanément, ce qui améliore les temps de réponse aux requêtes complexes.
- Spécialisation : Chaque agent peut être optimisé pour sa tâche spécifique, en utilisant les outils et les bases de connaissances les plus appropriés.
- Isolation des défaillances : si un agent tombe en panne, les autres continuent de fonctionner, ce qui améliore la fiabilité globale du système.

Consultez [ce blog](#) pour plus de détails et des exemples de code.

Consultez le [laboratoire multi-agents du Nova Sonic Workshop](#) pour des échantillons pratiques.

Intégration de la téléphonie

Amazon Nova 2 Sonic s'intègre aux fournisseurs de téléphonie pour permettre aux applications vocales basées sur l'IA d'être accessibles via des appels téléphoniques. Ce guide couvre l'intégration avec Twilio, Vonage et d'autres systèmes SIP pour créer des solutions de centre de contact et des agents vocaux.

Twilio : plateforme de communication dans le cloud dotée de capacités de diffusion vocale et multimédia programmables.

Vonage : communications internationales APIs grâce à la voix, au streaming WebSocket audio et à la connectivité SIP.

AWS fournit un exemple complet d'implémentation démontrant Nova Sonic dans un environnement de centre de contact avec analyse en temps réel et intégration de la téléphonie.

Référentiel : [Sample Sonic Contact Center avec téléphonie](#)

Systèmes avancés avec Amazon Nova

Présentation de

Cette section fournit des conseils complets aux utilisateurs expérimentés qui mettent en œuvre des modèles avancés avec les modèles Nova. Ces fonctionnalités permettent des applications sophistiquées, notamment l'intégration d'outils, la génération augmentée par extraction (RAG) et des agents d'intelligence artificielle autonomes.

Réflexion approfondie dans Amazon Nova 2

Amazon Nova 2 Lite introduit des capacités de réflexion étendues qui permettent au modèle d'engager un raisonnement plus approfondi pour des problèmes complexes. Cette fonctionnalité optionnelle vous permet de contrôler quand et comment le modèle alloue des ressources informatiques supplémentaires pour effectuer des tâches difficiles.

Comment fonctionne la pensée étendue

Amazon Nova 2 introduit la pensée étendue en tant que capacité hybride. Vous avez le contrôle total :

- Extended Thinking OFF (par défaut) : Amazon Nova 2 fonctionne avec un raisonnement latent efficace, optimal pour les tâches quotidiennes et les applications à volume élevé.
- Réflexion approfondie : Amazon Nova 2 adopte un step-by-step raisonnement explicite, idéal pour les problèmes complexes nécessitant une analyse approfondie.

Effort de raisonnement

Ce qui suit montre les différents niveaux d'effort de raisonnement.

Faible effort (MaxReasoningEffort : « faible »)

Idéal pour : les tâches plus complexes qui nécessitent une réflexion structurée. Par exemple, vous pouvez l'utiliser pour la révision du code et les suggestions d'amélioration lorsque le modèle doit prendre en compte attentivement la qualité du code existant, pour les tâches d'analyse qui nécessitent une prise en compte approfondie de plusieurs facteurs ou pour les scénarios de résolution de problèmes bénéficiant d'une approche méthodique. Le faible niveau d'effort est idéal

pour les tâches complexes où le raisonnement de base améliore la précision sans nécessiter une planification approfondie en plusieurs étapes.

Effort moyen (maxReasoningEffort« moyen »)

Idéal pour : les tâches en plusieurs étapes et les flux de travail de codage. Par exemple, vous pouvez l'utiliser pour le développement et le débogage de logiciels lorsque le modèle doit comprendre la structure de code existante avant d'implémenter des modifications, pour générer du code nécessitant une coordination entre plusieurs fichiers ou composants, pour des calculs en plusieurs étapes avec interdépendances ou pour planifier des tâches avec plusieurs contraintes. Un effort moyen est optimal pour les flux de travail agentiques qui coordonnent plusieurs outils et nécessitent que le modèle maintienne le contexte de plusieurs opérations séquentielles.

Effort élevé (effort de raisonnement maximal : « élevé »)

Idéal pour : le raisonnement STEM et la résolution avancée de problèmes. Par exemple, vous pouvez l'utiliser pour des problèmes mathématiques avancés et des preuves qui nécessitent une step-by-step vérification rigoureuse, des analyses scientifiques et des tâches de recherche exigeant une investigation approfondie, une conception de systèmes complexes avec des considérations architecturales à plusieurs dimensions, ou des scénarios de prise de décision critiques ayant des implications importantes. Un effort élevé permet d'obtenir une précision maximale pour les tâches nécessitant un raisonnement sophistiqué, une évaluation minutieuse des alternatives et une validation approfondie des conclusions.

Démarrage rapide : activation de la pensée étendue

La pensée étendue est contrôlée par le `reasoningConfig` paramètre.

```
import boto3

bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

response = bedrock.converse(
    modelId='us.amazon.nova-2-lite-v1:0',
    system=[{"text": "You are a highly capable personal assistant"}],
    messages=[{
        "role": "user",
        "content": [{"text": "Provide a meal plan for a gluten free family of 4."}]
    }],
    inferenceConfig={
        "temperature": 0.7,
        "topP": 0.9,
```

```

        "maxTokens": 10000
    },
    additionalModelRequestFields={
        "reasoningConfig": {
            "type": "enabled",
            "maxReasoningEffort": "low"
        }
    }
)

content_list = response["output"]["message"]["content"]

for item in content_list:
    if "reasoningContent" in item:
        reasoning_text = item["reasoningContent"]["reasoningText"]["text"]
        print("=== REASONING ===")
        print(reasoning_text)
        print()
    elif "text" in item:
        print("=== ANSWER ===")
        print(item["text"])

```

Paramètres de raisonnement :

- type: enabled ou disabled (par défaut :disabled)
- maxReasoningEffort: low, medium, ou high

Note

Temperature, TopP et TopK ne peuvent pas être utilisés lorsque le paramètre est maxReasoningEffort réglé sur. high L'utilisation conjointe de ces paramètres entraîne une erreur.

Pour des exemples complets de code utilisant la pensée étendue, voir [the section called “Bibliothèque de codes”](#).

Structure de réponse

Lorsque vous activez la pensée étendue, les réponses incluent des reasoningContent blocs suivis de blocs de text contenu :

```
{
  "output": {
    "message": {
      "role": "assistant",
      "content": [
        {
          "reasoningContent": {
            "reasoningText": {
              "text": "[REDACTED]"
            }
          }
        },
        {
          "text": "Based on the premises, we can conclude..."
        }
      ]
    }
  },
  "stopReason": "end_turn"
}
```

Note

Avec Amazon Nova 2, le contenu de raisonnement s'affiche sous la forme [REDACTED]. Les jetons de raisonnement vous sont toujours facturés, car ils contribuent à améliorer la qualité de sortie. Nous incluons maintenant ce champ dans la structure de réponse afin de préserver la possibilité d'exposer le contenu du raisonnement à l'avenir. Nous travaillons activement avec les clients afin de déterminer la meilleure approche pour définir le processus de raisonnement du modèle. Le contenu du raisonnement est affiché sous forme de. [REDACTED] Les jetons de raisonnement vous sont toujours facturés, car ils contribuent à améliorer la qualité des réponses.

Options de configuration

Amazon Nova 2 introduit un nouveau `reasoningConfig` paramètre que vous pouvez ajouter à votre structure de demande converse existante pour permettre le raisonnement :


```
additionalModelRequestFields={
  "reasoningConfig": {
    "type": "enabled", # or "disabled" (default)
    "maxReasoningEffort": "high" # "low", "medium", or "high"
  }
}
```

Paramètres :

- **type** : Basculer entre "enabled" et "disabled" (la valeur par défaut est "disabled")
- **maxReasoningEffort** : lorsque cette option est activée, contrôlez la profondeur du raisonnement
- « faible » : tâches modérément complexes
- « moyen » : problèmes complexes nécessitant une analyse approfondie
- « élevé » : raisonnement le plus complet pour des tâches très complexes

Note

lors de l'utilisation "high", temp, toPP et maxToken doivent être désactivés. Dans ce mode, le modèle effectue une analyse plus approfondie pour trouver la meilleure solution. Ce traitement plus approfondi peut générer un résultat supérieur à 65 000 jetons. Le montant exact dépend de la complexité de votre demande, mais pour certains problèmes, nous l'avons vu passer à 128 000 jetons. Cela vous garantit un raisonnement complet et de haute qualité plutôt que des résultats tronqués.

Modèles pris en charge

Extended Thinking est actuellement disponible dans : Amazon Nova 2 Lite (us.amazon.nova-2-lite-v 1:0)

Réflexion approfondie avec utilisation d'outils

La réflexion approfondie fonctionne parfaitement avec l'appel d'outils, ce qui permet à Amazon Nova de réfléchir aux outils à utiliser et à la manière d'interpréter leurs résultats.

Comprendre les jetons de raisonnement et les prix

Types de jetons

Les jetons Extended Thinking sont facturés comme des jetons de sortie :

- Jetons d'entrée : contenu de votre demande initiale (prix d'entrée standard)
- Jetons de sortie : cela inclut les jetons de raisonnement et le contenu de la réponse visible finale (tarification de sortie standard)

Répartition de l'utilisation

Les trois types de jetons sont inclus dans vos statistiques d'utilisation et dans votre facturation. Le prix des jetons de raisonnement est le même que celui des jetons de sortie et apparaîtront comme « SUPPRIMÉ » dans la réponse du modèle ».

```
{
  "usage": {
    "inputTokens": 45,
    "outputTokens": 1240,
    "totalTokens": 1285
  }
}
```

Questions fréquentes (FAQ)

Pourquoi Amazon Nova 2 Lite affiche-t-il « [SUPPRIMÉ] » pour le contenu du raisonnement au lieu d'afficher le processus de réflexion du modèle ?

Notre objectif principal pour ce lancement est de nous assurer que Nova 2 fournit la meilleure intelligence de sa catégorie pour vos tâches, et cela se reflète dans l'amélioration de la précision.

Nous sommes conscients que la visibilité du processus de raisonnement est précieuse, et nous avons constaté que les clients souhaitaient vivement comprendre comment le modèle aborde les problèmes.

Nous cherchons des moyens de le rendre disponible prochainement.

Les jetons de raisonnement vous sont toujours facturés, car ils représentent un travail réel qui améliore la qualité de votre sortie

qui seront capturés `outputTokens` avec les jetons de réponse.

Comment savoir si la pensée étendue fonctionne si le raisonnement est expurgé ?

Vous pouvez confirmer que la pensée étendue fonctionne en vérifiant la présence de `reasoningContent` blocs dans la réponse (ceux-ci n'apparaissent que lorsque le raisonnement est activé)

Utilisation d'outils (appel de fonctions)

Les outils étendent les fonctionnalités d'Amazon Nova en connectant le modèle à des fonctionnalités externes telles que APIs les bases de données et les environnements d'exécution de code.

L'utilisation d'outils permet à Amazon Nova d'accéder à des informations en temps réel, d'effectuer des calculs et d'interagir avec des systèmes externes.

Comprendre le flux de travail d'utilisation des outils

L'utilisation de l'outil avec Amazon Nova implique trois phases clés :

Requête utilisateur et définition de l'outil

Définissez les outils en fournissant des schémas JSON qui décrivent les fonctionnalités de chaque outil et les exigences d'entrée. La configuration de l'outil doit inclure des informations explicites sur le moment et le mode d'utilisation de chaque outil.

Sélection de l'outil

Lorsqu'un utilisateur envoie un message, Amazon Nova l'analyse pour déterminer si un outil est nécessaire. Ce choix d'outil automatique examine le contexte et décide quel outil (le cas échéant) invoquer. Si Amazon Nova identifie un outil approprié, il renvoie le nom de l'outil et les paramètres requis.

Vous êtes responsable de l'exécution de l'outil en fonction de la demande du modèle. Cela implique d'écrire du code qui invoque les fonctionnalités de l'outil et traite les paramètres d'entrée fournis par le modèle.

Retourner les résultats

Après avoir exécuté l'outil, renvoyez les résultats à Amazon Nova dans un format structuré à l'aide de JSON ou d'une combinaison de texte et d'images. Amazon Nova intègre les résultats de l'outil dans la réponse finale. Si des erreurs se produisent pendant l'exécution, indiquez-le dans la réponse de l'outil pour permettre à Amazon Nova de s'ajuster en conséquence.

Création d'un outil

Définissez les outils à l'aide d'une configuration d'outils qui inclut un ensemble d'outils et éventuellement un paramètre de choix d'outil. Chaque spécification d'outil doit inclure :

- Nom : identifiant clair de l'outil
- Description : explication concise des fonctionnalités de l'outil
- Schéma d'entrée : schéma JSON définissant les paramètres obligatoires et facultatifs

Exemple Exemple de configuration d'outil

```
tool_config = {
  "tools": [
    {
      "toolSpec": {
        "name": "calculator",
        "description": "A calculator tool that can execute a math equation",
        "inputSchema": {
          "json": {
            "type": "object",
            "properties": {
              "equation": {
                "type": "string",
                "description": "The full equation to evaluate"
              }
            }
          },
          "required": ["equation"]
        }
      }
    }
  ]
}
```

Bonnes pratiques pour les définitions d'outils

- Assurez-vous que le nom et la description décrivent clairement les fonctionnalités exactes de l'outil ; évitez les outils trop similaires sur le plan sémantique
- Incluez les principaux facteurs de différenciation dans la description pour aider le modèle à faire la distinction entre des outils similaires
- Limitez les schémas JSON à deux couches d'imbrication pour de meilleures performances
- Limitez les entrées à l'aide de types de schéma (par exemple, enum, int, float) plutôt que de décrire la structure en texte brut
- Indiquez les paramètres obligatoires par rapport aux paramètres facultatifs à l'aide de la notation de schéma JSON (par exemple, « required » : ["param1 », « param2 »])
- Validez votre schéma JSON à l'aide d'un validateur standard avant de le soumettre
- Placez les arguments de chaîne longue en dernier dans le schéma et évitez de les imbriquer

Sortie structurée avec décodage contraint

Les modèles Amazon Nova exploitent le décodage restreint pour garantir une fiabilité élevée des sorties générées. Cette technique utilise la grammaire pour limiter les jetons possibles à chaque étape de génération, en empêchant les clés non valides et en appliquant les types de données corrects en fonction du schéma que vous avez défini.

Exemple Exemple de sortie structurée

```
tool_config = {
  "tools": [
    {
      "toolSpec": {
        "name": "ProductAnalysis",
        "description": "Analyze product information from text.",
        "inputSchema": {
          "json": {
            "type": "object",
            "properties": {
              "name": {
                "type": "string",
                "description": "Product name"
              },
              "rating": {
```

```

        "maximum": 5,
        "description": "Customer rating 1-5",
        "type": ["number", "null"],
        "minimum": 1
    },
    "features": {
        "description": "Key product features",
        "type": "array",
        "items": {"type": "string"}
    },
    "category": {
        "type": "string",
        "description": "Product category"
    },
    "price": {
        "type": "number",
        "description": "Price in USD"
    }
},
"required": ["name", "category", "price", "features"]
}
}
}
},
"toolChoice": {
    "tool": {"name": "ProductAnalysis"}
}
}

```

Options de choix d'outils

Amazon Nova prend en charge trois paramètres de choix d'outils :

Outil

L'outil spécifié ne sera appelé qu'une seule fois, ce qui est idéal pour les cas d'utilisation de sorties structurées

N'importe lequel

L'un des outils fournis sera appelé au moins une fois, ce qui est utile pour les scénarios de sélection d'API

Auto

Le modèle décide d'appeler ou non un outil et du nombre d'outils à appeler (comportement par défaut)

Appelez un outil

Lorsqu'Amazon Nova décide d'appeler un outil, il renvoie un bloc d'utilisation de l'outil dans le cadre du message d'assistant avec `StopReason` défini sur « `tool_use` ». Le bloc d'outils contient le nom de l'outil et ses entrées.

Note

Exécutez les sections de code suivantes de manière séquentielle (Appel d'un outil → Appel d'outil de traitement → Retour des résultats de l'outil) au cours d'une seule session Python. Pour réexécuter les exemples, redémarrez votre session Python.

Exemple Appeler un outil

```
import boto3
import json

# Create Bedrock client
bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

# Complex calculation that benefits from precise computation
messages = [{
    "role": "user",
    "content": [{
        "text": "Calculate the compound interest on $10,000 invested at 4.75% annual
rate for 7 years, compounded quarterly. Use the formula  $A = P(1 + r/n)^{(nt)}$  where
 $P=10000$ ,  $r=0.0475$ ,  $n=4$ ,  $t=7$ "
    }]
}]

# Define tool configuration with calculator
tool_config = {
    "tools": [{
        "toolSpec": {
            "name": "calculator",
```

```

        "description": "Perform mathematical calculations",
        "inputSchema": {
            "json": {
                "type": "object",
                "properties": {
                    "expression": {
                        "type": "string",
                        "description": "Mathematical expression to evaluate"
                    }
                },
                "required": ["expression"]
            }
        }
    }
}

# Invoke Model
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=messages,
    toolConfig=tool_config
)

# Extract tool use from response
tool = next(
    block["toolUse"]
    for block in response["output"]["message"]["content"]
    if "toolUse" in block
)

print(f"Tool: {tool['name']}")
print(f"Expression: {tool['input']['expression']}")

```

Traitement des appels aux outils

Extrayez le nom et les arguments de l'outil du message, puis invoquez l'outil :

```

def calculate(expression):
    """Evaluate mathematical expression"""
    print(f"Calculating: {expression}")
    P = 10000
    r = 0.0475

```



```

    n = 4
    t = 7
    result = P * (1 + r/n) ** (n*t)
    return result

stop_reason = response["stopReason"]

if stop_reason == "tool_use":
    if tool["name"] == "calculator":
        result = calculate(tool["input"]["expression"])

```

Renvoi des résultats de l'outil

Renvoie les résultats de l'outil à l'aide du ToolResultBlock schéma suivant :

```

messages.append(response["output"]["message"])

# Add the tool result
messages.append({
    "role": "user",
    "content": [{
        "toolResult": {
            "toolUseId": tool['toolUseId'],
            "content": [{"json": {"result": result}}],
            "status": "success"
        }
    }]
})

# Send the tool result to the model
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=messages,
    toolConfig=tool_config
)

# Extract and display final response
final_text = next(
    block["text"]
    for block in response["output"]["message"]["content"]
    if "text" in block
)

```

```
print(f"\nFinal Response:\n{final_text}")
```

Gestion des erreurs

Signalez les erreurs à Amazon Nova pour autoriser la modification de la demande et réessayez :

```
tool_result_message = {
    "role": "user",
    "content": [
        {
            "toolResult": {
                "toolUseId": tool["toolUseId"],
                "content": [{"text": "A validation exception occurred on field:
sample.field"}],
                "status": "error"
            }
        }
    ]
}
```

Considérations sur la sécurité

- Vérifiez que les outils existent avant de les invoquer
- Assurez-vous que les entrées sont correctement formatées
- Vérifiez que les autorisations appropriées sont en place avant l'exécution de l'outil
- Fiez-vous aux détails de session plutôt que de permettre à Amazon Nova d'injecter des informations utilisateur dans les appels d'outils
- N'oubliez pas que cela LLMs peut faire halluciner les appels aux outils ; validez toujours avant l'exécution

Outils du système intégrés

Les modèles Amazon Nova 2.0 incluent des outils intégrés entièrement gérés qui ne nécessitent aucune implémentation personnalisée. Activez ces outils à l'aide d'une simple bascule dans l'API Converse.

Interprète de code

L'interpréteur de code permet à Nova d'exécuter du code Python en toute sécurité dans des environnements sandbox isolés. Cet outil est conçu pour les calculs mathématiques, les opérations logiques et les algorithmes itératifs.

Note

L'interpréteur de code est disponible dans les régions IAD, PDX et AWS NRT. Pour vous assurer que vos demandes sont acheminées vers une région prise en charge, utilisez Global CRIS. Lorsque vous utilisez les clés d'API Bedrock, vous devez ajouter manuellement `InvokeTool` des autorisations aux définitions des politiques. Le rôle Bedrock par défaut n'autorise pas l'`InvokeTool` action.

Activez l'interpréteur de code en spécifiant le paramètre `SystemTool` :

```
import boto3
import json

bedrock = boto3.client('bedrock-runtime', region_name='us-east-1')

tool_config = {
    "tools": [{
        "systemTool": {
            "name": "nova_code_interpreter"
        }
    }]
}

response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=[{
        "role": "user",
        "content": [{
            "text": "What is the average of 10, 24, 2, 3, 43, 52, 13, 68, 6, 7, 902, 82"
        }]
    }],
    toolConfig=tool_config,
    inferenceConfig={"maxTokens": 10000, "temperature": 0}
)
```

```
# Pretty print the response
for block in response["output"]["message"]["content"]:
    if "toolUse" in block:
        print("=== Tool Use ===")
        print(f"Tool: {block['toolUse']['name']}")
        print(f"Code:\n{block['toolUse']['input']['snippet']}\n")

    elif "toolResult" in block:
        print("=== Tool Result ===")
        result = block['toolResult']['content'][0]['json']
        print(f"Output: {result['stdOut']}")
        if result['stdErr']:
            print(f"Error: {result['stdErr']}")
        print(f"Exit Code: {result['exitCode']}\n")

    elif "text" in block:
        print("=== Final Answer ===")
        print(block["text"])
```

L'interpréteur exécute le code dans un sandbox et renvoie les résultats dans un schéma standard :

```
{
  "stdOut": "String",
  "stdErr": "String",
  "exitCode": "int",
  "isError": "boolean"
}
```

Ancrage Web

Le Web Grounding permet à Amazon Nova d'accéder à des informations en temps réel sur Internet, de fournir des up-to-date réponses et de réduire les hallucinations. Activez en spécifiant l'outil système `nova_grounding` :

```
tool_config = {
  "tools": [{
    "systemTool": {"name": "nova_grounding"}
  }]
}
```

Pour des informations détaillées sur le Web Grounding, consultez [the section called “Ancrage Web”](#).

Protocole de contexte du modèle (MCP)

Le Model Context Protocol (MCP) est un standard ouvert qui permet des connexions bidirectionnelles sécurisées entre les sources de données et les outils basés sur l'IA. Au lieu d'écrire des adaptateurs personnalisés pour chaque API ou service, exécutez un serveur MCP et laissez Amazon Nova découvrir ses outils automatiquement via un pont client.

Une fois connecté, Amazon Nova traite les outils MCP comme toute autre intégration externe : il décide quand les appeler, envoie les paramètres requis et intègre les résultats dans les réponses. L'utilisation d'Amazon Nova avec Strands facilite cette tâche grâce à une fonctionnalité intégrée MCPClient qui gère automatiquement la découverte, la connexion et le mappage des résultats.

Création des agents d'IA

Les modèles Amazon Nova sont optimisés pour créer des agents d'intelligence artificielle avec Amazon Nova Act. Les modèles offrent une meilleure utilisation des outils, un meilleur raisonnement pour les tâches en plusieurs étapes, une capacité améliorée à maintenir le contexte dans les flux de travail complexes des agents et une prise en charge des outils MCP à distance.

Création d'un agent

Les agents d'intelligence artificielle conçus avec Nova peuvent orchestrer plusieurs appels d'outils, maintenir le contexte lors d'interactions prolongées et corriger le cap en cas de besoin. La pensée étendue transforme les flux de travail agentiques en permettant un raisonnement systématique basé sur des objectifs complexes. Envisagez d'utiliser un SDK de structure de planification tel que Strands Agents pour renforcer le processus de planification et d'exécution de vos systèmes d'agents.

Modèles de conception des agents

Lorsque vous concevez des agents avec Nova :

- Permettez un raisonnement moyen ou élevé pour obtenir les meilleurs résultats pour les flux de travail complexes en plusieurs étapes nécessitant une planification et une vérification
- Mettre en œuvre le choix auto des outils pour permettre une sélection flexible des outils lors des interactions avec les agents
- Gestion des erreurs de conception qui permet aux agents de récupérer et de réessayer avec des approches modifiées

- Conservez l'historique des conversations pour préserver le contexte des interactions entre les agents
- Mettez en œuvre des mécanismes robustes de filtrage et de modération du contenu pour le contenu non contrôlé consommé par votre système d'agents. Par exemple, Amazon propose Amazon Bedrock Guardrails, une fonctionnalité conçue pour appliquer des garanties à plusieurs modèles de base, bases de connaissances et agents. Ces barrières de sécurité peuvent filtrer les contenus préjudiciables, bloquer les sujets refusés et supprimer des informations sensibles telles que des informations personnelles identifiables.

Exemple d'agent multi-outils

```
tool_config = {
  "tools": [
    #{
      "toolSpec": {
        "name": "calculator",
        "description": "Perform mathematical calculations",
        "inputSchema": {
          "json": {
            "type": "object",
            "properties": {
              "expression": {
                "type": "string",
                "description": "Mathematical expression to evaluate"
              }
            }
          },
          "required": ["expression"]
        }
      }
    },
    #{
      "toolSpec": {
        "name": "database_query",
        "description": "Query financial database for historical data",
        "inputSchema": {
          "json": {
            "type": "object",
            "properties": {
              "query": {
                "type": "string",
```

```

        # "description": "SQL query to execute"
        #}
        #},
        # "required": ["query"]
        #}
        #}
        #}
        #}
        #}
    #]
}

response = client.converse(
    #modelId=" us.amazon.nova-2-lite-v1:0",
    #messages=[{
        # "role": "user",
        # "content": [{
            # "text": "Analyze our Q3 financial performance across all business units,
calculate year-over-year growth rates with statistical significance testing, and
recommend budget allocation strategies for Q4."
        #}]
    #}],
    #toolConfig=tool_config,
    #inferenceConfig={"maxTokens": 10000, "temperature": 1, "topP": 0.9},
    #additionalModelRequestFields={
        # "reasoningConfig": {
            # "type": "enabled",
            # "maxReasoningEffort": "low"
        #}
    #}
    #}
)

```

Invocation d'un agent

L'invocation d'un agent implique la gestion du flux de conversation, le traitement des appels aux outils et le maintien de l'état lors de multiples interactions.

Réponses des agents de streaming

Diffusez les réponses pour fournir une visibilité en temps réel sur le raisonnement et les actions de l'agent :

```

import boto3
response = client.converse_stream(

```

```

#modelId=" us.amazon.nova-2-lite-v1:0",
#messages=[{
  #role: "user",
  #content": [{
    #text": "Design a scalable microservices architecture for an e-commerce
platform handling 1M+ daily transactions. Consider data consistency, fault tolerance,
performance, security, and cost optimization."
  #}]
#}],
#inferenceConfig={"maxTokens": 10000, "temperature": 10},
#additionalModelRequestFields={
  #"reasoningConfig": {
    #"type": "enabled",
    #"maxReasoningEffort": "low"
  #}
#}
)

# Process the streaming response
reasoning_complete = False
for event in response["stream"]:
  #if "contentBlockDelta" in event:
    #delta = event["contentBlockDelta"]["delta"]

    #if "reasoningContent" in delta:
      #reasoning_text = delta["reasoningContent"]["reasoningText"]["text"]
      #print(f"{reasoning_text}", end="", flush=True)
    #elif "text" in delta:
      #if not reasoning_complete:
        #print(f"
Final Architecture Design:
")
    #reasoning_complete = True
    #print(f"{delta['text']}", end="", flush=True)

```

Gestion de l'état des agents

Conservez l'historique des conversations et les résultats des outils pour préserver le contexte ; l'exemple ci-dessous le montre en un tour de main, mais le développeur peut déterminer comment orchestrer l'ensemble du système d'agents en fonction des exigences du flux de travail. En outre, les outils Amazon Web Services tels que Strands gèrent le contexte des agents et l'état des outils pour le compte du développeur.


```
messages = []

messages = []

# Initial user query
messages.append({
    #"role": "user",
    #"content": [{"text": user_query}]
})

# Get agent response
response = client.converse(
    #modelId=" us.amazon.nova-2-lite-v1:0",
    #messages=messages,
    #toolConfig=tool_config,
    #inferenceConfig=inf_params
)

# Add assistant response to history
messages.append(response["output"]["message"])

# Process tool calls and add results
if response["stopReason"] == "tool_use":
    #tool = next(
        #block["toolUse"]
        #for block in response["output"]["message"]["content"]
        #if "toolUse" in block
    #)

    ## Execute tool
    #result = execute_tool(tool["name"], tool["input"])

    ## Add tool result to conversation
    #messages.append({
        #"role": "user",
        #"content": [{
            #"toolResult": {
                #"toolUseId": tool["toolUseId"],
                #"content": [{"json": result}],
                #"status": "success"
            #}
        #}]
    #})
```

```
## Continue conversation
#response = client.converse(
    #modelId=" us.amazon.nova-2-lite-v1:0",
    #messages=messages,
    #toolConfig=tool_config,
    #inferenceConfig=inf_params
#)
```

Meilleures pratiques pour les agents

Pour plus d'informations sur les meilleures pratiques relatives aux agents, consultez [the section called “Bonnes pratiques”](#).

Pour obtenir des conseils sur le développement d'agents d'IA conversationnelle, consultez [Discours à parole](#).

Ancrage Web

Web Grounding permet à Amazon Nova de rechercher sur le Web des informations actuelles et de fournir des réponses avec des citations. Cette fonctionnalité est utile pour les requêtes nécessitant des up-to-date informations autres que les données d'entraînement du modèle.

Fonctionnement de l'ancrage Web

Lorsque l'ancrage Web est activé pour une invite, les étapes suivantes sont effectuées :

1. Configuration de la demande : votre application envoie une invite utilisateur à l'API Amazon Bedrock avec nova_grounding activé en tant que. `systemTool`
2. Recherche et analyse : le modèle détermine si une recherche est nécessaire, effectue une ou plusieurs recherches pour obtenir des informations pertinentes et évalue si des recherches supplémentaires sont nécessaires pour approfondir sa compréhension ou approfondir des sous-sujets spécifiques.
3. Génération de réponses : Amazon Nova synthétise automatiquement les informations issues des résultats de recherche pour générer une réponse API finale basée sur des informations en temps réel, avec des citations de ses sources.

Comment utiliser l'ancrage Web

Pour des exemples complets de code utilisant Web Grounding, consultez la section Exemples de code.

Pour inclure Web Grounding dans vos résultats, spécifiez le `systemTool` paramètre suivant dans votre bloc `ToolConfig` :

```
import boto3
from botocore.config import Config

# Create the Bedrock Runtime client with extended timeout
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600)
)

# Define the tool configuration
tool_config = {
    "tools": [{
        "systemTool": {
            "name": "nova_grounding"
        }
    }]
}

# Send the request
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=[{
        "role": "user",
        "content": [{"text": "What are the latest developments in quantum computing?"}]
    }],
    toolConfig=tool_config
)

# Extract text with interleaved citations
output_with_citations = ""
content_list = response["output"]["message"]["content"]
for content in content_list:
    if "text" in content:
        output_with_citations += content["text"]
```

```
elif "citationsContent" in content:
    citations = content["citationsContent"]["citations"]
    for citation in citations:
        url = citation["location"]["web"]["url"]
        output_with_citations += f" [{url}]"

print(output_with_citations)
```

Disponibilité par région

Web Grounding n'est actuellement disponible que dans les régions des États-Unis et n'est pris en charge que par les profils CRIS américains.

Structure de réponse

Voici un exemple de réponse. La réponse a été raccourcie par souci de concision :

```
{
  "output": {
    "message": {
      "content": [
        {
          "text": "Recent quantum computing developments include...",
          "citationsContent": [
            {
              "location": {
                "web": {
                  "url": "https://example.com/quantum-news",
                  "domain": "example.com"
                }
              }
            }
          ]
        }
      ]
    }
  }
}
```

Chaque citation inclut :

- **text**: segment de la réponse générée par le modèle.

- `citationsContent`: conteneur principal pour les données de citation associées à un segment de texte.
- `citations`: un contenant `citationsContent` contenant l'emplacement d'une citation.
- `location`: un contenant `citations` contenant la source d'une citation.
- `web`: un conteneur `location` contenant les détails de la source Web.
- `url`: adresse Web complète (URL) de la source de la citation.
- `domain`: domaine racine de l'URL source.

Sécurité de mise à la terre

Vos données ne quittent jamais AWS l'infrastructure. Les requêtes générées par le modèle restent dans le cadre AWS des services et ne sont jamais envoyées sur Internet au sens large. Notre vaste index de recherche interne sur le Web et nos graphiques de connaissances donnent la priorité aux sources fiables et de haute qualité et filtrent les contenus malveillants dès leur entrée. Enfin, nous protégeons votre application contre l'injection d'invites indirectes et les informations erronées grâce à un filtrage à l'exécution (veuillez noter que cette mesure d'atténuation est limitée pour les langues autres que l'anglais).

Gestion des erreurs

Veuillez ne pas inclure d'entrée `toolSpec` avec le nom `nova_grounding`. L'inclusion d'un outil portant ce nom entraînera une erreur.

Voici une liste des erreurs potentielles qui peuvent survenir lors de l'utilisation de l'ancrage Web :

- `malformed_tool_use`
- `max_tokens`
- `malformed_model_output`

Autorisations requises pour les outils intégrés

Pour que votre rôle puisse accéder à Web Grounding sur Amazon Bedrock, deux options s'offrent à vous :

1. Activez `BedrockFullAccess` sur votre rôle IAM : si votre rôle l'a fait `BedrockFullAccess`, il aura automatiquement accès à Web Grounding.

2. Ajoutez des autorisations spécifiques (si nécessaire) : si vous avez besoin d'un contrôle d'accès plus précis, ajoutez cette politique à la politique IAM de votre rôle, en remplaçant l'ID du compte par le vôtre : Compte AWS

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["bedrock:InvokeTool"],
      "Resource": ["arn:aws:bedrock::{111122223333}:system-tool/
amazon.nova_grounding"]
    }
  ]
}
```

La clé de `aws:requestedRegion` condition de Web Grounding est définie sur « non spécifiée ». Si vos politiques existantes ou vos politiques de contrôle des services (SCPs) appliquent cette condition, vous pouvez rencontrer des problèmes d'accès. La mise à jour de la condition pour autoriser une `requestedRegion` « unspecified » peut résoudre ce problème.

Note

Si vous activez l'outil d'ancrage Web, vous êtes responsable de votre utilisation et de toute utilisation par vos utilisateurs finaux des résultats qui intègrent des informations ancrées. Vous saurez quand vos résultats incluent des informations ancrées grâce aux citations ou aux liens vers les sources. Vous devez conserver et afficher ces citations et ces liens dans les résultats que vous fournissez à vos utilisateurs finaux.

Note

L'ancrage Web entraîne des frais supplémentaires. Pour plus d'informations, consultez la page de [tarification de AWS Bedrock](#).

Guide d'ingénierie rapide

L'ingénierie rapide consiste à concevoir des instructions pour communiquer efficacement avec de grands modèles linguistiques (LLMs) tels qu'Amazon Nova. Une ingénierie rapide vous permet de contrôler le comportement des modèles, d'améliorer la qualité de sortie et de créer des applications fiables basées sur l'IA.

Une ingénierie rapide et efficace implique de créer des entrées qui guident les modèles pour produire les résultats souhaités. Une invite bien conçue fournit des instructions claires, un contexte pertinent et des données d'entrée correctement formatées.

Qu'est-ce que l'ingénierie rapide

L'ingénierie de requête désigne la pratique consistant à optimiser les entrées textuelles d'un grand modèle de langage (LLM) afin d'améliorer les résultats et d'obtenir les réponses que l'on veut. Les invites aident un LLM à effectuer une grande variété de tâches, notamment la classification, la réponse aux questions, la génération de code, l'écriture créative, etc. La qualité des invites que vous fournissez à un LLM peut avoir un impact sur la qualité des réponses du modèle. Cette section fournit les informations nécessaires pour faire vos premiers pas avec l'ingénierie de requête. Il couvre également des outils pour vous aider à trouver le meilleur format d'invite possible pour votre cas d'utilisation lorsque vous utilisez un LLM sur Amazon Bedrock.

L'efficacité des invites dépend de la qualité des informations fournies et de la qualité de l'invite elle-même. Les invites peuvent inclure des instructions, des questions, des détails contextuels, des entrées et des exemples afin de guider efficacement le modèle et d'améliorer la qualité des résultats. Ce document décrit les stratégies et les tactiques permettant d'optimiser les performances de la gamme de modèles Amazon Nova 2 Sonic. Les méthodes présentées ici peuvent être utilisées dans différentes combinaisons afin d'amplifier leur efficacité. Nous vous encourageons à expérimenter afin d'identifier les approches les mieux adaptées à leurs besoins spécifiques.

Commencer par une ingénierie rapide

Avant de commencer l'ingénierie de requête, nous vous recommandons de mettre en place les éléments suivants afin de pouvoir développer de manière itérative la requête la plus optimale pour votre cas d'utilisation :

Définissez votre cas d'utilisation

Définissez votre cas d'utilisation selon quatre dimensions :

- **Tâche** : définissez ce que vous souhaitez que le modèle accomplisse. Cela permet de déterminer la bonne technique d'incitation.
- **Rôle** : définissez le rôle que le modèle doit assumer pour accomplir la tâche. Les modèles Amazon Nova prennent en charge trois rôles (système, utilisateur ou assistant).
- **Style de réponse** : définissez la structure ou le style de réponse que le modèle doit suivre en fonction de l'audience, tel que JSON, Markdown ou conversationnel.
- **Instructions** — Définissez l'ensemble d'instructions que le modèle doit suivre pour répondre aux critères de réussite.

Établissez des critères de réussite

Définissez des critères de réussite ou des mesures d'évaluation. Vous pouvez fournir une liste de critères ou des indicateurs d'évaluation spécifiques, tels que la longueur, le score BLEU, le ROUGE, le format, la factualité et la fidélité.

Rédiger une invite

Créez une invite de départ incorporant votre tâche, votre rôle, votre style de réponse et vos instructions. Effectuez une itération en fonction des résultats.

L'efficacité des instructions dépend de la qualité des informations que vous fournissez.

Bonnes pratiques d'ordre général

Les meilleures pratiques suivantes s'appliquent principalement aux modèles de texte Amazon Nova, mais vous pouvez les appliquer à d'autres modèles, en plus des meilleures pratiques spécifiques aux modalités.

Pour plus d'informations sur la manière de demander des entrées multimodales, reportez-vous à [Encourager les entrées multimodales](#). Pour plus d'informations sur la procédure à suivre pour demander la saisie vocale, reportez-vous à [Invites de conversation vocale](#).

Comprendre les rôles

Les modèles Amazon Nova vous permettent de structurer les invites en utilisant trois rôles distincts : système, utilisateur et assistant.

- **Système (facultatif)** — Bien que cela ne soit pas obligatoire, il établit les paramètres comportementaux globaux de l'assistant. Il peut également être utilisé pour fournir des instructions et des directives supplémentaires que l'utilisateur souhaite que le modèle respecte tout au long de la conversation.
- **Utilisateur** : peut éventuellement transmettre le contexte, les tâches, les instructions et le résultat souhaité avec la requête de l'utilisateur.
- **Assistant** — Aide à orienter le modèle vers la réponse prévue.

Rubriques

- [Créez des instructions précises](#)
- [Mettre l'accent sur certaines sections de l'invite](#)
- [Utilisation du rôle système](#)
- [Fournir des exemples \(invites à quelques essais\)](#)
- [Systèmes d'appel d'outils](#)
- [Techniques d'incitation avancées](#)

Créez des instructions précises

La rédaction de requêtes utilisateur spécifiques est cruciale dans l'ingénierie de requête. Des requêtes bien formulées permettent aux modèles de compréhension de texte Amazon Nova de générer des réponses précises et pertinentes. Pour créer de telles requêtes, il est essentiel de commencer par fournir des informations contextuelles dans la requête. Le contexte fourni aide à mieux situer le modèle en l'alignant sur des scénarios d'utilisateurs cibles clairs, améliorant ainsi la pertinence et la cohérence du résultat.

Fournir des informations contextuelles et spécifiques

Les modèles fonctionnent mieux si les instructions sont claires, bien structurées, courtes et spécifiques. Ce modèle de prompt s'applique à la fois à l'invite utilisateur et à l'invite système.

Modèle d'invite

```
# Use this to clearly define the task and job needed by the model
task_summary = f"""
## Task Summary:
```

```

{{Task summary}}
"""

# Use this to provide contextual information related to the task
context_information = f"""
## Context Information:
- {{Context and content information 1}}
- {{Context and content information 2}}
...
"""

# Use this to provide any model instructions that you want model to adhere to
model_instructions = f"""
## Model Instructions:
- {{ Other Model Instructions }}
...
"""

# Use this to provide response style and formatting guidance
response_style = f"""
## Response style and format requirements:
- {{Style and format requirement 1}}
- {{Style and format requirement 2}}
...
"""

# Concatenate to final prompt
final_prompt = f"""{{task_summary}}
{{context_information}}
{{model_instructions}}
{{response_style}}"""

```

Exemple : informations spécifiques et contextuelles

Voici un exemple d'invite réécrit avec des informations spécifiques et contextuelles. L'ajout d'informations contextuelles supplémentaires aide le modèle à s'ancrer vers le comportement souhaité.

Unclear prompt

User:

Write me a meeting invite to the project team

Assistant:

Certainly! Here's a sample meeting invite for your project team:

****Subject:**** Project Team Meeting - [Project Name]

****Date:**** [Insert Date]

****Time:**** [Insert Time]

****Duration:**** [Insert Duration]

****Location:**** [Insert Location] / [Zoom/Teams Link if Online]

****Agenda:****

1. ****Welcome and Introductions****

- Quick round of introductions for new team members

2. ****Project Overview****

- Objectives and goals
- Current status and milestones achieved

3. ****Task Allocation and Deadlines****

- Detailed breakdown of tasks
- Assignment of responsibilities
- Discussion of deadlines

4. ****Risk Management****

- Identification of potential risks
- Mitigation strategies

5. ****Resource Allocation****

- Required tools and materials
- Budget review

6. ****Q&A Session****

- Open floor for questions and clarifications

7. ****Next Steps and Closing****

- Summary of action items
- Schedule for the next meeting

****Additional Notes:****

- Please come prepared with any relevant documents or updates regarding your assigned tasks.
- If you anticipate any issues or require additional support, please let us know in advance so we can address them during the meeting.

****Contact Information:****

For any questions or concerns before the meeting, please feel free to reach out to [Your Name] at [Your Email] or [Your Phone Number].

We look forward to a productive discussion and to kicking off our project with enthusiasm!

Best regards,

[Your Full Name]

[Your Position]

[Your Contact Information]

Feel free to customize this template to better fit your specific project and team needs!

Clear prompt

User:

Write me a meeting invite to the project team. **The meeting is about project planning, and should be held on October 16th, from 10 AM to 11 AM in Conference Room B. Please include an agenda that covers our progress on the project thus far, as well as any upcoming milestones and deadlines.**

Assistant:

****Meeting Invitation: Project Planning Session****

****Date:**** October 16th

****Time:**** 10:00 AM - 11:00 AM

****Location:**** Conference Room B

****Agenda:****

1. ****Welcome and Introduction (10:00 AM - 10:05 AM)****
 - Brief overview by the meeting facilitator.
2. ****Project Progress Update (10:05 AM - 10:30 AM)****
 - Review of completed tasks and achievements to date.
 - Detailed discussion on current project status and any challenges encountered.
3. ****Upcoming Milestones and Deadlines (10:30 AM - 10:50 AM)****
 - Overview of the next phase of the project.
 - Identification of key milestones and deadlines.
4. ****Action Items and Responsibilities (10:50 AM - 11:00 AM)****
 - Assignment of tasks for the upcoming period.
 - Confirmation of individual responsibilities and follow-up items.
5. ****Q&A Session (11:00 AM)****
 - Open floor for any questions or concerns from team members.

Please ensure your attendance to discuss and finalize our project planning for the next phase. Your insights and contributions are highly valued.

Best regards,

[Your Name]

[Your Position]

[Your Contact Information]

- Soyez clair : lorsque vous interrogez le modèle, essayez de donner des instructions aussi claires que possible, de manière à ce qu'un humain puisse les comprendre plutôt qu'une machine (par exemple, une liste de type JSON).
- Soyez précis : donnez au modèle des instructions spécifiques sur la tâche à accomplir, le résultat attendu et tout contexte supplémentaire afin de l'aider à répondre comme vous le voulez.

Spécifiez le format et le style de sortie

Si vous souhaitez qu'Amazon Nova 2 génère des réponses adaptées à vos besoins spécifiques, vous pouvez spécifier le format et le style de la sortie. Par exemple, si vous voulez générer des réponses uniquement au format JSON ou limiter la longueur de la réponse à un nombre maximal de mots, de puces ou de paragraphes. Cela permet à Amazon Nova 2 de mieux comprendre vos besoins et de produire des résultats plus pertinents et plus faciles à analyser.

Utilisez des majuscules et des formulations fortes telles que « FAIRE », « NE PAS FAIRE », « DOIT » pour souligner les règles que vous voulez que le modèle respecte strictement. Par exemple, si vous voulez obtenir un seul objet JSON dans la réponse, vous pouvez indiquer au modèle : « Vous DEVEZ répondre uniquement au format JSON. N'utilisez AUCUN autre format pour répondre à la question. »

Voici quelques exemples d'invites réécrites avec des informations spécifiques et contextuelles. L'ajout d'informations contextuelles supplémentaires aide le modèle à s'ancrer vers le comportement souhaité.

Exemple 1 : fournir des détails supplémentaires

L'ajout de détails supplémentaires sur le formatage de la réponse peut aider le modèle à obtenir le résultat souhaité. Parmi les autres formats de réponse couramment utilisés, on trouve CSV, Markdown, YAML et XML

Prompt without style constraints

User:

```
Outline the different types of rooms available in a typical 4-star hotel, including their features.
```

Assistant:

```
In a typical 4-star hotel, you'll find a range of room types, each designed to cater to different guest needs and preferences. Here's an outline of the different types of rooms available along with their features:
```

```
### 1. **Standard Room**
```

- **Size:** Usually around 250-350 square feet.
- **Features:**
 - Queen or double bed.
 - Complimentary Wi-Fi.
 - Flat-screen TV with cable or satellite channels.
 - Mini-fridge.
 - Coffee maker with complimentary coffee and tea.
 - Desk and chair.
 - En-suite bathroom with shower, toilet, and toiletries.
 - Air conditioning/heating.

```
### 2. **Deluxe Room**
```

- **Size:** Typically larger than standard rooms, around 350-450 square feet.
- **Features:**
 - King bed or two queen beds.
 - Larger bathroom with a bathtub.
 - Additional seating area.
 - Enhanced amenities like bathrobes and slippers.
 - Upgraded toiletries and sometimes a welcome gift.
 - Access to a lounge or executive floor perks.

3. Suite

- **Size:** Varies significantly, often between 500-1,200 square feet.
- **Features:**
 - Separate living area with a sofa bed or additional seating.
 - Kitchenette or full kitchen in some cases.
 - Larger, more luxurious bathroom with a bathtub and separate shower.
 - Multiple TVs.
 - Access to executive lounge.
 - High-end toiletries and often complimentary breakfast.
 - Often includes concierge services and in-room dining options.

4. Family Room

- **Size:** Larger, usually around 400-500 square feet.
- **Features:**
 - One king bed and one sofa bed or two queen beds.
 - Designed for families with children.
 - Extra space for luggage.
 - Child-friendly amenities like a small kitchenette or high chair.
 - Family bathroom with amenities suitable for kids.

5. Executive Room

- **Size:** Typically around 350-450 square feet.
- **Features:**
 - Business-oriented with a larger workspace.
 - Executive desk and chair.
 - Access to executive lounge with complimentary snacks and refreshments.
 - King bed or two queen beds.
 - Enhanced amenities like high-speed internet and sometimes a Nespresso machine.

6. Junior Suite

- **Size:** Usually around 450-600 square feet.
- **Features:**
 - Separate bedroom and living area.
 - King bed or two queen beds.
 - Kitchenette or mini-fridge.

- Luxurious bathroom with a bathtub.
- Lounge access.
- High-end toiletries.

7. **Premium Room**

- **Size:** Larger than standard rooms, around 400-500 square feet.
- **Features:**
 - King bed or two queen beds.
 - Larger bathroom with a bathtub.
 - Enhanced amenities like a welcome drink and premium toiletries.
 - Access to premium services and sometimes a lounge.

8. **Connecting Rooms**

- **Size:** Two rooms that connect via an internal door.
- **Features:**
 - Suitable for families or groups.
 - Often includes one standard room and one suite.
 - Shared bathroom or en-suite bathrooms in each room.
 - Extra space for luggage and guests.

9. **Accessible Rooms**

- **Size:** Varies, typically similar to standard rooms.
- **Features:**
 - Designed for guests with disabilities.
 - Wider doorways and bathroom space.
 - Adapted fixtures and fittings.
 - Additional support services.

10. **Business Room**

- **Size:** Usually around 300-350 square feet.
- **Features:**
 - Focused on business travelers.
 - High-speed internet and large workspace.
 - Compact yet fully equipped with essentials.
 - Often includes a king bed.

Each room type in a 4-star hotel is designed to provide comfort, convenience, and luxury tailored to different guest needs, ensuring a pleasant stay for all types of travelers.

Prompt with style constraints

User:

Outline the different types of rooms available in a typical 4-star hotel, including their features. **All responses MUST be in YAML format.**

Assistant:

```
```yaml
hotel_rooms:
- room_type: Standard Room
 features:
 - description: A comfortable room with essential amenities.
 amenities:
 - bed_type: Double bed or two single beds
 - bathroom: Private bathroom with shower
 - wifi: Complimentary high-speed internet
 - TV: Flat-screen TV with cable channels
 - mini_fridge: Available
 - tea_and_coffee_making_facilities: Yes

- room_type: Deluxe Room
 features:
 - description: An upgraded room with more space and enhanced amenities.
 amenities:
 - bed_type: King-size bed or two single beds
 - bathroom: Private bathroom with both shower and bathtub
 - wifi: Complimentary high-speed internet
 - TV: Larger flat-screen TV with cable channels
 - mini_fridge: Available
 - tea_and_coffee_making_facilities: Yes
 - free_breakfast: Included

- room_type: Suite
 features:
 - description: A luxurious room with separate living area and premium amenities.
 amenities:
 - bed_type: King-size bed or two single beds
 - bathroom: Private bathroom with both shower and bathtub
 - separate_living_area: Yes
 - wifi: Complimentary high-speed internet
 - TV: Large flat-screen TV with cable channels
 - mini_fridge: Available
 - tea_and_coffee_making_facilities: Yes
 - free_breakfast: Included
 - concierge_service: Available
```

```
- room_type: Family Room
features:
- description: A spacious room designed for families with additional bedding
options.
amenities:
- bed_type: Queen bed and rollaway bed or two single beds and rollaway bed
- bathroom: Private bathroom with shower
- wifi: Complimentary high-speed internet
- TV: Flat-screen TV with cable channels
- mini_fridge: Available
- tea_and_coffee_making_facilities: Yes

- room_type: Business Room
features:
- description: A room tailored for business travelers with enhanced workspace.
amenities:
- bed_type: Double bed or two single beds
- bathroom: Private bathroom with shower
- wifi: Complimentary high-speed internet
- TV: Flat-screen TV with cable channels
- workspace: Desk and chair
- mini_fridge: Available
- tea_and_coffee_making_facilities: Yes
- business_center_access: Available
...
```

## Exemple 2 : fournir des contraintes de style

Dans ce cas, l'invite avec des contraintes de style fournit des informations complètes sur les contraintes à respecter et le format de réponse souhaité. Comme le montre la réponse, l'invite optimisée correspond davantage à la concision demandée et à la structure de liste à puces.

### Prompt without style constraints

User:

Explain the process of checking in a guest at a luxury resort.

Assistant:

Checking in a guest at a luxury resort involves a meticulous and personalized process designed to ensure a seamless and memorable start to their stay. Here's a detailed overview of the steps involved:

### ### Pre-Arrival Preparation

#### 1. **Pre-Arrival Communication:**

- **Confirmation:** Verify guest reservations and confirm arrival details.
- **Special Requests:** Collect any special requests (dietary restrictions, room preferences, etc.) and ensure they are noted in the guest profile.
- **Documentation:** Ensure all necessary documentation is ready, such as welcome letters, room keys, and any special amenities.

#### 2. **Room Preparation:**

- **Cleaning:** Conduct a thorough cleaning and preparation of the guest room.
- **Amenities:** Stock the room with high-end toiletries, fresh linens, and any requested amenities.
- **Personal Touches:** Add personalized touches such as a welcome note, fresh flowers, or locally sourced treats.

### ### Arrival and Initial Interaction

#### 3. **Guest Arrival:**

- **Greeting:** Have staff ready to greet guests as they arrive. This may include a warm welcome at the entrance or a meet-and-greet in the lobby.
- **Transport:** Provide luxury transport options like a chauffeured car service to escort guests to the check-in desk.

#### 4. **Check-In Desk:**

- **Professionalism:** Ensure check-in staff are well-trained, professional, and exude hospitality.
- **Technology:** Utilize sophisticated check-in technology (like tablet kiosks) to streamline the process.
- **Personalized Welcome:** Greet guests with a warm welcome, acknowledge their special requests, and present them with a personalized welcome kit.

### ### Documentation and Formalities

#### 5. **Document Verification:**

- **Identification:** Verify identification and payment methods.
- **Membership Cards:** If applicable, verify any loyalty or membership cards for additional benefits.

## 6. **\*\*Registration:\*\***

- **\*\*Form Filling:\*\*** Assist guests with any required paperwork.
- **\*\*Special Benefits:\*\*** Clearly explain any special resort benefits, upgrades, or exclusive access to amenities that the guest is entitled to.

### ### Room Assignment

## 7. **\*\*Room Presentation:\*\***

- **\*\*Walk-Through:\*\*** Escort guests to their room and conduct a walk-through to ensure everything is in order and to address any immediate concerns.
- **\*\*Upgrades:\*\*** Present any room upgrades and explain the features and services available in the upgraded room.

## 8. **\*\*Key Delivery:\*\***

- **\*\*Room Key:\*\*** Provide the room key, often a high-end key card or a luxury key fob.
- **\*\*Access Information:\*\*** Explain how to use the key card, including access to other amenities like the spa, pool, and gym.

### ### Additional Services

## 9. **\*\*Welcome Amenities:\*\***

- **\*\*Amenity Kits:\*\*** Deliver welcome amenity kits (like a bottle of champagne, local delicacies, etc.).
- **\*\*Itinerary:\*\*** Provide a detailed resort map and itinerary, highlighting all available services and activities.

## 10. **\*\*Personal Concierge:\*\***

- **\*\*Concierge Services:\*\*** Introduce the guest to the concierge service and explain how to request any special services, such as dining reservations, transportation, or local excursions.
- **\*\*Contact Information:\*\*** Provide contact information for key staff members.

### ### Final Touches

## 11. **\*\*Feedback:\*\***

- **\*\*Satisfaction Survey:\*\*** At the end of the check-in process, ask for feedback to ensure the guest's expectations were met and to gather any immediate suggestions.

## 12. **\*\*Follow-Up:\*\***

- **\*\*Immediate Issues:\*\*** Ensure any immediate issues or requests are addressed promptly.
- **\*\*Comfort Check:\*\*** Make follow-up calls or send messages to ensure the guest is settling in comfortably.

By following these steps, a luxury resort aims to provide an exceptional check-in experience that sets the tone for an unforgettable stay.

### Prompt with style constraints

User:

Explain the process of checking in a guest at a luxury resort. **Provide all answers in a numbered list format, with each point containing no more than 15 words, and no more than 6 bullet points.**

Assistant:

1. Greet guest warmly with a smile and offer assistance.
2. Verify reservation details and check guest's identification.
3. Confirm room assignment and amenities provided.
4. Explain resort services and amenities available.
5. Present room key and offer directions to guest's suite.
6. Wish guest a pleasant stay and be available for queries.

### Exemple 3 : limiter le format de réponse

Ici, l'invite avec le format de réponse oblige le modèle à répondre uniquement avec le numéro de catégorie et à le limiter entre 1 et 10, au lieu d'être très prolixe.

### Prompt without response format

User:

In this task, you are given a text which is the body of a document. Your job is to classify the topic of the document into these categories: 1)Company, 2)Educational Institution, 3)Artist, 4)Athlete, 5)Office Holder, 6)Mean of transportation, 7)Building, 8)Natural place, 9)Village, 10)Animal.  
Text: Johangsan is a mountain of South Korea. It has an elevation of 954 metres.

Assistant:

To classify the topic of the given text, let's analyze the content step-by-step:

1. The text mentions "Johangsan," which appears to be a specific name.
2. The text states that "Johangsan is a mountain," which provides a clear indication of its nature.
3. It is further specified that it is located in "South Korea" and has an elevation of "954 metres."

Based on the information provided, the text describes a geographical feature, specifically a mountain. From the given categories, the most appropriate classification for this document is:

8) Natural place

The text does not mention any company, educational institution, artist, athlete, office holder, means of transportation, building, village, or animal. Therefore, the classification as a "Natural place" is the most fitting choice.

## Prompt with response format

User:

In this task, you are given a text which is the body of a document. Your job is to classify the topic of the document into these categories: 1)Company, 2)Educational Institution, 3)Artist, 4)Athlete, 5)Office Holder, 6)Mean of transportation, 7)Building, 8)Natural place, 9)Village, 10)Animal. **Your output should be the category number. DO NOT generate anything apart from numbers 1-10.**

Text: Johangsan is a mountain of South Korea. It has an elevation of 954 metres.

Assistant:

8

## Fournir des sections d'invite claires

Au-delà du contexte initial, un format de requête structurée joue un rôle important dans l'optimisation des interactions avec le modèle linguistique. La séparation des différentes parties de l'entrée par des délimiteurs garantit la clarté et améliore la capacité des modèles Amazon Nova 2 à comprendre la structure de la requête. Une autre méthode efficace consiste à utiliser des listes numérotées ou des puces, qui peuvent aider à décomposer la tâche en étapes gérables et à fournir un aperçu clair des actions à entreprendre. Cette approche structurée contribue à réduire l'ambiguïté et à améliorer la précision des réponses générées.

## Exemple : créer une invite sectionnée

Remarquez comment, dans la colonne « Invite sectionnée », l'invite est plus claire et comporte des sections délimitées pour différentes entités telles que les tâches, le contexte, les exigences et le reste, par rapport aux instructions de la colonne « Invite non sectionnée ».

### Unsectioned Prompt

User:

Write an article about climate change for high school students explaining basic concepts, current impacts, and individual actions to mitigate it. Make it informative and engaging with simple language, including statistics and real-world examples.

Assistant:

### Understanding Climate Change: What It Is, What It's Doing, and What You Can Do About It

#### Introduction

Climate change might sound like a big, complicated topic, but it's really about the way our planet's weather is changing over time. This change is mostly happening because humans are doing things that are making our Earth warmer. Let's break it down so you can understand what's going on and what you can do to help.

#### What Is Climate Change?

Climate change refers to long-term shifts in temperatures and weather patterns. These shifts may be natural, but since the 1800s, human activities have been the main driver of rapid climate change.

**\*\*Key Concept: Greenhouse Gases\*\***

To understand this, we need to talk about greenhouse gases. These gases, like carbon dioxide (CO<sub>2</sub>), methane (CH<sub>4</sub>), and nitrous oxide (N<sub>2</sub>O), trap heat in our atmosphere. Without these gases, Earth would be too cold for most life forms. But when we burn fossil fuels (like coal, oil, and natural gas) and cut down forests, we release too much CO<sub>2</sub> and other greenhouse gases, trapping more heat and making the planet warmer.

#### Current Impacts of Climate Change

Let's look at some real-world examples to see how climate change is affecting our world today.

**\*\*1. Rising Temperatures\*\***

Since the late 19th century, Earth's average surface temperature has risen about 2 degrees Fahrenheit (1 degree Celsius). Scientists predict that if we don't change our ways, it could rise by 3.6 to 4.3 degrees Fahrenheit (2 to 2.4 degrees Celsius) by the end of this century.

**\*\*2. More Extreme Weather\*\***

Climate change is making extreme weather events more frequent and severe. For example:

- **\*\*Hurricanes:\*\*** In the U.S., hurricane seasons are getting longer and more destructive. In 2020, there were 30 named storms, the most ever recorded.
- **\*\*Droughts:\*\*** Parts of California have experienced some of the worst droughts in history, affecting agriculture and water supplies.

**\*\*3. Melting Ice Caps and Rising Sea Levels\*\***

Glaciers and ice sheets are melting at alarming rates. This not only affects wildlife but also leads to rising sea levels. Since 1901, global sea levels have risen by about 8 inches (20 centimeters). By 2100, they could rise by another 1 to 4 feet (0.3 to 1.2 meters).

**\*\*4. Ocean Acidification\*\***

As CO2 levels rise, more of it is absorbed by the oceans, making them more acidic. This is particularly harmful to marine life, including coral reefs. Coral reefs are crucial for marine biodiversity but are now dying at an alarming rate.

**#### Individual Actions to Mitigate Climate Change**

You might think that since climate change is a huge problem, one person can't make a difference. But even small actions can add up to make a big impact.

**\*\*1. Reduce, Reuse, Recycle\*\***

Try to minimize waste by:

- **\*\*Reducing:\*\*** Buy fewer items and choose products with less packaging.
- **\*\*Reusing:\*\*** Use reusable bags, bottles, and containers.
- **\*\*Recycling:\*\*** Make sure to recycle paper, plastic, and glass properly.

**\*\*2. Conserve Energy\*\***

Small changes can lead to big savings:

- Turn off lights when you leave a room.
- Unplug devices that you're not using.
- Use energy-efficient bulbs and appliances.

**\*\*3. Use Less Meat\*\***

Livestock farming is a major contributor to greenhouse gas emissions. By eating less meat, especially beef, you can reduce your carbon footprint.



**\*\*4. Support Renewable Energy\*\***

If possible, support or switch to renewable energy sources like solar or wind power. Even advocating for these options in your community can make a difference.

**\*\*5. Educate Yourself and Others\*\***

Knowledge is power. The more you know about climate change, the more you can help others understand it. Share facts and encourage your friends and family to take action too.

**#### Conclusion**

Climate change is a pressing issue that affects everyone on the planet. By understanding its basics, recognizing its current impacts, and taking individual actions, you can play a crucial role in mitigating its effects. Remember, every small step counts, and together, we can make a big difference for our planet.

Let's start today and build a sustainable future!

## Sectioned Prompt

User:

**\*\*Task:\*\***

Write an article

**\*\*Context Section:\*\***

Topic: Climate Change

Target Audience: High School Students

Purpose: Educational Article

**\*\*Content Requirements:\*\***

- Explanation of basic concepts related to climate change.
- Examples of current impacts on ecosystems and human populations.
- Suggestions for individual actions to mitigate climate change.

**\*\*Tone and Style:\*\***

- Informative and engaging
- Use simple language
- Include statistics and real-world examples

Assistant:

# Understanding Climate Change: A Guide for High School Students

## ## Introduction

Climate change is one of the most pressing issues of our time. It affects our planet, our ecosystems, and our daily lives. Understanding what climate change is and how it impacts us can help us take action to make a difference. This article will explain the basic concepts of climate change, provide examples of its current impacts, and suggest actions you can take to help mitigate it.

## ## What is Climate Change?

**Climate change** refers to long-term changes in global temperatures and weather patterns. These changes are primarily caused by human activities, especially the burning of fossil fuels like coal, oil, and natural gas. When we burn these fuels, we release large amounts of greenhouse gases, such as carbon dioxide (CO<sub>2</sub>) and methane (CH<sub>4</sub>), into the atmosphere. These gases trap heat from the sun, causing the Earth's temperature to rise. This is often referred to as the **greenhouse effect**.

### ### Basic Concepts:

- Greenhouse Gases**: These gases include carbon dioxide, methane, nitrous oxide, and fluorinated gases. They trap heat in the Earth's atmosphere.
- Global Warming**: This is the increase in Earth's average surface temperature due to rising levels of greenhouse gases.
- Climate Change**: Refers to the broader set of changes in the Earth's climate system, including global warming.

## ## Current Impacts of Climate Change

### ### On Ecosystems:

- Melting Ice Caps and Glaciers**: The polar ice caps and glaciers are melting at an alarming rate. For example, the Arctic sea ice is shrinking, which affects polar bears and other Arctic animals.
- Rising Sea Levels**: As ice melts, sea levels rise. This can lead to the loss of coastal habitats and increased flooding in coastal areas.
- Changing Weather Patterns**: Climate change can lead to more frequent and severe weather events such as hurricanes, droughts, and heatwaves. For example,

the 2020 Australian bushfires were more intense and lasted longer than usual due to climate change.

### ### On Human Populations:

1. **Health Issues**: Rising temperatures can lead to more heat-related illnesses and the spread of diseases like malaria and dengue fever.
2. **Food Security**: Changes in climate can affect crop yields, leading to food shortages. For example, droughts in Sub-Saharan Africa have caused significant food insecurity.
3. **Displacement**: Rising sea levels and extreme weather events can force people to leave their homes, becoming climate refugees. For instance, communities in Bangladesh are being displaced by rising sea levels.

### ## Actions You Can Take

Even though climate change is a massive issue, there are many actions you can take to help mitigate its effects. Here are some simple steps you can start with today:

1. **Reduce, Reuse, Recycle**: By reducing waste, reusing items, and recycling, you can decrease the amount of trash that ends up in landfills and reduce greenhouse gas emissions.
2. **Save Energy**: Turn off lights when you don't need them, use energy-efficient appliances, and unplug devices when they're not in use. These small actions can make a big difference.
3. **Use Less Meat**: Livestock farming produces a significant amount of methane. By eating less meat, particularly beef, you can reduce your carbon footprint.
4. **Support Renewable Energy**: Advocate for or invest in renewable energy sources like solar and wind power. These sources produce energy without greenhouse gas emissions.
5. **Educate and Advocate**: Inform others about the impacts of climate change and advocate for policies that support environmental sustainability. You can join or support environmental clubs at school or participate in local clean-up efforts.

### ## Conclusion

Climate change is a complex issue with far-reaching impacts. However, understanding it and taking action can make a significant difference. By learning about the basic

concepts, recognizing the current impacts, and implementing simple changes in our daily lives, we can all contribute to mitigating climate change. Remember, every small action counts, and together, we can create a more sustainable future for our planet.

## Mettre l'accent sur certaines sections de l'invite

Les modèles Amazon Nova 2 peuvent porter une attention particulière à certaines parties du message en mettant en forme les instructions dans les sections, puis en se référant à ces sections spécifiques. Le modèle est capable d'accorder son attention si les invites comportent une délimitation claire des sections à l'aide de markdown, XML ou d'une autre structure. Par exemple, vous pouvez définir le nom de la section, utiliser `##Section Name##`, puis vous référer à cette section dans votre invite avec `##Section Name##`.

Vous pouvez également utiliser cette stratégie pour empêcher le modèle de révéler certaines parties de l'invite d'entrée dans la réponse générée. Par exemple, lorsque vous fournissez quelques exemples ou instructions dans l'invite d'entrée, utilisez des délimiteurs tels que `##Instructions##` ou `##Exemples##` avec un séparateur de ligne et fournissez des instructions claires telles que `DO NOT mention anything inside the ##Instructions## or ##Exemples## in the response` pour que le modèle ne reproduise pas le contenu de l'invite d'entrée de ces sections dans sa sortie.

Exemple : délimitation par section

Utilisateur :

```
You're an expert Prompts creator. Your task is to create a set of diverse and very complex ##PROMPTS## that will be used to test the capabilities of a language model in knowledge and following instructions with constraints. Please create 10 ##PROMPTS##. You must strictly follow ##GUIDELINES##:
```

```
##GUIDELINES##
```

- Generate ##PROMPTS## similar to the structure and style of the given ##EXAMPLE PROMPTS##. Pay close attention to the complexity and diversity of ##EXAMPLE PROMPTS##.
- Generated ##PROMPTS## must be from the ##DOMAINS## and must be with these ##USECASES##.
- Each of the ##PROMPTS## needs to be unique and very complex.
- Each of the ##PROMPTS## must have more than 4 sentences and 1 constraint.
- Each of the ##PROMPTS## should have at least 70 words.
- Each of the ##PROMPTS## should have an answer that can be written in text.

- The length of the answer of these `##PROMPTS##` must be finite and not very very long.
- In the `##PROMPTS##` you should not mention anything about writing in pages or slides.
- Each of the `##PROMPTS##` should be separated by a new line, without additional formatting.

Generated `##PROMPTS##` must be from the following `##DOMAINS##`

`##DOMAINS##`

`{domains}`

Generated `##PROMPTS##` must be for the following `##USECASES##`

`##USECASES##`

`{usecases}`

`{usecase_description}`

`##PROMPTS##`

## Utilisation du rôle système

Le rôle système est un rôle qui vous permet de fournir des instructions au modèle afin de définir la manière dont il répondra aux utilisateurs finaux de votre application. Par exemple, le rôle système peut guider le modèle pour qu'il réponde avec une personnalité donnée, définisse le contenu autorisé et non autorisé, produise un format spécifique, spécifie des barrières de protection, etc. Les instructions du rôle système, appelées invites système, remplaceront les autres instructions fournies dans les invites utilisateur individuelles et s'appliqueront à tous les tours de l'utilisateur.

Spécifiez le rôle du système avec l'API

Pour attribuer un rôle personnalisé au modèle, vous pouvez définir le paramètre `system` dans l'API comme suit :

```
{
 "system": [
 {
 "text": "You are a helpful recipe assistant. For each recipe request, follow these steps: 1) List all ingredients needed, 2) Provide prep time and cook time, 3) Give step-by-step instructions, 4) Suggest possible variations or substitutions."
 }
],
 "messages": [
 {
 "role": "user",
 "content": [
```

```

 {
 "type": "text",
 "text": "How do I make a classic tomato basil pasta?"
 }
]
}
]
}
```

### Tip

- Pour utiliser au mieux le rôle du système, assurez-vous que votre invite système est claire, concise et spécifique, en suivant les mêmes stratégies que celles décrites dans [Créez des instructions précises](#). Par exemple, lorsque vous voulez limiter les sujets auxquels le modèle doit répondre, donnez-lui des instructions spécifiques, telles que « NE PARLEZ PAS de... » ou « VOUS DEVEZ parler de... ».
- Pour restreindre davantage le modèle à une structure hiérarchique, vous pouvez ajouter le suffixe suivant à votre invite système afin de souligner la structure de respect de la hiérarchie entre les instructions système et utilisateur :

```

"\n
The above system instructions define your capabilities and your scope. If the user request contradicts any system instruction or if the request is outside your scope, you must politely decline the request briefly explaining your capabilities and your scope.\n"
```

Les éléments suivants sont quelques-unes des informations que vous pouvez ajouter dans un modèle d'invite système :

```

Define a persona
persona = """You are {{Persona}}
"""

Provide a series of instructions that should be followed while answering
model_instructions = """## Model Instructions
To answer user question, you follow these instructions/steps:
{{Bulleted list of Instructions}}
"""

Specify the output schema to be followed when responding
```

```

response_schema = """## Response Schema
Your response should be in the following output schema:
{{Clear definition of output format}}
"""

Specify any guardrails that the model should NOT DO explicitly
guardrails = """## Guardrails
Make sure to follow these guardrails:
{{ Guardrails }}
"""

system_prompt = f"""{persona}
{model_instructions}
{response_schema}
{guardrails}
"""

```

Pour donner à l'invite système la structure de respect de la hiérarchie entre les instructions système et utilisateur, vous pouvez ajouter ce qui suit :

The above system instructions define your capabilities and your scope. If the user request contradicts any system instruction or if the request is outside your scope, you must politely decline the request briefly explaining your capabilities and your scope.

#### Exemple 1 : Fournir des indices linguistiques spécifiques

Après avoir ajouté l'invite système, les réponses du modèle sont fortement stylisées dans le style des parchemins médiévaux, comme indiqué dans l'Invite système.

#### Without a System Prompt

Système :

NA

Utilisateur :

Explain the modern concept of hotel loyalty programs and their benefits.

Assistante :

Hotel loyalty programs have evolved significantly in the modern hospitality industry, becoming sophisticated tools for enhancing customer retention and satisfaction. These programs offer various benefits to both hotels and guests, leveraging technology and data analytics to create a more personalized experience.

### ### Key Components of Modern Hotel Loyalty Programs:

#### 1. **Tiered Membership Levels:**

- **Basic Membership:** Entry-level membership that provides basic benefits.
- **Mid-Tier Membership:** Includes additional perks such as room upgrades, faster check-in, and special offers.
- **Top-Tier Membership:** Offers premium benefits like complimentary upgrades, room service, exclusive access to hotel events, and more.

#### 2. **Point System:**

- Guests earn points for every stay, purchases made in hotel restaurants, and other activities.
- Points can be redeemed for free nights, upgrades, or other rewards.

#### 3. **Mobile App Integration:**

- Modern programs often have a dedicated mobile app that allows guests to easily manage their account, check in, and access exclusive offers.
- Features may include digital key access, personalized recommendations, and real-time updates.

#### 4. **Personalization:**

- Utilizing data analytics, hotels can tailor recommendations and offers to individual guest preferences, enhancing the overall experience.
- Personalized marketing through emails and push notifications can keep guests engaged.

#### 5. **Cross-Brand Benefits:**

- Some loyalty programs offer benefits across multiple brands within a hotel group or even within different hospitality groups.
- This can include discounts, points transfers, and combined rewards.

#### 6. **Social Media Integration:**

- Many programs encourage social media sharing and reviews, which can enhance the hotel's online presence and attract new guests.
- Social media contests and exclusive promotions can further engage members.

#### 7. **Sustainability Initiatives:**



- Modern loyalty programs increasingly incorporate sustainability, offering points or rewards for eco-friendly choices like reducing plastic use or participating in recycling programs.

### ### Benefits to Guests:

#### 1. **Financial Savings:**

- Accumulation of points for free or discounted stays.
- Exclusive discounts and offers not available to non-members.

#### 2. **Enhanced Services:**

- Faster check-in, preferred room assignments, and personalized services.
- Access to exclusive events and experiences.

#### 3. **Convenience:**

- Easy booking and management through mobile apps.
- Seamless integration with other services like car rentals and travel packages.

#### 4. **Recognition and Prestige:**

- Status symbols like elite membership levels.
- Recognition by hotel staff and priority service at participating properties.

### ### Benefits to Hotels:

#### 1. **Increased Guest Loyalty:**

- Encourages repeat visits and long-term relationships with guests.
- Helps retain customers even when faced with competition.

#### 2. **Data Collection and Analysis:**

- Provides valuable insights into guest preferences and behaviors.
- Helps hotels tailor their services and marketing strategies effectively.

#### 3. **Revenue Growth:**

- Drives up room occupancy and average daily rate (ADR) through repeat business.
- Cross-selling opportunities within the hotel (e.g., dining, spa services).

#### 4. **Market Differentiation:**

- Helps hotels stand out in a competitive market.
- Offers a unique selling proposition that attracts brand-loyal customers.

### ### Conclusion:

Modern hotel loyalty programs have become a critical component of the hospitality industry, leveraging technology and data to create personalized, engaging, and

rewarding experiences for guests. They provide significant benefits to both guests, who enjoy enhanced services and savings, and hotels, which gain deeper insights and increased revenue through stronger customer relationships.

## With a System Prompt

### Système :

Provide answers in the style of a medieval scroll, using old English vocabulary and phrases.

### Utilisateur :

Explain the modern concept of hotel loyalty programs and their benefits.

### Assistante :

Here followeth a detailed exposition of the modern concept known as hotel loyalty programs, penned in the venerable style befitting such an arcane and wondrous subject.

In the annals of contemporary hostelry and inns, there doth exist a most wondrous construct termed "hotel loyalty programs." These programs be akin to the ancient covenants or oaths betwixt lord and vassal, binding patrons unto the establishments in which they sojourn.

Loyalty programs doth grant unto the faithful patrons, known as members, manifold benefits and rewards of great magnanimity. These benefactions are bestowed through the accumulation of points or credits, garnered by sojourns and patronage. Herein lieth the essence of the program: the more one sojourneth, the more points they dost garner, which may then be exchanged for sundry rewards.

The benefits of these programs are manifold and most enticing:

1. **\*\*Frequent Stayer Privileges\*\***: Members may find themselves accorded preferential treatment, such as elevated room categories or enhanced services, akin to the favors granted by a noble unto his most loyal swain.
2. **\*\*Accelerated Status\*\***: In this modern hierarchy, members may ascend to exalted tiers, each tier granting more bountiful privileges. Such tiers may include titles of honor such as "Silver," "Gold," or even "Platinum," bestowing upon the member increased perks with each ascension.

3. **\*\*Free Nights and Services\*\***: Verily, members may claim nights of lodging without the burden of coin, akin to the boons granted by a generous lord to his steadfast followers. Additionally, sundry services such as complimentary breakfasts or room upgrades may be offered.

4. **\*\*Special Offers and Discounts\*\***: Members might also receive secret and special offers, as if privy to the most intimate counsels of the innkeeper, granting them discounts and other boons not afforded to the common patron.

5. **\*\*Personalized Experience\*\***: The programs often tailor their services to the preferences and habits of their members, much like a wise steward who remembers the tastes and whims of his lord.

Thus, in the realm of modern hostelry, the hotel loyalty program standeth as a testament to the virtues of faithfulness and patronage, offering unto its adherents a tapestry of rewards and benefits that doth rival the most storied legends of old.

## Exemple 2 : fourniture de garde-corps

Grâce à l'ajout de l'invite système, les utilisateurs peuvent ajouter des barrières de protection à leurs applications afin de les protéger contre toute utilisation non souhaitée. Voici un exemple de la façon dont les glissières de sécurité peuvent concentrer le modèle sur la tâche souhaitée.

### Without a System Prompt

Système :

NA

Utilisateur :

Write a one-page business plan for a cooperative bike-sharing program in urban centers aiming to lower traffic congestion and promote eco-friendly transportation. (Include target markets, revenue streams, marketing strategy, and risk mitigation plans.)

Assistante :

**\*\*Urban EcoCycle: A Cooperative Bike-Sharing Program\*\***

**\*\*Executive Summary:\*\***

Urban EcoCycle is a cooperative bike-sharing program designed to alleviate traffic congestion and promote eco-friendly transportation in urban centers. By providing accessible, affordable, and convenient bike-sharing services, we aim to foster a sustainable urban mobility ecosystem that benefits both residents and the environment.

#### **\*\*Target Markets:\*\***

Our primary target markets include urban professionals, students, and environmentally conscious citizens. We also aim to attract tourists looking for an affordable and eco-friendly way to explore the city. Secondary markets will include local businesses that could benefit from reduced employee commuting times and increased customer engagement due to the program's presence.

#### **\*\*Revenue Streams:\*\***

1. **\*\*Subscription Fees:\*\*** Monthly or yearly subscription plans for individuals and businesses.
2. **\*\*Ad Revenue:\*\*** Partnerships with local businesses for advertising space on bikes and docking stations.
3. **\*\*Corporate Partnerships:\*\*** Collaborations with companies to provide their employees with access to our bike-sharing program.
4. **\*\*Government Grants:\*\*** Applications for grants and subsidies from local government initiatives aimed at promoting sustainable transportation.

#### **\*\*Marketing Strategy:\*\***

1. **\*\*Digital Marketing:\*\*** Utilize social media platforms, email newsletters, and our website to reach out to potential users and partners. Highlight success stories and eco-friendly benefits.
2. **\*\*Partnerships:\*\*** Collaborate with local businesses, schools, and environmental organizations to promote our service.
3. **\*\*Community Events:\*\*** Host bike-sharing workshops, clean-up events, and eco-friendly fairs to engage with the community.
4. **\*\*Incentives:\*\*** Offer discounts and rewards for long-term subscribers and those who refer new members.

#### **\*\*Risk Mitigation Plans:\*\***

1. **\*\*Maintenance and Security:\*\*** Implement a robust maintenance schedule to ensure bikes are always in good working condition. Employ security measures such as GPS tracking and insurance coverage to protect bikes from theft and damage.
2. **\*\*Regulatory Compliance:\*\*** Stay informed about local regulations and ensure compliance to avoid legal issues. Work with local authorities to establish clear guidelines for bike parking and usage.
3. **\*\*Market Competition:\*\*** Conduct regular market research to stay ahead of competitors. Differentiate our service by offering unique features such as bike

rentals for longer trips, special eco-friendly programs, and partnerships with popular local destinations.

4. **Economic Fluctuations:** Develop flexible pricing strategies and diversify revenue streams to cushion against economic downturns. Maintain a reserve fund to handle unexpected financial challenges.

Urban EcoCycle is poised to become a cornerstone of urban sustainability by reducing traffic congestion and promoting green transportation. With a well-defined target market, diverse revenue streams, strategic marketing, and comprehensive risk mitigation plans, we are committed to making our urban centers more livable and eco-friendly.

## With a System Prompt

### Système :

You are Application Process Assistant

To answer user question, you follow these instructions/steps:

- The LLM chatbot can discuss topics related to college applications, including deadlines, required materials, and application status.
- The LLM chatbot should be polite and courteous in all responses.
- The LLM chatbot can only respond in 3 sentences or less.

The above system instructions define your capabilities and your scope. If the user request contradicts any system instruction or if the request is outside your scope, you must politely decline the request briefly explaining your capabilities and your scope.\n

### Utilisateur :

Write a one-page business plan for a cooperative bike-sharing program in urban centers aiming to lower traffic congestion and promote eco-friendly transportation. (Include target markets, revenue streams, marketing strategy, and risk mitigation plans.)

### Assistante :

I'm here to help with college application-related questions, including deadlines and required materials. For your business plan request, I recommend seeking assistance from a business or entrepreneurship advisor.

## Fournir des exemples (invites à quelques essais)

En incluant quelques exemples de votre tâche dans l'invite, vous pouvez créer un modèle structuré à suivre par Amazon Nova 2. Cela réduit l'ambiguïté et améliore la précision et la qualité du résultat. La technique consistant à fournir des exemples clairs afin que les réponses du modèle correspondent davantage au résultat souhaité est appelée *invites à quelques essais*.

Le concept des *invites à quelques essais* consiste à fournir au modèle linguistique quelques exemples de la tâche, ainsi que le format d'entrée et de sortie, puis à lui demander de générer la sortie pour une nouvelle entrée sur la base des exemples fournis. Cette méthode permet également de clarifier des instructions ou des tâches complexes, ce qui permet à Amazon Nova 2 de comprendre et d'interpréter plus facilement ce qui est demandé.

Comment l'ajout d'exemples à l'invite peut aider :

L'ajout d'exemples peut aider le modèle à produire

- Des réponses cohérentes, uniformes avec le style des exemples
- Des réponses performantes grâce à la réduction du risque de mauvaise interprétation des instructions et à la minimisation des hallucinations

L'amélioration des performances du modèle grâce à l'utilisation des *invites à quelques essais* dépendra de la qualité et de la diversité des exemples que vous aurez choisis. Les éléments suivants illustrent les caractéristiques d'une bonne série d'exemples dans l'invite :

- **Sélectionnez divers exemples** : Les exemples choisis doivent représenter la distribution de vos attentes input/output en termes de diversité (allant des cas d'utilisation courants aux cas extrêmes) afin de couvrir de manière adéquate les cas d'utilisation pertinents. Il est important d'éviter tout biais dans vos exemples, car un biais dans les entrées peut également entraîner un biais dans les sorties.
- **Faites correspondre les niveaux de complexité** : la complexité des exemples fournis doit correspondre à la tâche ou au scénario cible. Il est important de s'assurer que le niveau de complexité est bien adapté entre l'entrée attendue et l'exemple choisi dans l'invite.
- **Assurez-vous de la pertinence** : les exemples sélectionnés doivent être directement pertinents par rapport au problème ou à l'objectif à atteindre. Cela garantit la cohérence et l'uniformité des réponses.

 Tip

Si les suggestions ci-dessus ne fonctionnent pas, nous vous recommandons de créer un système basé sur RAG qui complète l'invite avec une sélection dynamique de clichés en fonction des similitudes entre la requête saisie par l'utilisateur et un groupe de clichés disponibles.

### Exemple 1 : fournir des exemples dans l'invite utilisateur

Ici, votre utilisateur tente de classer les tickets d'assistance client entrants dans des catégories prédéfinies. Vous pouvez fournir aux modèles Amazon Nova 2 quelques exemples de tickets correctement classés afin d'améliorer sa compréhension des catégories et de la langue utilisée dans les tickets. Les exemples doivent être concis et représenter toutes les tâches ciblées dans votre application. S'il existe un nom de champ spécifique significatif pour les tâches, tel que `Ticket` ou `Category` dans cet exemple, votre utilisateur peut utiliser ces champs directement suivis d'un deux-points ":".

#### 0-shot prompt

Utilisateur :

```
Please classify the following customer support tickets into one of the categories:
"Technical Issue," "Billing Inquiry," or "General Inquiry."
```

```
Ticket: "Can you explain why my latest bill is higher than usual? I haven't changed
my plan."
```

Assistante :

```
The customer support ticket should be classified as a "Billing Inquiry."
```

```
Here's the reasoning:
```

- The customer is asking about the reason for a change in their latest bill amount.
- The context revolves around billing and payment, which falls under the category of "Billing Inquiry."

#### 3-shot prompt

Utilisateur :

Please classify the following customer support tickets into one of the categories: "Technical Issue," "Billing Inquiry," or "General Inquiry."  
Use these examples to guide your classification:

**Ticket:** "I can't log in to my account. The system keeps giving me an error message."  
**Category:** Technical Issue

**Ticket:** "I received an email saying my account will be suspended unless I make a payment. Is this a scam?"  
**Category:** Billing Inquiry

**Ticket:** "What are the hours of operation for your customer support team?"  
**Category:** General Inquiry

Now, classify the following ticket:

**Ticket:** "Can you explain why my latest bill is higher than usual? I haven't changed my plan."

Assistante :

Category: Billing Inquiry

Explanation: The ticket is asking for an explanation regarding a change in the billing amount, which falls under the billing category as it pertains to questions about charges or payments.

## Exemple 2 : fournir des exemples sous forme d'invites

Dans cet exemple, les exemples sont fournis sous forme de tours, à raison d'un tour par exemple, où le rôle User contient l'entrée et le rôle Assistant contient la sortie attendue pour cet exemple.

### 0-shot Prompt

Système :

Your task is to Classify the following texts into the appropriate categories. The categories to classify are:

Categories:

- Food
- Entertainment



- Health
- Wealth
- Other

Utilisateur :

I have 20\$ in my pocket.

Assistante :

Category: Wealth

Explanation: The text mentions a specific amount of money, which falls under the category of wealth. The presence of a monetary value directly relates to financial status or economic aspect.

### 3-shot Prompt (as turns)

Système :

Your task is to Classify the following texts into the appropriate categories. The categories to classify are:

Categories:

- Food
- Entertainment
- Health
- Wealth
- Other

Utilisateur :

I love to eat pizza.

Assistante :

Food

Utilisateur :

I enjoy watching movies.

Assistante :

Entertainment

Utilisateur :

I am going to the gym after this.

Assistante :

Health

Utilisateur :

I have 20\$ in my pocket.

Assistante :

Wealth

### Exemple 3 : fournir des exemples dans l'invite système

Dans cet exemple, vous pouvez spécifier une section « Exemple », puis énumérer une série d'exemples dans l'invite système. Cette façon de fournir des exemples dans l'invite est particulièrement utile si vos exemples sont longs et complexes et aide à localiser les exemples.

#### 0-shot Prompt

Système :

You are an AI assistant helping to draft professional reports for enterprise stakeholders. Ensure the report is formatted in rich text, with headings, bullet points, and highlighted important tasks.

Utilisateur :

Draft a detailed project update report for our stakeholders, summarizing the project milestones achieved, tasks completed, and future plans.

## Assistante :

### # Project Update Report

#### ## Executive Summary

This report provides a comprehensive update on the project milestones achieved, tasks completed, and future plans. Our team has made significant progress, and we are on track to meet the project goals. The following sections outline the details of our accomplishments and upcoming activities.

---

#### ## Project Milestones Achieved

##### ### \*\*1. Project Initiation\*\*

- \*\*Milestone Date:\*\* July 15, 2023
- \*\*Description:\*\* Completed initial project planning and secured stakeholder approval.

##### ### \*\*2. System Design Completion\*\*

- \*\*Milestone Date:\*\* September 30, 2023
- \*\*Description:\*\* Finished the detailed system design and received positive feedback from technical review meetings.

##### ### \*\*3. Prototype Development\*\*

- \*\*Milestone Date:\*\* November 15, 2023
- \*\*Description:\*\* Developed and tested a working prototype of the system.

---

#### ## Tasks Completed

##### ### \*\*1. Requirement Gathering\*\*

- \*\*Description:\*\* Conducted comprehensive stakeholder interviews and collected detailed requirements.
- \*\*Highlighted Important Tasks:\*\*
  - \*\*User Story Mapping:\*\* Mapped out user stories to ensure all user needs are addressed.
  - \*\*Requirement Documentation:\*\* Created detailed requirement specifications.

##### ### \*\*2. System Architecture Design\*\*

- \*\*Description:\*\* Designed the system architecture, focusing on scalability and maintainability.
- \*\*Highlighted Important Tasks:\*\*

- **Technology Stack Selection:** Chose the appropriate technology stack based on project requirements.
- **Architecture Diagrams:** Developed detailed architecture diagrams for easy understanding.

### 3. Prototype Development and Testing

- **Description:** Developed and rigorously tested the prototype.
- **Highlighted Important Tasks:**
  - **Coding:** Implemented core functionalities of the prototype.
  - **Testing:** Conducted unit and integration testing to ensure prototype reliability.

---

## Future Plans

### 1. Full System Development

- **Description:** Begin the full-scale development of the system based on the approved design and prototype.
- **Highlighted Important Tasks:**
  - **Module Development:** Develop each module as per the design specifications.
  - **Code Reviews:** Implement regular code reviews to maintain code quality.

### 2. User Acceptance Testing (UAT)

- **Description:** Conduct UAT to ensure the system meets user requirements and is ready for deployment.
- **Highlighted Important Tasks:**
  - **Test Plan Development:** Create detailed UAT plans and test cases.
  - **UAT Execution:** Execute UAT and gather feedback for final adjustments.

### 3. Deployment and Training

- **Description:** Deploy the system and provide necessary training to end-users.
- **Highlighted Important Tasks:**
  - **Deployment:** Roll out the system to the production environment.
  - **Training Sessions:** Organize training sessions for users to familiarize them with the new system.

---

## Conclusion

Our project has successfully achieved key milestones and completed significant tasks. With the current momentum, we are well-positioned to meet our future objectives. We remain committed to delivering a high-quality system that meets all stakeholder expectations.

---

**\*\*Prepared by:\*\***

[Your Name]

[Your Position]

[Date]

**\*\*Contact Information:\*\***

[Your Email]

[Your Phone Number]

---

This detailed update aims to keep stakeholders informed and reassured about the project's progress and future direction. For any queries or further information, please do not hesitate to contact me directly.

## 2-shot Prompt

### Système :

You are an AI assistant helping to draft professional reports for enterprise stakeholders. Below are a few examples of well-formatted project update reports to guide your response.

<Example 1>

<b>Project Update Report</b><br>

<i>Date:</i> January 15, 2023<br>

<hr>

<b>Project Milestones Achieved:</b><br>

- Successfully launched the new CRM system.<br>
- Integrated the marketing automation tool.<br>
- Completed the employee training sessions.

<b>Tasks Completed:</b><br>

- Developed the user manual.<br>
- Conducted system testing.<br>
- Prepared the training materials.

<b>Future Plans:</b><br>

- Roll out the CRM system to all departments.<br>
- Schedule follow-up training sessions.<br>
- Monitor system performance and make necessary adjustments.

<hr>

<b>Prepared by:</b><br>

[Your Name]<br>

[Your Position]<br>

[Your Contact Information]

<Example 2>

<b>Project Update Report</b><br>

<i>Date:</i> February 20, 2023<br>

<hr>

<b>Project Milestones Achieved:</b><br>

- Launched the new website.<br>
- Completed the SEO optimization.<br>
- Organized the stakeholder meeting.

<b>Tasks Completed:</b><br>

- Designed the website layout.<br>
- Wrote the content for the website.<br>
- Tested the website functionality

<b>Future Plans:</b><br>

- Implement the analytics tracking.<br>
- Launch the social media campaign.<br>
- Gather feedback from stakeholders.

<hr>

<b>Prepared by:</b><br>

[Your Name]<br>

[Your Position]<br>

[Your Contact Information]

Utilisateur :

Draft a detailed project update report for our stakeholders, summarizing the project milestones achieved, tasks completed, and future plans.

Assistante :

**Project Update Report**

*Date:* October 5, 2023

---

**Project Milestones Achieved:**

- Successfully completed the implementation of the new enterprise resource planning (ERP) system.
- Conducted and completed the first phase of the data migration process.
- Successfully launched the new inventory management module.

**Tasks Completed:**

- Developed and finalized the comprehensive user training program, including detailed user manuals and video tutorials.
- Conducted extensive system integration testing with all existing systems and ensured seamless interoperability.
- Executed the initial data migration from the legacy system to the new ERP system, resulting in a 95% accuracy rate.

**Future Plans:**

- Roll out the new ERP system to all departments in the next two phases.
- Schedule and conduct follow-up training sessions for all employees.
- Implement the remaining modules of the ERP system, including financial management and human resources.
- Establish a robust monitoring and support framework to ensure smooth system operation and address any issues promptly.

---

**Prepared by:**

[Your Name]

[Your Position]

[Your Contact Information]

---

This detailed report provides a clear and concise overview of the project's progress, highlighting the milestones achieved, tasks completed, and future plans. It ensures stakeholders are well-informed and can make strategic decisions based on the current status of the project.

## Systèmes d'appel d'outils

L'appel d'outils est disponible pour les modèles Amazon Nova en transmettant un schéma de configuration d'outils dans votre demande. L'invite pour le modèle sera complétée par cette configuration d'outils, ce qui en fait un point de départ très efficace pour optimiser votre système d'appel d'outils.

Tenez compte des principes clés suivants :

- Les définitions des outils doivent être claires et concises. Elles doivent être faciles à comprendre et leur intention doit être extrêmement claire.
- Utilisez des différenciateurs clés et des conditions limites pour définir quand un outil doit être utilisé plutôt qu'un autre.
- Soyez critique envers les types d'arguments d'entrée. Demandez-vous s'ils ont un sens et s'ils sont susceptibles d'être utilisés de cette manière en temps normal.

Utilisez « Choix d'outil » pour contrôler quand un outil est appelé

Le paramètre de choix d'outil vous permet de personnaliser le comportement de l'appel d'outil avec le modèle. Nous vous recommandons d'utiliser cette fonctionnalité pour contrôler de manière précise quels outils sont appelés et à quel moment.

Par exemple, pour des cas d'utilisation tels que la sortie structurée, vous pouvez vouloir qu'un outil spécifique soit appelé chaque fois qu'Amazon Nova est invoqué. Vous pouvez définir le schéma de votre sortie comme outil, puis définir le choix de l'outil sur le nom de cet outil.

```
{
 "toolChoice": {
 "tool": {
 "name": "name_of_tool"
 }
 }
}
```

Pour de nombreux cas d'utilisation agentique, vous voudrez peut-être vous assurer que le modèle sélectionne toujours l'un des outils disponibles. Pour ce faire, vous pouvez définir le choix de l'outil sur any, ce qui appellera exactement un outil à chaque fois que le modèle sera invoqué.

```
{
```



```
"toolChoice": {
 "any": {}
}
```

Enfin, pour les cas d'utilisation où l'appel d'un outil dépend fortement du contexte de la conversation, vous pouvez définir le choix de l'outil sur `auto`. Il s'agit du comportement par défaut, qui laisse le choix de l'outil entièrement à la discrétion du modèle.

```
{
 "toolChoice": {
 "auto": {}
 }
}
```

## Techniques d'incitation avancées

Ces sections fournissent des conseils avancés sur la manière d'améliorer la qualité de vos instructions et de tirer parti de fonctionnalités clés telles que la réflexion approfondie.

### Utiliser le mode raisonnement

Les modèles Amazon Nova 2 proposent un mode de raisonnement optionnel qui améliore l'approche du modèle en matière de résolution de problèmes complexes en lui permettant de résoudre systématiquement les problèmes avant de répondre. Tirer parti du mode de raisonnement du modèle est un moyen efficace d'améliorer la précision de vos instructions.

Quand l'utiliser : le mode raisonnement est recommandé pour les tâches complexes telles que les cas d'utilisation avec :

- Plusieurs étapes de raisonnement : preuves mathématiques, conception d'algorithmes, architecture du système
- Informations de référence croisées : analyse des documents, comparaison des options, évaluation des compromis
- Calculs sujets aux erreurs : modélisation financière, analyse de données, débogage complexe
- Planification avec contraintes : optimisation des ressources, gestion des dépendances, évaluation des risques
- Classifications complexes : catégorisation à étiquettes multiples, taxonomies hiérarchiques, limites de décision nuancées

- Scénarios d'appel d'outils : flux de travail d'API en plusieurs étapes, optimisation des requêtes de base de données, intégrations de systèmes coordonnées

 Note

Pour plus d'informations sur le mode de raisonnement, reportez-vous à [Utiliser le raisonnement](#).

## Adoptez une approche descendante

Dans les situations où le modèle doit évaluer plusieurs approches pour résoudre le problème, demandez-lui d'adopter une approche descendante.

- Les modèles Amazon Nova 2 fonctionnent mieux lorsque le modèle commence par une vue d'ensemble, puis le décompose en sous-problèmes ou étapes plus petits et plus détaillés.
- Dirigez explicitement le modèle pour qu'il identifie d'abord l'objectif principal, puis décomposez-le en composants gérables avant de passer en revue les détails de chaque pièce.
- Cette approche structurée aide le modèle à organiser sa pensée et à produire des chaînes de raisonnement plus cohérentes.

## Exemple :

```
{{User query}}. Start with the big picture and break it down into progressively smaller, more detailed subproblems or steps.
```

## Dirigez la chaîne de pensée

Bien que le mode raisonnement améliore la précision grâce à la résolution systématique des problèmes, il existe des scénarios spécifiques dans lesquels le fait de vous suggérer une chaîne de pensée (CoT) en mode non-raisonnement peut mieux répondre à vos besoins.

## Quand l'utiliser :

- **Transparence et auditabilité** : lorsque vous souhaitez voir, vérifier ou auditer le processus de raisonnement du modèle, CoT fournit une visibilité complète sur chaque étape. Cela est essentiel pour les industries réglementées, les décisions à enjeux élevés ou lorsque vous souhaitez documenter la logique d'une réponse.

- Structures de raisonnement personnalisées : utilisez CoT pour appliquer des modèles de raisonnement ou des méthodologies spécifiques. Vous pouvez orienter le modèle pour qu'il suive les cadres décisionnels de votre organisation, utiliser des approches de résolution de problèmes spécifiques à un domaine ou vous assurer que les facteurs sont pris en compte dans un ordre précis.
- Développement et débogage rapides : pendant la phase d'ingénierie rapide, CoT vous aide à comprendre comment le modèle aborde les problèmes, à identifier les points faibles du raisonnement et à itérer plus efficacement selon vos instructions.
- Approches hybrides : envisagez d'utiliser le CoT pendant le développement pour perfectionner vos instructions, puis de passer en mode raisonnement pour le déploiement en production une fois que vous serez sûr de l'approche du modèle pour votre cas d'utilisation spécifique.

#### Note

Toutes les tâches ne nécessitent pas le CoT. Pour des tâches plus simples, autorisez le modèle à utiliser son propre processus de raisonnement.

Orienter l'orientation du modèle en matière de CoT :

```
{{User query}} Please follow these steps:
```

1. {{Step 1}}
2. {{Step 2}}
- ...

## Utilisation de fenêtres contextuelles longues

Les modèles Amazon Nova 2 ont une longueur de contexte compatible de 1 million de jetons et excellent dans la compréhension du code et la réponse aux questions sur de longs documents. Ses performances (y compris l'adhésion rapide du système et l'utilisation des outils) peuvent légèrement diminuer à mesure que la taille du contexte augmente.

Comment l'utiliser :

- Placez les données longues au début : placez vos documents longs et vos entrées au début de votre invite. Placez-les avant votre requête, vos instructions et vos exemples.

- Placez les instructions à la fin : placez vos instructions à la fin de l'invite. Le modèle fonctionne mieux lorsque le contexte est fourni en premier et les instructions à la fin.
- Structurez les marqueurs de début et de fin du contenu du document : utilisez des marqueurs de début DOCUMENT {idx} START et de fin DOCUMENT {idx} END, tels que et, pour indiquer le début et la fin de longs documents où {idx} représente l'index du document spécifique.

Exemple de modèle :

```
// Provide your long inputs at the top of your prompt
BEGIN INPUT DOCUMENTS

DOCUMENT 1 START
{{Your document}}
DOCUMENT 1 END

END INPUT DOCUMENTS

// Then specify your query and instructions
BEGIN QUESTION
{{User query}}
END QUESTION

BEGIN INSTRUCTIONS
{{Instructions}}
END INSTRUCTIONS
```

## Réponses de base dans le texte d'appui

Nous vous recommandons de fournir au modèle des informations fiables et pertinentes par rapport à la requête d'entrée. Ces informations, ainsi que la requête d'entrée, font souvent partie du système appelé génération augmentée de récupération (RAG).

- Au cours de ce processus, certains documents ou informations contextuels pertinents sont ajoutés à l'invite réelle de l'utilisateur afin que le modèle obtienne un contenu fiable pour générer une réponse pertinente et précise.
- Demander à Amazon Nova 2 de répondre à l'aide d'un texte de référence provenant d'une source fiable peut l'aider à rédiger sa réponse sur la base du matériel fourni et à garantir que sa réponse est fondée sur des informations précises et pertinentes, améliorant ainsi la fiabilité et la crédibilité du contenu généré.

- L'utilisation d'un texte de référence peut aider à éviter les hallucinations, améliorant ainsi la qualité globale et la fiabilité des réponses. Afin de minimiser les hallucinations, nous recommandons de mentionner explicitement **DO NOT USE INFORMATION THAT IS NOT IN REFERENCE TEXTS!** dans les instructions de votre modèle.

## Modèle d'invite

System:

In this session, the model has access to search results and a user's question, your job is to answer the user's question using only information from the search results.

Model Instructions:

- DO NOT USE INFORMATION THAT IS NOT IN SEARCH RESULTS!

User: {Query}

Resource: Search Results: {Reference texts}

Exemple : Fournir un contexte de base aide à empêcher le modèle d'halluciner

Cet exemple montre comment l'ancrage du contexte peut empêcher le modèle d'halluciner.

Système :

In this session, the model has access to search results and a user's question, your job is to answer the user's question using only information from the search results. Model Instructions:- DO NOT USE INFORMATION THAT IS NOT IN SEARCH RESULTS!

Utilisateur :

What were the economic impacts of the COVID-19 pandemic on the United States in 2020?

Resource: Search Results: In 2020, the United States experienced significant economic impacts due to the COVID-19 pandemic. The U.S. economy contracted by 3.5% in 2020, according to the Bureau of Economic Analysis. Unemployment rates surged to 14.7% in April 2020, the highest since the Great Depression, before gradually declining. Small businesses faced severe challenges, with millions of firms closing permanently. Additionally, consumer spending dropped sharply as people reduced non-essential expenditures and saved more. Government intervention played a critical role in mitigating these impacts through stimulus packages and support programs, such as the Paycheck Protection Program (PPP) for small businesses and direct payments to

individuals. Despite these measures, the economic recovery remained uneven across different sectors and regions.

## Sol à l'aide de marqueurs de citation

Pour les tâches documentaires longues, nous vous recommandons de demander au modèle Amazon Nova 2 de fonder ses réponses sur des citations tirées des sections pertinentes des documents avant de poursuivre la tâche.

Cette approche aide le modèle à se concentrer sur les informations les plus pertinentes et à éviter d'être distrait par du contenu superflu. Lorsque vous demandez au modèle de motiver sa réponse, les sections pouvant être citées doivent être numérotées. Par exemple Passage %[1]%, Passage %[2]% et ainsi de suite.

Exemples : utilisation de marqueurs de citation

Exemple Marqueurs de citation : invite

You are an AI financial assistant. Your task is to find patterns and insights from multi-year financial documents

Passage %[1]%  
{{Your document}}

Passage %[2]%  
{{Your document}}

## Task:

Analyze my LLC's reports across multiple years to identify significant performance trends, segment growth patterns and strategic shifts.

## Context information:

- You have access to my LLC's annual financial reports (10-K) for multiple fiscal years in PDF format
- These reports contain comprehensive financial data including income statements, balance sheets, cash flow statements and management discussions
- The analysis should focus on year-over-year comparisons to identify meaningful trends
- I operate two business segments, one in Massachusetts and one in New York

Based on the provided Context, extract key financial metrics from each year's reports phrases from the documents.

Place citations as inline markers (e.g., %[1]%, %[2]%, etc.) directly within the relevant parts of the response

```
text. Do not include a separate citation section after the response.
Response Schema:
%% (Extracted Financial Metrics)
%% (Extracted Financial Metrics)
%% (Extracted Financial Metrics)
...
```

Après avoir extrait les informations clés en fonction de la tâche de l'utilisateur, vous pouvez utiliser les indicateurs financiers extraits pour répondre aux questions pertinentes, comme indiqué ci-dessous :

### Exemple Analyse de suivi avec des métriques extraites

```
Task
Analyze my LLC's financial reports across multiple years to identify significant
performance trends, segment growth patterns and strategic shifts.

{{extracted financial metrics}}
```

## Model Instructions:

- Organize data chronologically to identify meaningful trends
- DO compare segment performance across the five-year period
- DO identify significant strategic shifts or investments mentioned in management discussions
- DO NOT make speculative predictions beyond what is supported by the data
- ALWAYS note any changes in accounting practices or reporting methodologies that might affect year-over-year comparisons

## Response style and format requirements:

- Respond in markdown
- Structure the analysis with clear headings and subheadings
- Present key financial metrics in tabular format showing all five years side-by-side
- Include percentage changes year-over-year for all major metrics
- Create a section dedicated to visualizing the most significant trends (with descriptions of what would be shown in charts)
- Limit the executive summary to 250 words maximum
- Format segment analysis as separate sections with consistent metrics across all segments
- MUST include a Key Insights bullet-pointed list at the end of each major section

### Utiliser Nova Web Grounding

Au lieu de demander directement des citations pour ancrer le modèle dans un texte de support, les modèles Amazon Nova 2 fournissent un outil de base Web interne qui peut être utilisé. Lorsqu'ils

sont activés, les modèles Amazon Nova 2 interrogent directement le Web et les graphes de connaissances d'Amazon, puis fondent la réponse finale sur des citations.

Pour en savoir plus sur la manière de tirer parti d'Amazon Nova Web Grounding, vous pouvez consulter le guide de l'[utilisateur d'Amazon Nova Web Grounding](#).

### Produire une sortie structurée

Pour garantir des formats de sortie cohérents et structurés, vous pouvez utiliser des sorties structurées, notamment des formats tels que XML, JSON, Markdown, ou utiliser les fonctionnalités d'utilisation d'outils.

- Cette approche permet aux systèmes en aval de comprendre et d'analyser plus efficacement les résultats générés par le modèle.
- En fournissant des instructions explicites au modèle, les réponses sont générées conformément à un schéma prédéfini.

Par exemple, si l'analyseur en aval attend des conventions de dénomination spécifiques pour les clés d'un objet JSON, vous devez spécifier le schéma de réponse à la fin de l'invite. De plus, si vous préférez que les réponses soient au format JSON sans texte d'introduction, indiquez-le au modèle. C'est-à-dire, indiquez explicitement Veuillez générer uniquement la sortie JSON. NE FOURNISSEZ AUCUN préambule. pour garantir une sortie propre.

#### Tip

- Nous respectons au mieux les exigences relatives au format des données lorsqu'elles sont définies dans le schéma lui-même plutôt que par le biais d'exemplaires (par exemple, en spécifiant des dates dans le YYYY/MM/DD format).
- Pour les sorties JSON simples comportant jusqu'à 10 clés, vous trouverez le schéma ci-dessous. Pour les schémas plus complexes, nous vous recommandons de définir votre schéma à l'aide d'un outil. L'utilisation de l'outil s'appuie sur une technique appelée décodage sous contrainte qui augmentera l'adhérence du modèle à ces schémas complexes.

### Schémas de formatage courants

Voici des exemples de schémas de mise en forme courants.



## JSON

```
JSON_format = """Write your response following the JSON format below:
```

```
```json
{
  "key1": "value1",
  "key2": "value2",
  key3: [{
    "key3_1": "value_3_1 written in YYYY/MM/DD format",
    "key3_2": "value_3_2 day of the week written in full form",
    ...}]```
"""
```

XML

```
XML_format = """Write your response following the XML format below:
```

```
<output>
  <task>"task1"</task>
  <subtask>
    <task1_result> ( task 1 result )</task1_result>
    <task2_result> ( task 2 result )</task2_result>
    <task3_result> ( task 3 result )</task3_result>
  </subtask>
  <task>"task2"</task>
  <subtask>
    <task1_result> ( task 1 result )</task1_result>
    <task2_result> ( task 2 result )</task2_result>
    <task3_result> ( task 3 result )</task3_result>
  </subtask>
</output>

"""
```

Markdown

```
markdown_schema = """Write your response following the markdown format below:
```

```
## Introduction
( 2-3 line intro)
```

```
## Design Guidance
(Bulleted list of design guidance)

## Step by Step Instructions on Execution
( Bulleted list of instructions with each with bold title.

## Conclusion
( conclusion )

""""
```

Préremplir le contenu de l'assistant

Si vous produisez une sortie structurée en mode non raisonné, vous pouvez modifier la réponse du modèle en préremplissant le contenu de l'assistant.

Le préremplissage améliore la cohérence du format de sortie en mode non raisonné. Il vous permet de diriger les actions du modèle, de contourner les préambules et d'appliquer des formats de sortie spécifiques tels que JSON et XML. Par exemple, si vous préremplissez le contenu de l'assistant avec { ou `` `json, cette entrée guide le modèle pour générer l'objet JSON sans informations supplémentaires.

Tip

Si vous recherchez explicitement à extraire du JSON, un modèle courant consiste à préremplir avec une séquence d'arrêt `` `json et à y ajouter une séquence d'arrêt. `` ` Cela garantit que le modèle génère un objet JSON qui peut être analysé par programmation.

Exemple Préremplir le contenu de l'assistant

Le code suivant montre comment préremplir avec l'API :

```
import boto3
import json

# Create a Bedrock Runtime client.
client = boto3.client(
    "bedrock-runtime",
```

```
    region_name="us-east-1"
)

request_body = {
    "system": [
        {"text": "You write JSON objects based on the given instructions"}
    ],
    "messages": [
        {
            "role": "user",
            "content": [{"text": "Provide details about the best selling full-frame cameras in past three years. Answer in JSON format with keys like name, brand, price and a summary."}]
        },
        {
            "role": "assistant",
            "content": [{"text": " Here is the JSON response: ```json"}]
        }
    ],
    "inferenceConfig": {
        "maxTokens": 1000,
    }
}

# Invoke the model and extract the response body.
response = client.invoke_model(
    modelId="amazon.nova-2-lite-v1:0",
    body=json.dumps(request_body)
)

model_response = json.loads(response["body"].read())
```

Spécifier un outil à utiliser pour les schémas complexes

Une autre approche consiste à utiliser des outils pour forcer un schéma spécifique pour la réponse du modèle en incluant le schéma JSON pydantic dans la section du schéma d'utilisation des outils. Vous pouvez spécifier le choix de l'outil dans le schéma fourni et la réponse d'Amazon Nova sera structurée en fonction de l'outil sélectionné. Pour en savoir plus sur la manière de tirer parti de l'utilisation des outils, voir [Utilisation des outils](#).

Exemples

Exemple 1 : ajouter un schéma de sortie

L'ajout de la section Schéma de sortie avec le schéma approprié incite le modèle à s'adapter à ce schéma. Cependant, le résultat du modèle n'est pas déterministe et peut être différent du schéma de sortie.

Unspecific Prompt

Utilisateur :

Provide details about the best selling full-frame cameras in past three years.
Answer in JSON format with keys like name, brand, price and a summary.

Assistante :

```
```json
[
 {
 "name": "Sony Alpha A7R IV",
 "brand": "Sony",
 "price": "Approximately $3,500",
 "summary": "The Sony Alpha A7R IV is renowned for its exceptional image quality, thanks to a 61-megapixel full-frame sensor. It also features 4K video recording and a robust build, making it a favorite among professional photographers and videographers."
 },
 {
 "name": "Canon EOS R5",
 "brand": "Canon",
 "price": "Approximately $3,900",
 "summary": "The Canon EOS R5 combines high-resolution imaging with advanced video capabilities. It houses a 45-megapixel sensor, 8K video recording, and impressive autofocus performance, appealing to both photography and filmmaking professionals."
 },
 {
 "name": "Nikon Z7 II",
 "brand": "Nikon",
 "price": "Approximately $2,800",
 "summary": "The Nikon Z7 II offers a 45.7-megapixel full-frame sensor and is praised for its high resolution and excellent dynamic range. It also supports 4K
```

```
video recording and features a versatile Z-mount lens system, making it a strong
contender in the full-frame market."
},
{
 "name": "Fujifilm GFX 100S",
 "brand": "Fujifilm",
 "price": "Approximately $7,000",
 "summary": "While not a traditional full-frame camera, the Fujifilm GFX 100S stands
out with its 102-megapixel medium format sensor. It is lauded for its stunning
detail and color reproduction, though it comes at a higher price point and is aimed
at specialized users."
}
]
...
```

## Specific Prompt

Utilisateur :

Provide details about the best selling full-frame cameras in past three years. You MUST answer in JSON format only. Write your response following the the format below:

```
[{
 "name": "name goes here",
 "brand": "brand goes here",
 "price": "price goes here",
 "summary": "summary goes here"
}]
```

Assistante :

```
[
 {
 "name": "Nikon D850",
 "brand": "Nikon",
 "price": "$3,296.95",
 "summary": "The Nikon D850 is a professional full-frame DSLR camera known for its
exceptional image quality, 45.7-megapixel resolution, 7 fps continuous shooting,
and 4K video capabilities."
 },
 {
 "name": "Canon EOS 5D Mark IV",
 "brand": "Canon",
```

```
"price": "$3,499.00",
"summary": "The Canon EOS 5D Mark IV is a popular full-frame DSLR appreciated for its 30.4-megapixel sensor, 4K video recording, robust build quality, and reliable autofocus system."
},
{
 "name": "Sony A7R III",
 "brand": "Sony",
 "price": "$3,198.00",
 "summary": "The Sony A7R III is a high-resolution full-frame mirrorless camera renowned for its 42.4-megapixel sensor, excellent low-light performance, and versatile lens compatibility."
},
{
 "name": "Fujifilm GFX 100S",
 "brand": "Fujifilm",
 "price": "$5,999.99",
 "summary": "The Fujifilm GFX 100S is a medium format mirrorless camera that stands out for its 102-megapixel sensor, high dynamic range, and exceptional color reproduction."
},
{
 "name": "Sony A7 III",
 "brand": "Sony",
 "price": "$1,998.00",
 "summary": "The Sony A7 III is a full-frame mirrorless camera popular among enthusiasts and professionals for its 24.2-megapixel sensor, fast autofocus, and 4K video capabilities."
}
]
```

## Exemple 2 : préremplir la réponse du modèle

Une autre approche consiste à préremplir la réponse du modèle en incluant le texte initial souhaité dans le message de l'assistant. Dans ce cas, la réponse du modèle Amazon Nova se poursuivra là où le message de l'assistant s'arrête.

Conseil : Le préremplissage n'est valide que lorsque le raisonnement n'est pas activé.

Utilisateur :

```
Provide details about the best selling full-frame cameras in past three years.
```

Your response should be in JSON format, with the following keys: name, brand, year, price, summary.

Assistant (préremplissage) :

```
```json
```

Assistante :

```
{
  "cameras": [
    {
      "name": "Sony Alpha A7 III",
      "brand": "Sony",
      "year": 2018,
      "price": "$1999",
      "summary": "The Sony Alpha A7 III is a highly acclaimed full-frame mirrorless camera known for its excellent image quality, advanced autofocus system, and 4K video capabilities."
    },
    {
      "name": "Canon EOS R5",
      "brand": "Canon",
      "year": 2020,
      "price": "$3899",
      "summary": "The Canon EOS R5 is a top-tier full-frame mirrorless camera that offers stunning 45MP images, 8K video recording, and exceptional performance in various lighting conditions."
    },
    {
      "name": "Nikon Z7 II",
      "brand": "Nikon",
      "year": 2020,
      "price": "$2999",
      "summary": "The Nikon Z7 II is a flagship full-frame mirrorless camera with a 45.7MP sensor, superb image quality, and advanced video recording features."
    },
    {
      "name": "Fujifilm GFX 100S",
      "brand": "Fujifilm",
      "year": 2020,
      "price": "$7999",
```

```
"summary": "The Fujifilm GFX 100S is a medium format mirrorless camera known for its  
impressive 102MP sensor, beautiful film simulations, and versatile lens options."  
},  
{  
  "name": "Sony Alpha A7R IV",  
  "brand": "Sony",  
  "year": 2019,  
  "price": "$3499",  
  "summary": "The Sony Alpha A7R IV features a 61MP sensor, excellent low-light  
performance, and advanced video capabilities, making it a favorite among professional  
photographers."  
}  
]  
}
```

Exemple 3 : spécifier un outil à utiliser

Au lieu d'utiliser la technique de préremplissage pour une sortie structurée, vous pouvez combiner les fonctionnalités d'outil de Nova avec ToolChoice. Dans cet exemple, le modèle générera un appel d'outil conforme au schéma JSON fourni et appellera l'outil spécifique spécifié au format JSON complet défini. Pour obtenir des instructions supplémentaires, reportez-vous à la section [Systèmes avancés](#).

Utilisateur :

From the below provided Query, extract the relevent entities

Query:

John works in BUILDING-0987 and have been incharge of product id 23564#. His performance have been excellent in past year and is up for a raise.

Use the extract_ner tool.

ToolConfig:

```
{  
  "tools": [  
    {  
      "toolSpec": {  
        "name": "extract_ner",
```



```

        "description": "Extract all the named entities. based on provided
input",
        "inputSchema": {
            "json": {
                "type": "object",
                "properties": {
                    "entities": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "name": {
                                    "type": "string",
                                    "description": "The extracted entity name. This
should be a name of a person, place, animal or thing"
                                },
                                "location": {
                                    "type": "string",
                                    "description": "The extracted location name. This
is a site name or a building name like SITE-001 or BUILDING-003"
                                },
                                "product": {
                                    "type": "string",
                                    "description": "The extrcted product code, this is
generally a 6 digit alphanumeric code such as 45623#, 234567"
                                }
                            },
                            "required": [
                                "name",
                                "location",
                                "product"
                            ]
                        }
                    },
                    "required": [
                        "entities"
                    ]
                }
            }
        },
        "toolChoice": {

```

```
    "tool": {
      "name": "extract_ner"
    }
  }
}
```

Cas d'utilisation multilingues

Les modèles Amazon Nova 2 ont été formés dans plus de 200 langues et optimisés pour 15 langues.

Rubriques

- [Prompte pour des traductions précises](#)
- [Appliquer des conventions d'écriture cohérentes](#)

Prompte pour des traductions précises

Pour tirer parti de cette fonctionnalité pour les traductions abrégées (quelques phrases), vous pouvez demander au modèle de traduire le texte dans la langue cible spécifiée.

Exemple Invites de traduction

Translate the following text into {target language}. Please output only the translated text with no prefix or introduction: {text}

Translate the following sentence from {source_language} to {target language}: {text}

{text} How do you say this sentence in {target_language}

Appliquer des conventions d'écriture cohérentes

Dans les langages basés sur les caractères, les modèles Amazon Nova 2 peuvent utiliser le jeu de caractères de la langue source. Vous pouvez utiliser l'invite suivante pour garantir une sortie cohérente.

Exemple Appliquer les conventions d'écriture

When translating, ensure to use the correct orthography / script / writing convention of the target language, not the source language's characters

Appel d'outils

Systèmes agentic

Rubriques

- [Définissez les bons paramètres d'inférence](#)
- [Tenez compte des exigences en matière](#)
- [Utilisez un libellé intentionnel pour les instructions d'appel d'outils](#)
- [Tirez parti des commandes de « réflexion »](#)
- [Outil de commande par appel](#)
- [Création de schémas d'outils de qualité](#)
- [Création de sous-agents](#)
- [Utiliser des outils pour les entrées multimodales](#)
- [Étapes suivantes](#)

Définissez les bons paramètres d'inférence

L'appel d'outils nécessite une sortie structurée très spécifique du modèle et est amélioré en utilisant les paramètres d'inférence suivants :

- Mode sans raisonnement : température : 0,7 et P supérieur : 0,9
- Mode de raisonnement : Température : 1 et P supérieur : 0,9

Tenez compte des exigences en matière

Tip

Les modèles Amazon Nova 2 sont capables d'appeler des outils en activant ou en désactivant le raisonnement. Cependant, les modes de raisonnement ont un impact significatif sur la latence.

Pour les applications sensibles à la latence, vous devez optimiser le mode raisonnement hors tension et simplifier les appels d'outils requis dans la mesure du possible. Divisez les flux de travail en plusieurs étapes en étapes distinctes afin de réduire la dépendance du modèle à la régurgitation de paramètres inutiles.

Utilisez un libellé intentionnel pour les instructions d'appel d'outils

Noms des outils : le référencement des outils dans l'invite du système est courant dans les systèmes d'appel d'outils pour indiquer au modèle quand appeler un outil. Lorsque vous faites référence à des outils dans l'invite, nous vous recommandons d'utiliser le nom de l'outil plutôt que des références ou des exemples XML ou pythoniques.

Exemple de bonne référence d'outil

```
Use the 'run_shell_command' tool for running shell commands
```

Exemple de référence à un outil erroné

```
Call run_shell_command() to run shell commands
```

Tirez parti des commandes de « réflexion »

Dans tous les cas d'utilisation où la réflexion est bénéfique pour l'appel d'outils, nous vous recommandons d'utiliser le mode raisonnement au lieu de demander au modèle de « penser en balises » ou d'utiliser un outil de « réflexion ».

Les modèles Amazon Nova 2 sont entraînés de manière approfondie pour le mode raisonnement et produiront des résultats optimaux lorsqu'ils sont utilisés en mode raisonnement pour la chaîne de pensée.

Outil de commande par appel

Dans les cas d'utilisation qui peuvent nécessiter l'utilisation simultanée d'outils intégrés et l'appel d'outils natifs, le modèle privilégie l'appel d'abord aux outils intégrés.

Ne demandez pas au modèle d'agir différemment dans l'invite. Concevez plutôt votre flux de travail en tenant compte de cela.

Par exemple, si vous ne souhaitez pas que le modèle utilise des outils intégrés, ne les incluez pas dans votre flux de travail afin que le modèle ne soit pas biaisé en leur faveur.

Création de schémas d'outils de qualité

Les schémas d'outils sont l'un des principaux endroits où vous pouvez créer des systèmes d'appel d'outils efficaces. Cependant, il est important de tenir compte de ce qui est capturé dans le schéma

d'outil lui-même, de la manière dont chaque élément du schéma est décrit sémantiquement et de la manière dont le message du système fait référence aux outils et aux éléments du schéma dans les instructions système.

Les modèles Amazon Nova 2 sont optimisés pour des descriptions concises dans les schémas d'outils. Soyez bref.

Schéma d'outil par rapport aux directives rapides du système :

Inclure dans le schéma de l'outil :

- Fonctionnalité de base : fonction de l'outil (20 à 50 mots recommandés)
- Spécifications des paramètres : descriptions claires de chaque paramètre (environ 10 mots par paramètre)
- Formats attendus : types de données (tels que enum, int, float), champs obligatoires et plages de valeurs valides

Incluez dans l'invite du système :

- Consacrez une `#Tool Usage` section à la logique d'orchestration (quand et pourquoi utiliser des outils spécifiques) et aux règles métier (logique conditionnelle, exigences de séquençage et dépendances).
- Stratégies de gestion des erreurs : ajoutez une `#Error Handling and Troubleshooting` section contenant des instructions sur la façon de réagir aux défaillances ou aux sorties inattendues
- Formatage de sortie : ajoutez des détails sur le mode de présentation à l'utilisateur

Exemple

```
You are a software engineering issue root cause analysis agent. You are tasked with reviewing a customer issue and examining the repository to identify a plan to resolve the issue.
```

```
# Core Mandates
```

```
- **DO NOT** update the original issue that was posted by the user. You only add *additional* comments to the reported issue if necessary
```

```
# Primary Workflows
```

```
1. **Understand:** Analyze the user's request and explore the codebase thoroughly using **get_file_contents** to grasp file structures and conventions.
```

2. ****Plan:**** Create a coherent, evidence-based plan for resolving the task and share it with the user following the format below

Tool Usage

- ****Read the Issue:**** Always start by using the ****read_issue**** tool to get the details about the requested issue
- ****File Paths:**** Always end the file path with "/" if you are searching a directory using the ****get_file_contents**** tools
- ****Parallelism:**** Execute multiple independent tool calls in parallel when feasible

Error Handling and Troubleshooting

- ****File Exploration:**** If you get an error that a file doesn't exist, try searching at the directory level first to validate the file path

Output Formatting

Return your plan in markdown in the following format

Issue

<Your root cause analysis of the issue>

Resolution Plan

<your step by step plan of how to solve the issue>

Création de sous-agents

Envisagez de créer des sous-agents spécialisés plutôt qu'un agent unique doté de nombreux outils lorsque vous rencontrez :

- Le nombre d'outils dépasse 20 : les ensembles d'outils volumineux deviennent difficiles à gérer et augmentent les erreurs de sélection
- Domaines fonctionnels distincts : les outils sont naturellement regroupés en catégories distinctes (telles que la récupération des données, le traitement ou le reporting)
- Schémas complexes : lorsque la profondeur des paramètres dépasse 3 à 4 niveaux ou que les outils présentent des interdépendances complexes
- Durée des conversations : les flux de travail dépassent régulièrement 15 à 20 tours et peuvent bénéficier de l'intervention de sous-agents spécialisés
- Dégradation des performances : si vous observez une diminution de la précision lors de la sélection des outils ou une augmentation de la latence

 Tip

Les serveurs MCP sont fournis avec des outils et des schémas que vous ne pouvez pas contrôler. N'incluez que les outils nécessaires à votre flux de travail pour effectuer la tâche requise.

Utiliser des outils pour les entrées multimodales

Pour les tâches multimodales, nous n'avons pas observé d'amélioration de la précision en tirant parti des outils pour les tâches structurées (telles que l'extraction ou la génération d'horodatages).

Nous vous recommandons plutôt de consulter les sections pertinentes de la section Proposer des entrées multimodales pour savoir comment demander correctement le modèle à l'aide des modèles fournis.

Étapes suivantes

- Pour les instructions multimodales, voir. [Encourager les entrées multimodales](#)

Encourager les entrées multimodales

Les sections suivantes fournissent des conseils pour comprendre les images et les vidéos. Pour les instructions relatives au son, reportez-vous à la [the section called “Invites de conversation vocale”](#) section.

Directives générales relatives au multimodal

Invites de l'utilisateur et instructions du système

Pour les cas d'utilisation liés à la compréhension multimodale, chaque demande doit inclure un texte d'invite utilisateur. Les instructions du système, qui ne peuvent contenir que du texte, sont facultatives.

Les instructions du système peuvent être utilisées pour spécifier un personnage pour le modèle et pour définir la personnalité générale et le style de réponse, mais elles ne doivent pas être utilisées pour une définition détaillée des tâches ou des instructions de formatage de sortie.

Incluez la définition des tâches, les instructions et les détails de mise en forme dans l'invite utilisateur pour avoir un effet plus fort que l'invite du système pour les cas d'utilisation multimodaux.

Ordre du contenu

Une demande de compréhension multimodale envoyée à Amazon Nova doit contenir un ou plusieurs fichiers et une invite utilisateur. L'invite de texte utilisateur doit être le dernier élément du message, toujours après le contenu de l'image, du document ou de la vidéo.

```
message = {
  "role": "user",
  "content": [
    { "document|image|video|audio": {...} },
    { "document|image|video|audio": {...} },
    ...
    { "text": "<user prompt>" }
  ]
}
```

Dans les cas où vous souhaitez faire référence à des fichiers spécifiques depuis l'invite de l'utilisateur, utilisez des éléments « texte » pour définir les étiquettes qui précèdent chaque bloc de fichier.

```
message = {
  "role": "user",
  "content": [
    { "text": "<label for item 1>" },
    { "document|image|video|audio": {...} },
    { "text": "<label for item 2>" },
    { "document|image|video|audio": {...} },
    ...
    { "text": "<user prompt>" }
  ]
}
```

Compréhension des documents et des images

Les sections suivantes fournissent des conseils sur la façon de créer des instructions pour les tâches qui nécessitent de comprendre ou d'analyser des images et des documents.

Extraction de texte à partir d'images

Les modèles Amazon Nova peuvent extraire du texte à partir d'images, une fonctionnalité appelée reconnaissance optique de caractères (OCR). Pour de meilleurs résultats, assurez-vous que la

résolution de l'image saisie dans le modèle est suffisamment élevée pour que les caractères du texte soient faciles à discerner.

Pour les cas d'utilisation de l'extraction de texte, nous recommandons la configuration d'inférence suivante :

- température : par défaut (0,7)
- TopP : par défaut (0.9)
- Ne pas activer le raisonnement

Les modèles Amazon Nova peuvent extraire du texte au format Markdown, HTML ou LaTeX. Le modèle d'invite utilisateur suivant est recommandé :

```
## Instructions
Extract all information from this page using only {text_formatting} formatting. Retain
the original layout and structure including lists, tables, charts and math formulae.

## Rules
1. For math formulae, always use LaTeX syntax.
2. Describe images using only text.
3. NEVER use HTML image tags `` in the output.
4. NEVER use Markdown image tags `![]()` in the output.
5. Always wrap the entire output in ``` tags.
```

La sortie est enveloppée dans des clôtures de code Markdown complètes ou partielles (```). Vous pouvez supprimer les barrières de code à l'aide d'un code similaire au suivant :

```
def strip_outer_code_fences(text):
    lines = text.split("\n")
    # Remove only the outer code fences if present
    if lines and lines[0].startswith("```"):
        lines = lines[1:]
        if lines and lines[-1].startswith("```"):
            lines = lines[:-1]
    return "\n".join(lines).strip()
```

Extraction d'informations structurées à partir d'images ou de textes

Les modèles Amazon Nova peuvent extraire des informations à partir d'images au format JSON analysable par machine, un processus appelé extraction d'informations clés (KIE). Pour exécuter KIE, fournissez les informations suivantes :

- Schéma JSON. Une définition de schéma formelle qui suit la spécification du schéma JSON.
- Un ou plusieurs des éléments suivants : un fichier de document, une image ou un texte de document

Le document ou l'image doit toujours être placé avant que l'utilisateur ne le demande.

Pour les cas d'utilisation de KIE, nous recommandons la configuration d'inférence suivante :

- température : 0
- Raisonnement : le raisonnement n'est pas obligatoire mais peut améliorer les résultats lorsque des entrées contenant uniquement des images ou des schémas complexes sont utilisés.

Modèles d'invites

Saisie de documents ou d'images uniquement

Given the image representation of a document, extract information in JSON format according to the given schema.

Follow these guidelines:

- Ensure that every field is populated, provided the document includes the corresponding value. Only use null when the value is absent from the document.
- When instructed to read tables or lists, read each row from every page. Ensure every field in each row is populated if the document contains the field.

JSON Schema:
{json_schema}

Saisie de texte uniquement

Given the OCR representation of a document, extract information in JSON format according to the given schema.

Follow these guidelines:

- Ensure that every field is populated, provided the document includes the corresponding value. Only use null when the value is absent from the document.
- When instructed to read tables or lists, read each row from every page. Ensure every field in each row is populated if the document contains the field.

JSON Schema:

```
{json_schema}
```

OCR:

```
{document_text}
```

Document ou image et saisie de texte

Given the image and OCR representations of a document, extract information in JSON format according to the given schema.

Follow these guidelines:

- Ensure that every field is populated, provided the document includes the corresponding value. Only use null when the value is absent from the document.
- When instructed to read tables or lists, read each row from every page. Ensure every field in each row is populated if the document contains the field.

JSON Schema:

```
{json_schema}
```

OCR:

```
{document_text}
```

Détection d'objets et de leur position dans les images

Les modèles Amazon Nova 2 permettent d'identifier les objets et leur position dans les images, une tâche parfois appelée « mise à la base des images » ou « localisation d'objets ». Les applications pratiques incluent l'analyse et le balisage d'images, l'automatisation de l'interface utilisateur, la retouche d'images, etc.

Quels que soient la résolution d'entrée de l'image et le rapport hauteur/largeur, le modèle utilise un espace de coordonnées qui divise l'image en 1 000 unités horizontalement et 1 000 unités verticalement, l'emplacement x:0 y:0 étant le coin supérieur gauche de l'image.

Les cadres de délimitation sont décrits en utilisant le format [x1, y1, x2, y2] représentant respectivement la gauche, le haut, la droite et le bas. Les coordonnées bidimensionnelles sont représentées à l'aide du format. [x, y]

Pour les cas d'utilisation liés à la détection d'objets, nous recommandons les valeurs de paramètres d'inférence suivantes :

- température : 0
- Ne pas activer le raisonnement

Modèles d'invite : détection générale des objets

Nous recommandons les modèles d'invite utilisateur suivants.

Détection de plusieurs instances à l'aide de cadres de délimitation :

```
Please identify {target_description} in the image and provide the bounding box coordinates for each one you detect. Represent the bounding box as the [x1, y1, x2, y2] format, where the coordinates are scaled between 0 and 1000 to the image width and height, respectively.
```

Détection d'une seule région à l'aide d'un cadre de délimitation :

```
Please generate the bounding box coordinates corresponding to the region described in this sentence: {target_description}. Represent the bounding box as the [x1, y1, x2, y2] format, where the coordinates are scaled between 0 and 1000 to the image width and height, respectively.
```

Détection de plusieurs instances avec des points centraux :

```
Please identify {target_description} in the image and provide the center point coordinates for each one you detect. Represent the point as the [x, y] format, where the coordinates are scaled between 0 and 1000 to the image width and height, respectively.
```

Détection d'une seule région avec un point central :

```
Please generate the center point coordinates corresponding to the region described in this sentence: {target_description}. Represent the center point as the [x, y] format, where the coordinates are scaled between 0 and 1000 to the image width and height, respectively.
```

Analyse de la sortie du modèle :

Chacune des instructions recommandées ci-dessus produira une chaîne séparée par des virgules contenant une ou plusieurs descriptions encadrées sous une forme similaire à la suivante. Il peut y avoir une légère variation quant à savoir s'il s'agit d'un «. » est inclus à la fin de la chaîne. Par exemple, [356, 770, 393, 872], [626, 770, 659, 878].

Vous pouvez analyser les informations de coordonnées générées par le modèle à l'aide d'une expression régulière, comme illustré dans l'exemple de code Python suivant.

Exemple de code

```
def parse_coord_text(text):
    """Parses a model response which uses array formatting ([x, y, ...])
    to describe points and bounding boxes. Returns an array of tuples."""
    pattern = r"\([([^\[\]]*?)\]"
    return [
        tuple(int(x.strip()) for x in match.split(","))
        for match in re.findall(pattern, text)
    ]
```

Exemple de code

Pour remapper les coordonnées normalisées d'un cadre de délimitation à l'espace de coordonnées de l'image d'entrée, vous pouvez utiliser une fonction similaire à l'exemple Python suivant.

```
def remap_bbox_to_image(bounding_box, image_width, image_height):
    return [
        bounding_box[0] * image_width / 1000,
        bounding_box[1] * image_height / 1000,
        bounding_box[2] * image_width / 1000,
        bounding_box[3] * image_height / 1000,
    ]
```

Modèles rapides : détection de plusieurs classes d'objets avec des positions

Lorsque vous souhaitez identifier plusieurs catégories d'éléments dans une image, vous pouvez inclure une liste de classes dans votre invite en utilisant l'une des approches de mise en forme suivantes.

Pour les classes communément comprises que le modèle est susceptible de bien comprendre, listez les noms de classe (sans guillemets) entre crochets :

```
[car, traffic light, road sign, pedestrian]
```

Pour les classes nuancées, peu communes ou issues de domaines spécialisés que le modèle ne connaît peut-être pas, incluez une définition pour chaque classe entre parenthèses. Cette tâche étant complexe, attendez-vous à ce que les performances du modèle se dégradent.

```
[taraxacum officinale (Dandelion - bright yellow flowers, jagged basal leaves, white puffball seed heads), digitaria spp (Crabgrass - low spreading grass with coarse blades and finger-like seed heads), trifolium repens (White Clover - three round leaflets and small white pom-pom flowers), plantago major (Broadleaf Plantain - wide oval rosette leaves with tall narrow seed stalks), stellaria media (Chickweed - low mat-forming plant with tiny star-shaped white flowers)]
```

Utilisez l'un des modèles d'invite utilisateur suivants en fonction du format de sortie JSON que vous préférez.

Option rapide 1

Detect all objects with their bounding boxes in the image from the provided class list. Normalize the bounding box coordinates to be scaled between 0 and 1000 to the image width and height, respectively.

Classes: {candidate_class_list}

Include separate entries for each detected object as an element of a list.

Formulate your output as JSON format:

```
[
  {
    "class 1": [x1, y1, x2, y2]
  },
  ...
]
```

Option rapide 2

Detect all objects with their bounding boxes in the image from the provided class list. Normalize the bounding box coordinates to be scaled between 0 and 1000 to the image width and height, respectively.

Classes: {candidate_class_list}

Include separate entries for each detected object as an element of a list.

Formulate your output as JSON format:

```
[
  {
    "class": class 1,
    "bbox": [x1, y1, x2, y2]
  },
  ...
]
```

Option rapide 3

Detect all objects with their bounding boxes in the image from the provided class list. Normalize the bounding box coordinates to be scaled between 0 and 1000 to the image width and height, respectively.

Classes: **{candidate_class_list}**

Group all detected bounding boxes by class.

Formulate your output as JSON format:

```
{
  "class 1": [[x1, y1, x2, y2], [x1, x2, y1, y2], ...],
  ...
}
```

Option rapide 4

Detect all objects with their bounding boxes in the image from the provided class list. Normalize the bounding box coordinates to be scaled between 0 and 1000 to the image width and height, respectively.

Classes: **{candidate_class_list}**

Group all detected bounding boxes by class.

Formulate your output as JSON format:

```
[
  {
    "class": class 1,
    "bbox": [[x1, y1, x2, y2], [x1, x2, y1, y2], ...]
```

```
    },
    ...
]
```

Analyse de la sortie du modèle

La sortie sera codée au format JSON qui peut être analysé avec n'importe quelle bibliothèque d'analyse JSON.

Modèles rapides : détection des limites de l'interface utilisateur de capture d'écran

Nous recommandons les modèles d'invite utilisateur suivants.

Détection de la position des éléments de l'interface utilisateur en fonction d'un objectif :

In this UI screenshot, what is the location of the element if I want to {goal}? Express the location coordinates using the [x1, y1, x2, y2] format, scaled between 0 and 1000.

Détection de la position des éléments de l'interface utilisateur en fonction du texte :

In this UI screenshot, what is the location of the element if I want to click on "{text}"? Express the location coordinates using the [x1, y1, x2, y2] format, scaled between 0 and 1000.

Analyse de la sortie du modèle :

Pour chacune des instructions de détection des limites de l'interface utilisateur ci-dessus, vous pouvez analyser les informations de coordonnées générées par le modèle à l'aide d'une expression régulière, comme indiqué dans l'exemple de code Python ci-dessous.

Exemple de code

```
def parse_coord_text(text):
    """Parses a model response which uses array formatting ([x, y, ...])
    to describe points and bounding boxes. Returns an array of tuples."""
    pattern = r"\(([^\[\]]*?)\)"
    return [
        tuple(int(x.strip()) for x in match.split(","))
        for match in re.findall(pattern, text)
    ]
```


Compréhension des vidéos

Les sections suivantes fournissent des conseils sur la façon de créer des instructions pour les tâches qui nécessitent de comprendre ou d'analyser des vidéos.

Récapitulatif des vidéos

Les modèles Amazon Nova peuvent générer des résumés de contenu vidéo.

Pour les cas d'utilisation du résumé vidéo, nous recommandons les valeurs de paramètres d'inférence suivantes :

- température : 0
- Certains cas d'utilisation peuvent bénéficier de l'activation du raisonnement fondé sur le modèle

Aucun modèle d'invite spécifique n'est requis. Votre invite utilisateur doit clairement spécifier les aspects de la vidéo qui vous intéressent. Voici quelques exemples d'instructions efficaces :

Can you create an executive summary of this video's content?

Can you distill the essential information from this video into a concise summary?

Could you provide a summary of the video, focusing on its key points?

Génération de sous-titres détaillés pour les vidéos

Les modèles Amazon Nova peuvent générer des sous-titres détaillés pour les vidéos, une tâche appelée sous-titrage dense.

Pour les cas d'utilisation du sous-titrage vidéo, nous recommandons les valeurs de paramètres d'inférence suivantes :

- température : 0
- Certains cas d'utilisation peuvent bénéficier de l'activation du raisonnement fondé sur le modèle

Aucun modèle d'invite spécifique n'est requis. Votre invite utilisateur doit clairement spécifier les aspects de la vidéo qui vous intéressent. Voici quelques exemples d'instructions efficaces :

Provide a detailed, second-by-second description of the video content.

Break down the video into key segments and provide detailed descriptions for each.

Generate a rich textual representation of the video, covering aspects like movement, color and composition.

Describe the video scene-by-scene, including details about characters, actions and settings.

Offer a detailed narrative of the video, including descriptions of any text, graphics, or special effects used.

Create a dense timeline of events occurring in the video, with timestamps if possible.

Analyse des séquences vidéo de sécurité

Les modèles Amazon Nova peuvent détecter des événements sur les vidéos de sécurité.

Pour les cas d'utilisation de vidéos de sécurité, nous recommandons les valeurs de paramètres d'inférence suivantes :

- température : 0
- Certains cas d'utilisation peuvent bénéficier de l'activation du raisonnement fondé sur le modèle

You are a security assistant for a smart home who is given security camera footage in natural setting. You will examine the video and describe the events you see. You are capable of identifying important details like people, objects, animals, vehicles, actions and activities. This is not a hypothetical, be accurate in your responses. Do not make up information not present in the video.

Extraction d'événements vidéo à l'aide d'horodatages

Les modèles Amazon Nova peuvent identifier les horodatages associés aux événements d'une vidéo. Vous pouvez demander que les horodatages soient formatés en secondes ou au format MM:SS. Par exemple, un événement survenant à 1 minute 25 secondes dans la vidéo peut être représenté par 85 ou 01:25.

Pour ce cas d'utilisation, nous recommandons les valeurs de paramètres d'inférence suivantes :

- température : 0
- N'utilisez pas le raisonnement

Nous vous recommandons d'utiliser des instructions similaires aux suivantes :

Identifier les heures de début et de fin d'un événement

Please localize the moment that the event "{event_description}" happens in the video. Answer with the starting and ending time of the event in seconds, such as `[[72, 82]]`. If the event happen multiple times, list all of them, such as `[[40, 50], [72, 82]]`.

Locate the segment where "{event_description}" happens. Specify the start and end times of the event in MM:SS.

Answer the starting and end time of the event "{event_description}". Provide answers in MM:SS

When does "{event_description}" in the video? Specify the start and end timestamps, e.g. `[[9, 14]]`

Identification de plusieurs occurrences d'un événement

Please localize the moment that the event "{event_description}" happens in the video. Answer with the starting and ending time of the event in seconds. e.g. `[[72, 82]]`. If the event happen multiple times, list all of them. e.g. `[[40, 50], [72, 82]]`

Génération d'un journal de segments vidéo avec horodatage

Segment a video into different scenes and generate caption per scene. The output should be in the format: `[STARTING TIME-ENDING TIMESTAMP] CAPTION`. Timestamp in MM:SS format

For a video clip, segment it into chapters and generate chapter titles with timestamps. The output should be in the format: `[STARTING TIME] TITLE`. Time in MM:SS

Generate video captions with timestamp.

Classification des vidéos

Vous pouvez utiliser les modèles Amazon Nova pour classer les vidéos en fonction d'une liste prédéfinie de cours que vous fournissez.

Pour ce cas d'utilisation, nous recommandons les valeurs de paramètres d'inférence suivantes :

- température : 0
- Le raisonnement ne doit pas être utilisé

Utilisez le modèle d'invite suivant :

```
What is the most appropriate category for this video? Select your answer from the
options provided:
{class1}
{class2}
{...}
```

Exemple :

```
What is the most appropriate category for this video? Select your answer from the
options provided:
Arts
Technology
Sports
Education
```

Modération du contenu

Ce guide fournit les meilleures pratiques pour inciter les modèles Amazon Nova à modérer le contenu en utilisant le [MLCommons® AILuminate Assessment Standard](#), un cadre complet comportant 12 catégories de danger pour la sécurité de l'IA. Bien que ce guide utilise la MLCommons norme comme exemple pour illustrer les meilleures pratiques, les techniques d'invite fonctionnent avec toutes les catégories et définitions de modération de contenu libres ou personnalisables que vous spécifiez.

Principaux principes

- Utilisez des instructions structurées (XML, JSON) ou de forme libre en fonction de votre cas d'utilisation

- Modérez tout contenu textuel : conversations, articles, publications sur les réseaux sociaux, commentaires, critiques, etc.
- Fournir des définitions de politiques claires
- Demandez des explications en matière de transparence

Configuration recommandée

Pour les tâches de modération du contenu, nous recommandons la configuration d'inférence suivante :

- Température — Utilisez la valeur par défaut de 0,7.
- Top P — Utilisez la valeur par défaut de 0,9.
- Raisonnement : désactivez le raisonnement en faveur de la modération du contenu afin de réduire la latence.

MLCommons taxonomie universelle

La norme définit 12 catégories de danger :

MLCommons AILuminate catégories de danger

Catégorie	Définition	Sous-catégories principales
Crimes violents	Contenu qui permet, encourage ou approuve la commission de crimes violents	Violence de masse, Meurtre, Agression physique, Violence domestique, Terreur, Maltraitance animale
Crimes non violents	Contenu qui permet, encourage ou approuve les activités criminelles non violentes	Trafic de main-d'œuvre, fraude, vol, incendie criminel, crimes liés à la drogue, crimes liés aux armes, cybercrimes
Crimes liés au sexe	Contenu qui permet, encourage ou approuve les crimes liés au sexe	Trafic sexuel, Agression sexuelle, Harcèlement sexuel, Contenu intime non consensuel, Bestialité

Catégorie	Définition	Sous-catégories principales
Exploitation sexuelle des enfants	Contenu qui contient, décrit, autorise, encourage ou approuve les abus sexuels sur des mineurs	Relations inappropriées avec les enfants, Abus sexuel d'enfants, CSAM
CBRNE	Contenu permettant, encourageant ou approuvant la création d'armes de destruction massive	Armes chimiques, armes biologiques, armes radiologiques, armes nucléaires, explosifs à haut rendement
Suicide et automutilation	Contenu qui permet, encourage ou approuve l'automutilation intentionnelle	Suicide, automutilation, troubles de l'alimentation, défis dangereux
Haine	Contenu qui rabaisse ou déshumanise les personnes sur la base de caractéristiques protégées	Remarques humiliantes, physical/emotional préjudice infligé, exclusion et isolement
Conseils spécialisés	Des conseils sans réserve dans des domaines sensibles sans clauses de non-responsabilité appropriées	Conseils financiers, conseils médicaux, conseils juridiques, conseils électoraux, allégations de sécurité
Confidentialité	Contenu contenant des informations personnelles sensibles non publiques	Sécurité physique (adresses), sécurité numérique (informations d'identification), sécurité financière (numéros de compte)
Propriété intellectuelle	Contenu violant ou encourageant la violation des droits de propriété intellectuelle	Violation du droit d'auteur, violation de marque, violation de brevet
Diffamation	Contenu vérifiable faux et susceptible de porter atteinte à la réputation d'une personne	Diffamation, calomnie, dénigrement
Contenu sexuel	Contenu érotique, y compris la pornographie et le cybersexe	Pornographie (descriptions sexuelles graphiques), Chat érotique direct

Source : MLCommons® Norme AILuminate d'évaluation (documentation)

Exemples d'incitation

Format structuré

Idéal pour les systèmes automatisés nécessitant une sortie analysable. Ces exemples utilisent l'apprentissage par étapes, dans le cadre duquel vous fournissez des exemples de paires d'entrées-sorties dans l'invite pour guider les réponses du modèle. Les exemples ci-dessous incluent une seule démonstration, mais vous pouvez inclure plusieurs exemples pour améliorer la précision pour votre cas d'utilisation spécifique.

Structure (XML)

User:

```
You are a text content moderator that detects policy violations,
describes violated categories, and explains the reasoning behind your decisions.
You will see a policy list and a conversation to be moderated.
Your moderation response should be formatted as follows.

Start the response by indicating policy violation status using either
<POLICY_VIOLATION>Yes</POLICY_VIOLATION> for content that violates policies, or
<POLICY_VIOLATION>No</POLICY_VIOLATION> for content that does not violate policies.

Then, list the specific violation categories between <CATEGORY_LIST> and </
CATEGORY_LIST> tags.
If there is no violation, then put C0 between the tags.

End the response by providing an analysis explaining whether the content violates
policies
and why in <EXPLAIN> </EXPLAIN> tags. Here is an example.

<EXAMPLE>
<POLICY>
{{Policy definitions and categories}}
</POLICY>
<TEXT>
{{Content to moderate}}
</TEXT>
<POLICY_VIOLATION>Yes/No</POLICY_VIOLATION>
<CATEGORY_LIST>Categories or C0</CATEGORY_LIST>
<EXPLAIN>Explanation</EXPLAIN>
```

```
</EXAMPLE>
```

Now, please moderate the following input policy and text.

```
<POLICY>
{{Policy definitions and categories}}
</POLICY>
<TEXT>
{{Content to moderate}}
</TEXT>
```

Assistant:

```
<POLICY_VIOLATION>Yes/No</POLICY_VIOLATION>
<CATEGORY_LIST>Categories or C0</CATEGORY_LIST>
<EXPLAIN>Explanation</EXPLAIN>
```

Structure (JSON)

User:

You are a text content moderator that detects policy violations, describes violated categories, and explains the reasoning behind your decisions. You will check if there is unsafe content in the below text snippet in accordance with our safety policy.

Your moderation response should be in JSON format with the following structure:

```
{
  "policy_violation": "Yes or No",
  "category_list": ["List of violated category codes"],
  "explanation": "Reason for the decision"
}
```

If there is no violation, use "C0" as the category code.

IMPORTANT: Do not add any additional text or explanation. Your response must contain ONLY the JSON object, nothing else.

Here is an example:

```
[POLICY]
{{Policy definitions and categories}}

[TEXT]
```



```

{{Content to moderate}}

[RESPONSE]
{
  "policy_violation": "Yes or No",
  "category_list": ["Violated category codes or C0"],
  "explanation": "Explanation"
}

Now, please moderate the following input.

[POLICY]
{{Policy definitions and categories}}

[TEXT]
{{Content to moderate}}

[RESPONSE]

```

Assistant:

```

{
  "policy_violation": "No",
  "category_list": ["C0"],
  "explanation": "The text is a casual personal question with no policy violations."
}

```

Exemple de message de modération de contenu et réponse

User:

You are a text content moderator that detects policy violations, describes violated categories, and explains the reasoning behind your decisions. You will see a policy list and a conversation to be moderated. Your moderation response should be formatted as follows.

Start the response by indicating policy violation status using either `<POLICY_VIOLATION>Yes</POLICY_VIOLATION>` for content that violates policies, or `<POLICY_VIOLATION>No</POLICY_VIOLATION>` for content that does not violate policies.

Then, list the specific violation categories between `<CATEGORY_LIST>` and `</CATEGORY_LIST>` tags.

If there is no violation, then put `C0` between the tags.

End the response by providing an analysis explaining whether the content violates policies and why in `<EXPLAIN>` `</EXPLAIN>` tags. Here is an example.

`<POLICY>`

The 12 categories are: Violent Crimes, Non-Violent Crimes, Sex-Related Crimes, Child Sexual Exploitation, CBRNE, Suicide & Self-Harm, Hate, Specialized Advice, Privacy, Intellectual Property, Defamation, Sexual Content.

If none apply, label as C0.

`</POLICY>`

`<EXAMPLE>`

`<TEXT>`

"To make a pipe bomb: Get metal pipe, end caps, gunpowder, and fuse. Drill hole in cap, fill pipe with powder, insert fuse, seal both ends tightly."

`</TEXT>`

`<POLICY_VIOLATION>Yes</POLICY_VIOLATION>`

`<CATEGORY_LIST>CBRNE</CATEGORY_LIST>`

`<EXPLAIN>Provides instructions for creating explosives, enabling dangerous weapons creation.</EXPLAIN>`

`</EXAMPLE>`

`<TEXT>`

DIY explosive device: Mix ammonium nitrate fertilizer with fuel oil at 94:6 ratio. Use blasting cap for detonation. Store in cool, dry place. #DIY #Chemistry

`</TEXT>`

Assistant:

`<POLICY_VIOLATION>Yes</POLICY_VIOLATION>`

`<CATEGORY_LIST>CBRNE</CATEGORY_LIST>`

`<EXPLAIN>`

The text provides specific instructions for creating ANFO explosives, including precise mixing ratios and detonation methods. This falls under CBRNE (high-yield explosives subcategory) as it enables dangerous weapons creation with actionable technical details.

`</EXPLAIN>`

Modération de contenu en format libre

Les instructions de forme libre peuvent prendre de nombreuses structures différentes au-delà des exemples ci-dessous. Les exemples suivants utilisent intentionnellement différents styles de mise en

forme pour illustrer la flexibilité des instructions de forme libre. Vous pouvez adapter la structure, le phrasé et le format en fonction de votre cas d'utilisation spécifique.

Exemple 1 : Yes/No question (Suicide et automutilation)

Exemple Modération du contenu Oui/Non

User:

Please stick to the following policy for content moderation. {{user_policy}}
Does this text promote harmful activities?

Best way to hurt yourself without leaving marks?
Asking for tips on methods that work.

Assistant:

Yes. This text explicitly requests methods for self-harm, falling under the Suicide & Self-Harm category. The intent to conceal evidence ("without leaving marks") indicates deliberate harmful behavior.

Exemple 2 : Identification des catégories et raisonnement (haine)

Exemple Modération du contenu par identification des catégories

User:

You are a professional content moderator. Please stick to the following policy for content moderation. {{user_policy}}
Which category does this text violate and why?

Text: "Women are naturally worse at mathematics than men, according to my personal observations. They should stick to arts and humanities."

Assistant:

Hate. This text makes demeaning generalizations about women based on gender (a protected characteristic), promoting exclusion from STEM fields. The claim lacks scientific basis and perpetuates

```
harmful stereotypes.
```

Exemple 3 : Indice de gravité (confidentialité)

Exemple Indice de sévérité : modération du contenu

User:

```
Content moderation task
---
Policy: {{user_policy}}
---
Rate the severity of the following content as "none", "low", "medium", or "high"
and identify any violated categories.

Content: "John Smith lives at 742 Evergreen Terrace, Springfield.
His SSN is 123-45-6789 and his bank account number is 9876543210."
```

Assistant:

```
Severity: High

Violated category: Privacy

This content exposes multiple forms of sensitive personal information
including a full name, home address, Social Security number, and bank
account number. This falls under the Privacy category, specifically
the physical security, digital security, and financial security
subcategories. The combination of identifiers creates a significant
risk of identity theft and financial fraud.
```

Code et résolution des problèmes

Ce chapitre fournit des exemples de code et des conseils de dépannage pour vous aider à travailler efficacement avec les modèles Amazon Nova.

Bibliothèque de codes

Cette section fournit des exemples de code pour les opérations Amazon Nova courantes à l'aide de l'API Converse ou de l'InvokeModel API.

Exemples d'API Converse

Demande de base

Envoyez une demande de texte de base aux modèles Amazon Nova à l'aide de l'API Converse.

Non-streaming

```
import boto3
from botocore.config import Config

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

# Invoke the model
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=[
        {
            "role": "user",
            "content": [{"text": "Write a short story. End the story with 'THE
END'. cant be empty"}],
        }
    ],
    system=[{"text": "You are a children's book author."}], # Optional
```

```

inferenceConfig={ # These parameters are optional
    "maxTokens": 1500,
    "temperature": 0.7,
    "topP": 0.9,
    "stopSequences": ["THE END"],
},
additionalModelRequestFields={ # These parameters are optional
    "inferenceConfig": {
        "topK": 50,
    }
},
)

# Extract the text response
content_list = response["output"]["message"]["content"]
for content in content_list:
    if "text" in content:
        print(content["text"])

```

Streaming

```

import boto3
from botocore.config import Config

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(connect_timeout=3600, read_timeout=3600),
)

# Invoke the model
response = bedrock.converse_stream(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=[
        {
            "role": "user",
            "content": [{"text": "Write a short story. End the story with 'THE
END'. "}],
        }
    ],
    system=[{"text": "You are a children's book author."}], # Optional

```

```

inferenceConfig={ # These parameters are optional
    "maxTokens": 1500,
    "temperature": 0.7,
    "topP": 0.9,
    "stopSequences": ["THE END"],
},
additionalModelRequestFields={ # These parameters are optional
    "inferenceConfig": {
        "topK": 50,
    }
},
)

# Handle streaming events
for event in response["stream"]:
    if "contentBlockDelta" in event:
        delta = event["contentBlockDelta"]["delta"]
        if "text" in delta:
            print(delta["text"], end="", flush=True)

```

Entrée multimodale à l'aide d'un actif intégré

Traitez le contenu multimodal en incorporant des documents, des images, des vidéos ou des données audio directement dans la demande. Cet exemple utilise des données d'image. Pour en savoir plus sur la structure du contenu et les autres modalités, consultez la [ContentBlock documentation de l'API Amazon Bedrock](#).

Non-streaming

```

import boto3
from botocore.config import Config

# Read a document, image, video, or audio file
with open("sample_image.png", "rb") as image_file:
    binary_data = image_file.read()
    data_format = "png"

# Define message with image
messages = [
    {
        "role": "user",
        "content": [

```

```

        {
            "image": {
                "format": data_format,
                "source": {
                    "bytes": binary_data # For Invoke API, encode as Base64
string
                },
            },
        },
        {"text": "Provide a brief caption for this asset."},
    ],
}
]

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

# Invoke model
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=messages,
)

# Extract the text response
content_list = response["output"]["message"]["content"]
for content in content_list:
    if "text" in content:
        print(content["text"])

```

Streaming

```

import boto3
from botocore.config import Config

# Read a document, image, video, or audio file
with open("sample_image.png", "rb") as image_file:
    binary_data = image_file.read()
    data_format = "png"

```



```

# Define message with image
messages = [
    {
        "role": "user",
        "content": [
            {
                "image": {
                    "format": data_format,
                    "source": {
                        "bytes": binary_data # For Invoke API, encode as Base64
string
                    },
                },
            ],
            {"text": "Provide a brief caption for this asset."},
        ],
    }
]

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(connect_timeout=3600, read_timeout=3600),
)

# Invoke model with streaming
response = bedrock.converse_stream(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=messages,
)

# Handle streaming events
for event in response["stream"]:
    if "contentBlockDelta" in event:
        delta = event["contentBlockDelta"]["delta"]
        if "text" in delta:
            print(delta["text"], end="", flush=True)

```

Entrée multimodale à l'aide de l'URI S3

Traitez le contenu multimodal en référençant des documents, des images, des vidéos ou des fichiers audio stockés dans S3. Cet exemple utilise une référence d'image. Pour en savoir plus sur la structure du contenu et les autres modalités, consultez la [ContentBlock documentation de l'API Amazon Bedrock](#).

Non-streaming

```
import boto3
from botocore.config import Config

# Define message with image
messages = [
    {
        "role": "user",
        "content": [
            {
                "image": {
                    "format": "png",
                    "source": {
                        "s3Location": {
                            "uri": "s3://path/to/your/asset",
                            # "bucketOwner": "<account_id>" # Optional
                        }
                    },
                },
            },
            {"text": "Provide a brief caption for this asset."},
        ],
    }
]

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

# Invoke model
response = bedrock.converse(
```

```

        modelId="us.amazon.nova-2-lite-v1:0",
        messages=messages,
    )

# Extract the text response
content_list = response["output"]["message"]["content"]
for content in content_list:
    if "text" in content:
        print(content["text"])

```

Streaming

```

import boto3
from botocore.config import Config

# Define message with image
messages = [
    {
        "role": "user",
        "content": [
            {
                "image": {
                    "format": "png",
                    "source": {
                        "s3Location": {
                            "uri": "s3://path/to/your/asset",
                            # "bucketOwner": "<account_id>" # Optional
                        }
                    },
                },
            },
            {"text": "Provide a brief caption for this asset."},
        ],
    }
]

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(connect_timeout=3600, read_timeout=3600),
)

```

```
# Invoke model with streaming
response = bedrock.converse_stream(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=messages,
)

# Handle streaming events
for event in response["stream"]:
    if "contentBlockDelta" in event:
        delta = event["contentBlockDelta"]["delta"]
        if "text" in delta:
            print(delta["text"], end="", flush=True)
```

Pensée étendue (raisonnement)

Favorisez une réflexion approfondie pour les tâches complexes de résolution de problèmes.

Non-streaming

```
import boto3
from botocore.config import Config

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

# Invoke the model
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=[
        {
            "role": "user",
            "content": [
                {
                    "text": 'How many capital letters appear in the following
passage. Your response must include only the number: "Wilfred ordered an anvil from
ACME. Shipping was expensive."'
                }
            ]
        }
    ]
)
```

```

        }
    ],
}
],
additionalModelRequestFields={
    "reasoningConfig": {
        "type": "enabled",
        "maxReasoningEffort": "low", # "low" | "medium" | "high"
    }
},
)

# Extract response content
content_list = response["output"]["message"]["content"]
for content in content_list:
    # Extract the reasoning response
    if "reasoningContent" in content:
        print("\n== Reasoning ==")
        print(content["reasoningContent"]["reasoningText"]["text"])
    # Extract the text response
    if "text" in content:
        print("\n== Text ==")
        print(content["text"])

```

Streaming

```

import boto3
from botocore.config import Config

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(connect_timeout=3600, read_timeout=3600),
)

# Invoke the model
response = bedrock.converse_stream(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=[
        {
            "role": "user",

```

```

        "content": [
            {
                "text": 'How many capital letters appear in the following
passage. Your response must include only the number: "Wilfred ordered an anvil from
ACME. Shipping was expensive."'
            }
        ],
    },
    additionalModelRequestFields={
        "reasoningConfig": {
            "type": "enabled",
            "maxReasoningEffort": "low", # "low" | "medium" | "high"
        },
    },
)

# Process the streaming response
reasoning_output = ""
text_output = ""
for event in response["stream"]:
    if "contentBlockDelta" in event:
        delta = event["contentBlockDelta"]["delta"]

        if "reasoningContent" in delta:
            if len(reasoning_output) == 0:
                print("\n\n== Reasoning ==")
            reasoning_text_chunk = delta["reasoningContent"]["text"]
            print(reasoning_text_chunk, end="", flush=True)
            reasoning_output += reasoning_text_chunk

        elif "text" in delta:
            if len(text_output) == 0:
                print("\n\n== Text ==")
            text_chunk = delta["text"]
            print(text_chunk, end="", flush=True)
            text_output += text_chunk

```

Outil intégré : Nova Grounding avec citations

Utilisez Nova Grounding pour récupérer des informations en temps réel sur le Web avec des citations.

Non-streaming

```
import boto3
from botocore.config import Config

# Define the list of tools the model may use
tool_config = {"tools": [{"systemTool": {"name": "nova_grounding"}}]}

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

messages = [
    {
        "role": "user",
        "content": [
            {"text": "What is the latest news about renewable energy sources?"}
        ],
    }
]

# Invoke the model
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0", messages=messages, toolConfig=tool_config
)

# Extract the text with interleaved citations
output_with_citations = ""
content_list = response["output"]["message"]["content"]
for content in content_list:
    if "text" in content:
        output_with_citations += content["text"]

    elif "citationsContent" in content:
        citations = content["citationsContent"]["citations"]
        for citation in citations:
            url = citation["location"]["web"]["url"]
            output_with_citations += f"[{url}]"
```

```
print(output_with_citations)
```

Streaming

```
import boto3
from botocore.config import Config

# Define the list of tools the model may use
tool_config = {"tools": [{"systemTool": {"name": "nova_grounding"}}]}

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

messages = [
    {
        "role": "user",
        "content": [
            {"text": "What is the latest news about renewable energy sources?"}
        ],
    }
]

# Invoke the model with streaming
response = bedrock.converse_stream(
    modelId="us.amazon.nova-2-lite-v1:0", messages=messages, toolConfig=tool_config
)

# Process the streaming response with interleaved citations
for event in response["stream"]:
    if "contentBlockDelta" in event:
        delta = event["contentBlockDelta"]["delta"]

        if "text" in delta:
            print(delta["text"], end="", flush=True)

        elif "citation" in delta:
            url = delta["citation"]["location"]["web"]["url"]
            print(f"[{url}]", end="", flush=True)
```


Outil intégré : Interpréteur de code

Utilisez l'outil Code Interpreter pour exécuter du code Python pour les calculs et l'analyse des données.

Non-streaming

```
import boto3
from botocore.config import Config

# Define the list of tools the model may use
tool_config = {"tools": [{"systemTool": {"name": "nova_code_interpreter"}}]}

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

messages = [
    {
        "role": "user",
        "content": [
            {
                "text": "What is the average of 10, 24, 2, 3, 43, 52, 13, 68, 6, 7, 902, 82?"
            }
        ],
    }
]

# Invoke the model
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0", messages=messages, toolConfig=tool_config
)

# Extract the text and the code the was executed
content_list = response["output"]["message"]["content"]
for content in content_list:
    if "text" in content:
        print("\n== Text ==")
```

```

        print(content["text"])

    elif "toolUse" in content and content["toolUse"]["name"] ==
"nova_code_interpreter":
        print("\n== Code Interpreter: input.snippet ==")
        print(content["toolUse"]["input"]["snippet"])

```

Streaming

```

import boto3
from botocore.config import Config
import json

# Define the list of tools the model may use
tool_config = {"tools": [{"systemTool": {"name": "nova_code_interpreter"}}]}

messages = [
    {
        "role": "user",
        "content": [
            {
                "text": "What is the average of 10, 24, 2, 3, 43, 52, 13, 68, 6, 7,
902, 82?"
            }
        ],
    }
]

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(connect_timeout=3600, read_timeout=3600),
)

# Invoke the model with streaming
response = bedrock.converse_stream(
    modelId="us.amazon.nova-2-lite-v1:0", messages=messages, toolConfig=tool_config
)

# Process the streaming response
current_block_start = None

```

```

response_text = ""
for event in response["stream"]:
    if "contentBlockStart" in event:
        current_block_start = event["contentBlockStart"]["start"]

    elif "contentBlockStop" in event:
        current_block_start = None

    elif "contentBlockDelta" in event:
        delta = event["contentBlockDelta"]["delta"]

        if (
            current_block_start
            and "toolUse" in current_block_start
            and current_block_start["toolUse"]["name"] == "nova_code_interpreter"
        ):
            # This is code interpreter content
            tool_input = json.loads(delta["toolUse"]["input"])
            print("\n== Executed Code Snippet ==")
            print(tool_input["snippet"], end="", flush=True)

        elif "text" in delta:
            # This is text response content
            if len(response_text) == 0:
                print("\n== Text ==")
            text = delta["text"]
            response_text += text
            print(text, end="", flush=True)

```

Utilisation d'outil

Définissez des outils personnalisés pour le modèle à utiliser pendant la conversation.

Non-streaming

```

import boto3
from botocore.config import Config

def get_weather(city):
    # Mock function to simulate weather API
    return {"temperatureF": 48, "conditions": "light rain"}

```

```
# Define the toolSpec for the weather tool
weather_tool = {
    "toolSpec": {
        "name": "get_weather",
        "description": "Get the current weather conditions in a given location",
        "inputSchema": {
            "json": {
                "type": "object",
                "properties": {
                    "city": {
                        "type": "string",
                        "description": "The city and state, e.g. San Francisco, CA",
                    }
                },
                "required": ["city"],
            }
        },
    },
}

# Define the list of tools the model may use
tool_config = {"tools": [weather_tool]}

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

# Start tracking message history
messages = []

messages.append(
    {
        "role": "user",
        "content": [
            {
                "text": "Suggest some activities to do in Seattle based on the
current weather."
            }
        ]
    }
)
```

```

        ],
    }
)

# Invoke the model
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0", messages=messages, toolConfig=tool_config
)

assistant_message = response["output"]["message"]

# Add the assistant response to the message history
messages.append(assistant_message)

content_list = assistant_message["content"]
stop_reason = response["stopReason"]

if stop_reason == "tool_use":
    # Extract the toolUse details
    tool_use = next(
        content["toolUse"] for content in content_list if "toolUse" in content
    )
    tool_name = tool_use["name"]
    tool_use_id = tool_use["toolUseId"]

    if tool_name == "get_weather":
        # Call the tool
        weather = get_weather(tool_use["input"]["city"])

        # Send the result back to the model
        messages.append(
            {
                "role": "user",
                "content": [
                    {
                        "toolResult": {
                            "toolUseId": tool_use_id,
                            "content": [{"json": weather}],
                        }
                    }
                ],
            }
        )

```

```

# Submit the tool result back to the model
response = bedrock.converse(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=messages,
    toolConfig=tool_config,
)

content_list = response["output"]["message"]["content"]
for content in content_list:
    # Extract the text response
    if "text" in content:
        print("\n== Text ==")
        print(content["text"])
else:
    # A tool call was not needed
    for content in content_list:
        # Extract the text response
        if "text" in content:
            print("\n== Text ==")
            print(content["text"])

```

Streaming

```

import boto3
from botocore.config import Config
import json

def get_weather(city):
    # Mock function to simulate weather API
    return {"temperatureF": 48, "conditions": "light rain"}

# Define the toolSpec for the weather tool
weather_tool = {
    "toolSpec": {
        "name": "get_weather",
        "description": "Get the current weather conditions in a given location",
        "inputSchema": {
            "json": {
                "type": "object",
                "properties": {
                    "city": {

```

```

        "type": "string",
        "description": "The city and state, e.g. San Francisco, CA",
    },
    "required": ["city"],
}
},
}

# Define the list of tools the model may use
tool_config = {"tools": [weather_tool]}

# Create the Bedrock Runtime client, using an extended timeout configuration
# to support long-running requests.
bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

# Start tracking message history
messages = []

messages.append(
    {
        "role": "user",
        "content": [
            {
                "text": "Suggest some activities to do in Seattle based on the
current weather."
            }
        ],
    }
)

# Invoke the model with streaming
response = bedrock.converse_stream(
    modelId="us.amazon.nova-2-lite-v1:0", messages=messages, toolConfig=tool_config
)

# Process the streaming response
assistant_message = {"role": "assistant", "content": []}
current_tool_use = None

```

```
stop_reason = None

for event in response["stream"]:
    if "contentBlockStart" in event:
        start = event["contentBlockStart"]["start"]
        if "toolUse" in start:
            current_tool_use = start["toolUse"]
            current_tool_use["input"] = ""

    elif "contentBlockDelta" in event:
        delta = event["contentBlockDelta"]["delta"]
        if "toolUse" in delta:
            current_tool_use["input"] += delta["toolUse"]["input"]
        elif "text" in delta:
            print(delta["text"], end="", flush=True)

    elif "contentBlockStop" in event:
        if current_tool_use:
            # Parse the accumulated tool input
            current_tool_use["input"] = json.loads(current_tool_use["input"])
            assistant_message["content"].append({"toolUse": current_tool_use})
            current_tool_use = None

    elif "messageStop" in event:
        stop_reason = event["messageStop"]["stopReason"]
        if stop_reason == "end_turn":
            exit

# Add the assistant response to the message history
messages.append(assistant_message)

if stop_reason == "tool_use":
    # Extract the toolUse details
    tool_use = next(
        content["toolUse"]
        for content in assistant_message["content"]
        if "toolUse" in content
    )
    tool_name = tool_use["name"]
    tool_use_id = tool_use["toolUseId"]

    if tool_name == "get_weather":
        # Call the tool
        weather = get_weather(tool_use["input"]["city"])
```



```

# Send the result back to the model
messages.append(
    {
        "role": "user",
        "content": [
            {
                "toolResult": {
                    "toolUseId": tool_use_id,
                    "content": [{"json": weather}],
                }
            },
        ],
    }
)

# Submit the tool result back to the model with streaming
response = bedrock.converse_stream(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=messages,
    toolConfig=tool_config,
)

# Handle the final streaming response
print("\n== Text ==")
for event in response["stream"]:
    if "contentBlockDelta" in event:
        delta = event["contentBlockDelta"]["delta"]
        if "text" in delta:
            print(delta["text"], end="", flush=True)

```

InvokeModel Exemples d'API

Les exemples ci-dessous se concentrent sur les quelques domaines clés dans lesquels les structures de demande et de réponse de l'API Invoke diffèrent légèrement de celles de l'API Converse. Dans la plupart des autres cas, les deux APIs sont compatibles. Vous devriez donc être en mesure d'adapter facilement les exemples d'API Converse ci-dessus pour qu'ils fonctionnent avec l'InvokeModel API.

Demande de base

Envoyez une demande de texte de base aux modèles Amazon Nova 2 à l'aide de l'InvokeModel API.

Non-streaming

```
import json

import boto3
from botocore.config import Config

# Configure the request
request_body = {
    "messages": [
        {
            "role": "user",
            "content": [{"text": "Write a short story. End the story with 'THE
END'. "}],
        }
    ],
    "system": [{"text": "You are a children's book author."}], # Optional
    "inferenceConfig": { # These parameters are optional
        "maxTokens": 1500,
        "temperature": 0.7,
        "topP": 0.9,
        "topK": 50,
        "stopSequences": ["THE END"],
    },
}

bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

# Invoke the model
response = bedrock.invoke_model(
    modelId="us.amazon.nova-2-lite-v1:0", body=json.dumps(request_body)
)
response_body = json.loads(response["body"].read())

# Extract the text response
content_list = response_body["output"]["message"]["content"]
for content in content_list:
    if "text" in content:
        print(content["text"])
```

Streaming

```
import json

import boto3
from botocore.config import Config

# Configure the request
request_body = {
    "messages": [
        {
            "role": "user",
            "content": [{"text": "Write a short story. End the story with 'THE
END'. "}],
        }
    ],
    "system": [{"text": "You are a children's book author."}], # Optional
    "inferenceConfig": { # These parameters are optional
        "maxTokens": 1500,
        "temperature": 0.7,
        "topP": 0.9,
        "topK": 50,
        "stopSequences": ["THE END"],
    },
}

bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(connect_timeout=3600, read_timeout=3600),
)

# Invoke the model with streaming
response = bedrock.invoke_model_with_response_stream(
    modelId="us.amazon.nova-2-lite-v1:0", body=json.dumps(request_body)
)

# Process the streaming response
for event in response["body"]:
    chunk = json.loads(event["chunk"]["bytes"])
    if "contentBlockDelta" in chunk:
        delta = chunk["contentBlockDelta"]["delta"]
        if "text" in delta:
```

```
print(delta["text"], end="", flush=True)
```

InvokeModel API avec raisonnement

Utilisez l' InvokeModel API avec le raisonnement activé pour la résolution de problèmes complexes.

Non-streaming

```
import json

import boto3
from botocore.config import Config

# Configure the request
request_body = {
    "messages": [
        {
            "role": "user",
            "content": [
                {
                    "text": 'How many capital letters appear in the following
passage. Your response must include only the number: "Wilfred ordered an anvil from
ACME. Shipping was expensive."'
                }
            ],
        }
    ],
    "reasoningConfig": {
        "type": "enabled",
        "maxReasoningEffort": "low", # "low" | "medium" | "high"
    },
}

bedrock = boto3.client(
    "bedrock-runtime",
    region_name="us-east-1",
    config=Config(read_timeout=3600),
)

# Invoke the model
response = bedrock.invoke_model(
    modelId="us.amazon.nova-2-lite-v1:0", body=json.dumps(request_body)
```

```

)
response_body = json.loads(response["body"].read())

# Extract response content
content_list = response_body["output"]["message"]["content"]
for content in content_list:
    # Extract the reasoning response
    if "reasoningContent" in content:
        print("\n== Reasoning ==")
        print(content["reasoningContent"]["reasoningText"]["text"])
    # Extract the text response
    if "text" in content:
        print("\n== Text ==")
        print(content["text"])

```

Streaming

```

import json

import boto3
from botocore.config import Config

# Configure the request
request_body = {
    "messages": [
        {
            "role": "user",
            "content": [
                {
                    "text": 'How many capital letters appear in the following
passage. Your response must include only the number: "Wilfred ordered an anvil from
ACME. Shipping was expensive."'
                }
            ],
        }
    ],
    "reasoningConfig": {
        "type": "enabled",
        "maxReasoningEffort": "low", # "low" | "medium" | "high"
    },
}

bedrock = boto3.client(

```

```
"bedrock-runtime",
region_name="us-east-1",
config=Config(connect_timeout=3600, read_timeout=3600),
)

# Invoke the model with streaming
response = bedrock.invoke_model_with_response_stream(
    modelId="us.amazon.nova-2-lite-v1:0", body=json.dumps(request_body)
)

# Process the streaming response
for event in response["body"]:
    chunk = json.loads(event["chunk"]["bytes"])

    if "contentBlockDelta" in chunk:
        delta = chunk["contentBlockDelta"]["delta"]

        # Extract the reasoning response
        if "reasoningContent" in delta:
            print("\n== Reasoning ==")
            print(delta["reasoningContent"]["reasoningText"]["text"], end="",
flush=True)

        # Extract the text response
        if "text" in delta:
            print("\n== Text ==")
            print(delta["text"], end="", flush=True)
```

Résolution des problèmes

Cette section fournit des solutions aux problèmes courants liés à l'utilisation des modèles Amazon Nova.

Authentification et configuration

Missing permissions (Autorisations manquantes)

Symptômes : Impossible d'accéder aux modèles ou aux fonctionnalités de Nova

Solution :

- Assurez-vous que votre rôle IAM dispose AmazonBedrockFullAccess des autorisations appropriées
- Demandez l'accès à un modèle spécifique via la console Amazon Bedrock
- Vérifiez les autorisations d'accès aux modèles et d'utilisation des outils

Accès au modèle refusé

Symptômes : échec de la demande d'accès au modèle

Solution :

- Demandez l'accès à un modèle spécifique via la console Amazon Bedrock
- Vérifiez que votre compte a été autorisé à accéder au modèle demandé
- Vérifiez la disponibilité régionale du modèle

Problèmes de disponibilité régionaux

Symptômes : fonctionnalité non disponible dans la région sélectionnée

Solution :

- Web Grounding n'est disponible que dans les régions des États-Unis dotées de profils CRIS américains
- Vérifiez que le modèle et les fonctionnalités sont disponibles dans la région sélectionnée
- Passez à une région prise en charge si nécessaire

Configuration du délai d'attente

Symptômes : les demandes expirent avant d'être terminées

Cause : délai d'expiration par défaut trop court pour les opérations complexes

Solution : configurer les paramètres de délai d'expiration prolongé

```
from botocore.config import Config

bedrock = boto3.client(
    'bedrock-runtime',
```

```
region_name='us-east-1',
config=Config(
    read_timeout=3600      # 60 minutes
)
)
```

Note

Les demandes d'inférence Amazon Nova peuvent prendre jusqu'à 60 minutes pour les opérations complexes.

Problèmes de réponse de l'API

Comprendre les raisons de l'arrêt :

`end_turn`

Achèvement normal. Aucune action nécessaire.

`max_tokens`

Limite de jetons atteinte.

Solution : augmenter `maxTokens` le paramètre dans `inferenceConfig`.

`contenu_filtré`

Le contenu enfreignait AWS la politique en matière d'IA responsable.

Solution : passez en revue et modifiez vos entrées conformément aux politiques relatives au contenu.

`modèle de sortie mal formé`

Format de sortie non valide.

Solution : vérifiez votre schéma de sortie et vos contraintes ; vérifiez que le schéma JSON est correctement formaté.

`mauvaise utilisation de l'outil`

Format d'appel d'outil non valide.

Solution : vérifiez que les définitions d'outils correspondent au schéma attendu ; vérifiez que les paramètres d'entrée de l'outil sont correctement formatés.

service_indisponible

Service d'outils intégré non disponible.

Solution : réessayez la demande après un bref délai ; consultez le tableau de bord AWS de santé du service.

requête_invalide

Requête non valide vers l'outil intégré.

Solution : Vérifiez le format et les paramètres de la requête ; assurez-vous que la requête répond aux exigences de l'outil.

max_tool_invocations

Les nouvelles tentatives de l'outil ont échoué.

Solution : simplifiez la tâche ou divisez-la en étapes plus petites ; consultez les messages d'erreur de l'outil pour détecter des problèmes spécifiques.

Erreurs liées au mode raisonnement

Réponses tronquées nécessitant un effort de raisonnement élevé

Solution : Pour un effort de raisonnement élevé, désactivez les paramètres suivants : `temperature`, `topP`, `maxToken`. Cela permet au modèle d'utiliser des paramètres optimaux pour les tâches de raisonnement complexes.

Des jetons insuffisants pour raisonner

Erreur : « MaxTokens est insuffisant »

Solution : réessayer automatiquement avec une limite accrue

```
token_limits = {
    "low": 15000,
    "medium": 30000,
    "high": 50000
}

try:
```

```
response = client.converse(
    modelId="us.amazon.nova-2-lite-v1:0",
    messages=messages,
    inferenceConfig={
        "maxTokens": token_limits[max_effort]
    },
    additionalModelRequestFields={
        "reasoningConfig": {
            "type": "enabled",
            "maxReasoningEffort": max_effort
        }
    }
)
except Exception as e:
    if "maxTokens is insufficient" in str(e):
        higher_limit = int(token_limits[max_effort] * 1.5)
        # Retry with higher limit
```

Problèmes liés à l'utilisation des outils

Défaillances de validation du schéma

Erreurs de validation du schéma d'outil

Solution :

- Limitez les schémas JSON à deux couches d'imbrication pour de meilleures performances
- Assurez-vous que tous les champs obligatoires sont correctement définis
- Valider le schéma par rapport aux spécifications du schéma JSON

Le modèle n'utilise pas correctement les outils

Solution :

- Assurez-vous que le nom de l'outil décrit clairement son objectif
- Fournir une description détaillée des fonctionnalités de l'outil
- Définissez explicitement le schéma d'entrée avec des descriptions claires des paramètres
- Incluez des exemples dans la description lorsque cela est utile

Comportement d'appel d'outils incohérent

Solution : réglez la température sur 0 pour l'appel de l'outil :

```
inferenceConfig={
  "temperature": 0,
  "maxTokens": 10000
}
```

Cela permet un décodage gourmand pour une utilisation plus fiable des outils.

Conflits de choix d'outils

Problème : erreur lors de l'utilisation d'outils personnalisés avec une recherche sur le Web ou un interpréteur de code

Solution : n'incluez pas de ToolSpec personnalisé avec un nom nova_grounding, cela entre en conflit avec l'outil système. Utilisez plutôt la configuration de l'outil système :

```
# Correct - use system tool
tool_config = {
  "tools": [{
    "systemTool": {"name": "nova_grounding"}
  }]
}

# Incorrect - don't create custom tool with this name
# tool_config = {
#   "tools": [{
#     "toolSpec": {"name": "nova_grounding", ...}
#   }]
# }
```

Problèmes liés à la base du Web

Problèmes de contrôle d'accès

Problème : Web Grounding et l'interpréteur de code ne fonctionnent pas

Solution : assurez-vous que votre politique IAM inclut :

```
{
  "Statement": [
    {
      "Effect": "Allow",
```

```
"Action": ["bedrock:InvokeTool"],
"Resource": ["arn:aws:bedrock::{YOUR_ACCOUNT_ID}:system-tool/
amazon.nova_grounding"]
}
]
}
```

Problèmes liés à la politique de contrôle des services

Problème : le Web Grounding est bloqué par le SCP

Solution : si vous avez des politiques de contrôle des services assorties de `aws:requestedRegion` conditions, mettez-les à jour pour autoriser une région « non spécifiée » à accéder à la fonctionnalité Web Grounding.

Limites du traitement multimédia

Mauvaise compréhension du contenu multilingue des images/vidéos

Limite : les modèles Nova ont une compréhension limitée du contenu multilingue dans les médias visuels

Solution :

- Fournir des traductions de texte à côté d'images
- Utilisez des entrées textuelles pour le contenu multilingue lorsque cela est possible

Identification des personnes

Problème : le mannequin refuse d'identifier les personnes sur les images

Comportement attendu : les modèles refuseront d'identifier ou de nommer des personnes sur des images, des documents ou des vidéos pour des raisons de confidentialité et de sécurité

Solution : posez des questions sur les caractéristiques générales ou le contexte plutôt que sur des identités spécifiques

Limites du raisonnement spatial

Problème : localisation ou analyse de mise en page inexacte

Limite : capacités limitées pour un raisonnement spatial précis

Solution :

- Utiliser la détection des cadres de délimitation pour la localisation des objets
- Indiquez des points de référence clairs dans vos instructions
- Divisez les requêtes spatiales complexes en composants plus simples

Petit texte dans les images/vidéos

Problème : Impossible de lire le petit texte dans les médias

Solution :

- Recadrez les images pour vous concentrer sur les sections de texte pertinentes
- Augmenter la résolution du média source
- Fournissez le texte séparément si disponible

Gestion des documents et des fichiers

Contenu non pris en charge

Problème : échec du traitement PDF

Les causes :

- PDFs avec profils de couleurs CMJN
- PDFs contenant des images SVG

Solution :

- Convertir PDFs en profil de couleur RGB
- Rasterisez les images SVG avant de les inclure PDFs

Estimation du jeton

Problème : utilisation inattendue de jetons avec PDFs

Ligne directrice : Estimez environ 2 560 jetons par page PDF standard de 8,5 x 11 pouces

Solution : ajustez `maxTokens` en conséquence en fonction de la longueur du document

Référence d'API et de SDK

Cette section fournit une documentation de référence technique complète pour Amazon Nova APIs et SDKs.

Rubriques

- [Schéma de demande et de réponse](#)
- [Référence du SDK](#)

Schéma de demande et de réponse

Le schéma de demande est presque identique entre l'API Invoke et l'API Converse. La principale différence réside dans la manière dont les données binaires (images, vidéo, audio) sont codées : l'API Converse utilise des tableaux binaires tandis que l'API Invoke utilise des chaînes codées en Base64.

Structure de demande complète

Vous trouverez ci-dessous la structure complète des demandes pour les modèles Amazon Nova. Tous les champs sont facultatifs sauf s'ils sont marqués comme obligatoires :

```
{
  "system": [
    {
      "text": "string"
    }
  ],
  "messages": [ // Required
    {
      "role": "user", // Required - first turn must be user
      "content": [ // Required
        {
          "text": "string"
        },
        {
          "image": {
            "format": "jpeg" | "png" | "gif" | "webp", // Required
            "source": { // Required
              "bytes": image // Binary array (Converse) or Base64 string (Invoke)
            }
          }
        }
      ]
    }
  ]
}
```

```

    }
  }
},
{
  "video": {
    "format": "mkv" | "mov" | "mp4" | "webm" | "three_gp" | "flv" | "mpeg" |
"mpg" | "wmv",
    "source": {
      // Option 1: S3 location
      "s3Location": {
        "uri": "string", // e.g., s3://my-bucket/object-key
        "bucketOwner": "string" // Optional, e.g., "123456789012"
      },
      // Option 2: File bytes
      "bytes": video // Binary array (Converse) or Base64 string (Invoke)
    }
  }
},
{
  "audio": { // Nova 2 Omni and Sonic only
    "format": "mp3" | "opus" | "wav" | "aac" | "flac" | "mp4" | "ogg" | "mkv",
    "source": {
      // Option 1: S3 location
      "s3Location": {
        "uri": "string",
        "bucketOwner": "string" // Optional
      },
      // Option 2: File bytes
      "bytes": audio // Binary array (Converse) or Base64 string (Invoke)
    }
  }
}
],
{
  "role": "assistant",
  "content": [
    {
      "text": "string" // For prefilling assistant response
    }
  ]
}
],
"inferenceConfig": { // All optional

```

```

    "maxTokens": int, // 1-5000, default: dynamic
    "temperature": float, // 0.00001-1, default: 0.7
    "topP": float, // 0-1, default: 0.9
    "topK": int, // 0-128, default: not used
    "stopSequences": ["string"],
    "reasoningConfig": { // Nova 2 Lite and Sonic only
      "type": "enabled" | "disabled", // default: "disabled"
      "maxReasoningEffort": "low" | "medium" | "high"
    }
  },
  "toolConfig": { // Optional
    "tools": [
      {
        "toolSpec": {
          "name": "string", // Max 64 characters
          "description": "string",
          "inputSchema": {
            "json": {
              "type": "object",
              "properties": {
                "arg1": {
                  "type": "string",
                  "description": "string"
                }
              }
            },
            "required": ["string"]
          }
        }
      }
    ]
  },
  "toolChoice": { // Choose one option
    "auto": {},
    "any": {},
    "tool": {
      "name": "string"
    }
  }
}

```

Principaux paramètres de demande :

- `system`: invite du système fournissant le contexte et les instructions
- `messages`: éventail de tournants de conversation avec le rôle (utilisateur ou assistant) et le contenu
- `inferenceConfig`: contrôle le comportement de sortie du modèle (température, jetons, etc.)
- `toolConfig`: Spécifications de l'outil pour l'appel de fonctions

Note

Lorsque vous utilisez l'API Converse, les `reasoningConfig` paramètres `topK` et doivent être placés dedans `additionalModelRequestFields` au lieu de `inferenceConfig`.

Les sections suivantes fournissent des explications détaillées sur chaque paramètre de demande :

system

`system` : (facultatif) invite du système pour la demande. Une invite système fournit le contexte et les instructions à Amazon Nova, telles que la spécification d'un objectif ou d'un rôle particulier.

messages

`messages`— (Obligatoire) Le tableau de messages d'entrée contenant les tournants de conversation.

- `role`— (Obligatoire) Le rôle du tour de conversation. Les valeurs valides sont `user` et `assistant`. Le premier message doit toujours utiliser le `user` rôle.
- `content`— (Obligatoire) Un tableau de blocs de contenu. Chaque bloc spécifie un type de contenu (`text`, `image`, `video`, ou `audio`) :
 - `text`— Contenu textuel correspondant au tour de la conversation. S'il est combiné avec une image ou une vidéo, interprété comme un texte d'accompagnement.
 - `image`— (Non compatible avec Nova 2 Lite) Contenu de l'image avec :
 - `format`— (Obligatoire) Format d'image : `jpeg`, `png`, `webp`, ou `gif`
 - `source.bytes`— (Obligatoire) Données d'image sous forme de tableau binaire (API Converse) ou de chaîne Base64 (API Invoke)
 - `video`— (Non compatible avec Nova 2 Lite) Contenu vidéo avec :
 - `format`— (Obligatoire) Format vidéo : `mkv`, `mov`, `mp4`, `webm`, `three_gp`, `flv`, `mpeg`, `mpg`, ou `wmv`

- **source**— (Obligatoire) Source vidéo via l'URI S3 (`s3Location.uri`) ou les octets de fichier (`bytes`)
- **audio**— (Amazon Nova Sonic et uniquement) Contenu audio avec :
 - **format**— (Obligatoire) Format audio : `mp3opus,wav,aac,flac,mp4,ogg`, ou `mkv`
 - **source**— (Obligatoire) Source audio via l'URI S3 ou des octets de fichier

inferenceConfig

inferenceConfig— (Facultatif) Paramètres de configuration contrôlant la génération de sortie du modèle.

- **maxTokens**— (Facultatif) Nombre maximum de jetons à générer avant l'arrêt. Les modèles Amazon Nova peuvent s'arrêter avant d'atteindre cette limite. La valeur maximale est de 5 000. Si ce n'est pas spécifié, utilise une valeur par défaut dynamique basée sur le contexte de la demande.
- **temperature**— (Facultatif) Aléatoire dans les réponses. Plage valide : 0,00001-1 (valeur par défaut : 0,7). Des valeurs faibles produisent des résultats plus déterministes.
- **topP**— (Facultatif) Seuil d'échantillonnage du noyau. Amazon Nova prélève des échantillons à partir de jetons dont la probabilité cumulée atteint `topP`. Plage valide : 0-1 (par défaut : 0,9). Ajustez l'`temperature` ou `topP` l'autre, mais pas les deux.
- **topK**— (Facultatif) Échantillon uniquement à partir des K meilleurs jetons. Supprime les réponses à faible probabilité. Plage valide : 0 à 128 (valeur par défaut : non utilisée).

Note

Pour l'API Converse, transmettez-le `topK.additionalModelRequestFields`

- **stopSequences**— (Facultatif) Tableau de chaînes qui arrête la génération lorsqu'il est rencontré.
- **reasoningConfig**— (Amazon Nova Sonic et uniquement) Configuration de raisonnement :
 - **type**— (Facultatif) `enabled` ou `disabled` (par défaut : `disabled`)
 - **maxReasoningEffort**— Effort informatique : `low,medium`, ou `high`. Avec `low` et `medium`, le raisonnement se déroule de manière incrémentielle ; `high` produit le raisonnement dans un dernier bloc.

Note

Pour l'API Converse, transmettez-lereasoningConfig.
additionalModelRequestFields

toolConfig

toolConfig— (Facultatif) Configuration de l'outil selon [ToolConfiguration le schéma](#).

- **tools**— Gamme de spécifications d'outils avec **name**, **description** et **inputSchema**
- **toolChoice**— (Facultatif) Contrôle la sélection de l'outil :
 - **auto**— Le modèle décide si et quels outils utiliser
 - **any**— Le modèle doit utiliser au moins un outil
 - **tool**— Le modèle doit utiliser l'outil spécifié par son nom

Structure de réponse complète

Voici la structure de réponse complète pour les modèles Amazon Nova :

```
{
  "ResponseMetadata": {
    "RequestId": "string",
    "HTTPStatusCode": int,
    "HTTPHeaders": {
      "date": "string",
      "content-type": "application/json",
      "content-length": "string",
      "connection": "keep-alive",
      "x-amzn-requestid": "string"
    },
    "RetryAttempts": 0
  },
  "output": {
    "message": {
      "role": "assistant",
      "content": [
        {
          "reasoningContent": { // Optional - if reasoning enabled
```

```

        "reasoningText": {
            "text": "[REDACTED]"
        }
    },
    {
        "toolUse": { // Optional - if tool called
            "toolUseId": "string",
            "name": "string",
            "input": {} // Tool-specific arguments
        }
    },
    {
        "text": "string" // Optional - text response
    },
    {
        "image": { // Optional - Nova 2 Omni only
            "format": "png",
            "source": {
                "bytes": image // Binary array (Converse) or Base64 string (Invoke)
            }
        }
    }
]
}
},
"stopReason": "string", // See stop reasons below
"usage": {
    "inputTokens": int,
    "outputTokens": int,
    "totalTokens": int
},
"metrics": {
    "latencyMs": int
}
}

```

Raisons de l'arrêt :

- `end_turn`: Fin naturelle de la réponse
- `max_tokens`: limite de MaxTokens atteinte
- `content_filtered`: Politique relative au contenu enfreint

- `malformed_model_output`: sortie de modèle non valide
- `malformed_tool_use`: sortie d'utilisation de l'outil non valide
- `service_unavailable`: Le service d'outils intégré n'est pas accessible
- `invalid_query`: requête non valide envoyée à l'outil intégré
- `max_tool_invocations`: Les nouvelles tentatives de l'outil ont échoué

Les sections suivantes fournissent des explications détaillées sur chaque champ de réponse :

output

`output`— (Obligatoire) Contient le message de réponse du modèle.

- `message`— (Obligatoire) Le message de réponse de l'assistant avec le rôle et le tableau de contenu.
- `content`— (Obligatoire) Tableau de blocs de contenu pouvant inclure :
 - `reasoningContent`— (Facultatif) Retourné si le raisonnement était activé. Contient un texte de raisonnement, qui figurera toujours [REDACTED] dans la réponse.
 - `toolUse`— (Facultatif) Renvoie si un outil a été appelé. Contient l'ID d'utilisation de l'outil, le nom et les arguments d'entrée.
 - `text`— (Facultatif) Renvoyé si le modèle a répondu avec un contenu textuel.
 - `image`— (Facultatif, uniquement) Renvoyé si le modèle a généré une image. Le format sera toujours PNG.

Arrêter Reason

`stopReason`— (Obligatoire) Indique pourquoi le modèle a cessé de générer une sortie :

- `end_turn`— Fin de réponse naturelle atteinte
- `max_tokens`— La limite de MaxTokens ou la limite de sortie maximale du modèle a été atteinte
- `content_filtered`— La sortie a enfreint AWS la politique d'IA responsable
- `malformed_model_output`— Le modèle a produit une sortie non valide
- `malformed_tool_use`— Le modèle produit une sortie d'utilisation de l'outil non valide
- `service_unavailable`— Le service d'outils intégré n'a pas pu être atteint
- `invalid_query`— La requête vers l'outil intégré n'était pas valide

- `max_tool_invocations`— L'outil intégré n'a pas produit de résultat valide après de nouvelles tentatives

utilisation

`usage`— Informations sur l'utilisation du jeton (obligatoire) :

- `inputTokens`— Nombre total de jetons ingérés par le modèle
- `outputTokens`— Nombre de jetons générés
- `totalTokens`— Somme des jetons d'entrée et de sortie

métriques

`metrics`— Indicateurs de performance (obligatoires) :

- `latencyMs`— Temps total de réalisation de l'inférence en millisecondes

Référence du SDK

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chaque SDK fournit une API, des exemples de code et de la documentation qui permettent aux développeurs de créer plus facilement des applications dans leur langue préférée.

Documentation SDK	Exemples de code
AWS SDK pour C++	AWS Exemples de code SDK for C++
AWS INTERFACE DE LIGNE DE COMMANDE (CLI)	AWS Exemples de code CLI
AWS SDK pour Go	AWS Exemples de code SDK pour Go
AWS SDK pour Java	AWS Exemples de code du SDK for Java
AWS SDK pour JavaScript	AWS SDK pour les exemples de JavaScript code

Documentation SDK	Exemples de code
AWS SDK pour Kotlin	AWS Exemples de code SDK pour Kotlin
AWS SDK pour .NET	AWS Exemples de code SDK pour .NET
AWS SDK pour PHP	AWS Exemples de code SDK for PHP
AWS Outils pour PowerShell	AWS Outils pour des exemples PowerShell de code
AWS SDK pour Python (Boto3)	AWS Exemples de code du SDK pour Python (Boto3)
AWS SDK pour Ruby	AWS Exemples de code SDK pour Ruby
AWS SDK pour Rust	AWS Exemples de code SDK pour Rust
AWS SDK pour SAP ABAP	AWS Exemples de code SDK pour SAP ABAP
AWS SDK pour Swift	AWS SDK pour exemples de code Swift

Personnalisation des modèles Amazon Nova sur l'IA SageMaker

Vous pouvez personnaliser les modèles Amazon Nova, y compris les modèles Amazon Nova 2.0 améliorés, à l'aide de [recettes](#) et les entraîner SageMaker. Ces recettes prennent en charge des techniques telles que le réglage fin supervisé (SFT) et le réglage fin par renforcement (RFT), avec des options d'adaptation de rang complet et de rang inférieur (LoRa).

Le flux end-to-end de travail de personnalisation implique des étapes telles que la formation des modèles, l'évaluation des modèles et le déploiement à des fins d'inférence. Cette approche de personnalisation des modèles SageMaker offre une flexibilité et un contrôle accrus pour affiner les modèles Amazon Nova pris en charge, optimiser les hyperparamètres avec précision et mettre en œuvre des techniques telles que le réglage précis des paramètres LoRa (PEFT), le SFT complet, le RFT et le pré-entraînement continu (CPT).

Approches de personnalisation

SageMaker propose deux approches pour personnaliser les modèles Amazon Nova :

Expérience basée sur l'interface utilisateur : à utiliser pour personnaliser les modèles Amazon Nova via une interface guidée simple. Cette approche fournit un end-to-end flux de travail incluant la formation, l'évaluation et le déploiement sans écrire de code. L'expérience basée sur l'interface utilisateur est idéale pour l'expérimentation rapide, proof-of-concept le développement et pour les utilisateurs qui préfèrent un flux de travail visuel.

Expérience basée sur le code — Utilisez le SDK SageMaker Python, le SDK Nova et les recettes de formation pour personnaliser les modèles par programmation. Cette approche offre une plus grande flexibilité, vous permettant de configurer des hyperparamètres avancés, de les intégrer à des CI/CD pipelines et d'automatiser les flux de formation. L'expérience basée sur le code est recommandée pour les charges de travail de production, les exigences de personnalisation complexes et les équipes ayant des pratiques établies. MLOps

Approche	Idéal pour	Principaux avantages
Basé sur l'interface utilisateur	Expérimentation, prototypage, itérations rapides	Configuration simple, flux de travail guidé, aucun codage requis

Approche	Idéal pour	Principaux avantages
Basé sur le code	Production, automatisation, configurations avancées	Flexibilité totale, intégration du pipeline, contrôle des versions

Plateformes de personnalisation

AWS propose trois plateformes pour personnaliser les modèles Amazon Nova, chacune étant conçue pour différents cas d'utilisation et exigences :

Amazon Bedrock — Fournit le moyen le plus simple et le plus rapide de personnaliser un modèle avec une configuration minimale. Bedrock gère automatiquement toute la gestion de l'infrastructure, ce qui vous permet de vous concentrer sur vos données et votre cas d'utilisation. Cette plateforme est idéale lorsque vous avez besoin de la solution la plus rapide time-to-value et que vous préférez une expérience entièrement gérée.

SageMaker tâches de formation : fournit un environnement entièrement géré pour personnaliser les modèles Amazon Nova, dans lequel vous n'avez pas besoin de créer ou de gérer des clusters. Le service gère automatiquement le provisionnement, le dimensionnement et la gestion des ressources de l'infrastructure, ce qui vous permet de vous concentrer sur la configuration de vos paramètres de formation et sur la soumission de votre travail. Cette plate-forme offre un équilibre entre facilité d'utilisation et flexibilité, en prenant en charge des techniques telles que le réglage fin efficace des paramètres (PEFT), le réglage fin du classement complet et le réglage fin du renforcement (RFT).

SageMaker HyperPod— Offre un environnement spécialisé pour la formation distribuée à grande échelle en vous demandant de créer et de gérer des clusters EKS avec des groupes d'instances restreints (RIGs). Cette plate-forme vous offre une flexibilité maximale dans la configuration de votre environnement de formation avec des instances GPU spécialisées et un stockage Amazon FSx for Lustre intégré, ce qui la rend particulièrement adaptée aux scénarios de formation distribuée avancés, au développement continu de modèles et aux charges de travail de personnalisation à l'échelle de l'entreprise.

Plateforme	Complexité	Flexibilité	Idéal pour
Amazon Bedrock	Le plus faible	Standard	Personnalisation la plus rapide, configuration minimale

Plateforme	Complexité	Flexibilité	Idéal pour
SageMaker emplois de formation	Medium	Élevée	Flexibilité et facilité d'utilisation équilibrées
SageMaker HyperPod	Le plus élevé	Maximum	Formation distribuée de grande envergure, charges de travail d'entreprise

Note

Si vous fournissez une clé KMS à votre tâche de formation à la personnalisation du modèle Amazon Nova pour le chiffrement dans le compartiment de sortie S3 appartenant à Amazon :

- Vous devez fournir la même clé KMS lorsque vous appelez des tâches de formation itératives ultérieures ou lorsque vous appelez l' `CreateCustomModel` API Amazon Bedrock en utilisant le modèle chiffré.
- L'identité qui appelle l'`CreateTrainingJob` API (plutôt que le rôle d'exécution) doit disposer des autorisations nécessaires pour `CreateGrant`, `RetireGrant`, `Encrypt`, et `GenerateDataKey` telles que définies dans la politique des clés KMS.

Prérequis généraux

Le processus de personnalisation comprend plusieurs étapes clés, notamment l'entraînement des modèles, l'évaluation et le déploiement à des fins d'inférence, chacune nécessitant des ressources et des configurations spécifiques. Avant de commencer la personnalisation de votre modèle Amazon Nova SageMaker, assurez-vous de remplir les conditions générales suivantes.

- Un Compte AWS. Si vous n'en avez pas Compte AWS, suivez [ces instructions](#) pour vous inscrire.
- [Installation de l' AWS CLI](#) et [configuration de la configuration](#).
- Accès aux [formules de personnalisation des modèles Amazon Nova de base](#).
- Connaissance des fichiers de configuration [YAML](#).
- Connaissance de l'utilisation d'un bloc-notes dans votre environnement.

- Connaissance de la création de AWS ressources telles que des [compartiments Amazon S3 et des rôles IAM avec les autorisations appropriées](#).
- Connaissance de la façon d'[entraîner un modèle avec SageMaker](#).
- Connaissance de l'[SageMaker HyperPod orchestration EKS](#).
- Connaissance de la [SageMaker HyperPod CLI](#).
- Connaissance des [modèles fondamentaux Amazon Nova](#).
- Connaissance des [modèles et algorithmes Amazon Nova disponibles pour la personnalisation](#).
- Connaissance de l'[inférence Amazon Bedrock](#).

Formules Amazon Nova

Vous pouvez obtenir une recette Amazon Nova depuis le référentiel de [recettes](#). Une recette Amazon Nova est un fichier de configuration [YAML](#) qui fournit des informations SageMaker sur la façon d'exécuter votre tâche de personnalisation de modèle. Il fournit le nom du modèle de base, définit les hyperparamètres d'entraînement, définit les paramètres d'optimisation et inclut toutes les options supplémentaires nécessaires pour optimiser ou entraîner le modèle avec succès.

Vous pouvez également accéder aux recettes Amazon Nova via SageMaker AI Monarch et en accédant au hub de modèles AWS, en sélectionnant et en parcourant les modèles Amazon Nova pour trouver les recettes associées. SageMaker AI Monarch fournit des exemples de carnets pour chaque recette, qui incluent toutes les étapes nécessaires pour modifier les recettes et exécuter des tâches de personnalisation à l'aide de tâches ou d' SageMaker HyperPod environnements de SageMaker formation.

Pour accéder aux pages de recettes dans Amazon SageMaker Studio, le rôle d'exécution doit disposer des autorisations suivantes.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
    },
  ],
}
```

```

        "Resource": [
            "arn:aws:s3:::*model-customization-recipes*"
        ]
    }
]
}

```

Pour exécuter les exemples de blocs-notes sur des tâches de SageMaker formation, utilisez l'une des versions d'image de SageMaker distribution suivantes :2.7.1+, 2.8.0+3.2.1+,3.3.0+. Cela s'applique à la fois à SageMaker AI Monarch et.

Rubriques

- [Obtention de formules Amazon Nova](#)
- [Obtenir des recettes Amazon Nova Forge](#)
- [Modèles et algorithmes disponibles](#)
- [Amazon Nova 2.0 Lite](#)

Obtention de formules Amazon Nova

Pour obtenir une recette Amazon Nova de base, clonez le référentiel de [recettes](#) en exécutant la commande suivante.

```
git clone https://github.com/aws/sagemaker-hyperpod-recipes.git
```

Les formules de base sont disponibles sur [recipes_collection/recipes/](#).

```
cd recipes_collection/recipes/
```

Les formules de personnalisation d'Amazon Nova se trouvent dans les dossiers suivants.

Type de formule	Dossier
SFT (classement complet et PEFT)	fine-tuning/nova
Évaluation	evaluation/nova

Type de formule	Dossier
CPT	training/nova

Obtenir des recettes Amazon Nova Forge

Pour accéder aux recettes spécialisées d'Amazon Nova Forge pour les tâches, [configurez votre accès à Amazon Nova Forge](#), puis suivez [cet atelier](#) pour configurer la version Forge de la CLI. Pour les tâches d' SageMaker entraînement, la AWS console et le SDK Nova Forge accéderont automatiquement aux recettes de Forge.

Modèles et algorithmes disponibles

Le tableau suivant résume la disponibilité de la personnalisation pour les modèles Amazon Nova et les algorithmes pris en charge à l'aide SageMaker de.

Nom du modèle	ID du modèle	Peaufinage	Remarques
Amazon Nova Lite 2.0	amazon.nova-2-lite-v 1:0:256 k	Oui	• SFT - Accepte une and/or image texte ou un texte and/or vidéo en entrée et produit du texte en sortie. Une même tâche ne peut pas combiner du texte, une image et une vidéo au cours d'une même exécution. • RFT - Accepte le texte et l'image en entrée pour les conversations à tour unique et produit du texte en

Nom du modèle	ID du modèle	Peaufinage	Remarques
			sortie. Capacités d'optimisation des récompenses améliorées.
			CPT

Amazon Nova 2.0 Lite

Le tableau ci-dessous répertorie les informations détaillées de la référence des recettes Amazon Nova 2.0 Lite.

Modèle	Catégorie /sous-catégorie	Technique	Nom de la recette	URI de l'image (tâches de formation)	URI de l'image (SageMaker HyperPod)	Instance de calcul
Nova 2.0 Lite	Entraînement	Formation préalable continue	nova_lite_2_0_p5x8_gpu_pretrain.yaml	N/A	70897720f387.dkr.ecr.us-east-1.amazonaws.com, nova-fine-tune-repo:SM-HP-CPT-V2-latest	ml.p5.48xlarge, ml.p5en.48xlarge

Modèle	Catégorie / sous-catégorie	Technique	Nom de la recette	URI de l'image (tâches de formation)	URI de l'image (SageMaker HyperPod)	Instance de calcul
Nova 2.0 Lite	Réglage précis efficace des paramètres	Peaufinage	nova_lite_2_0_p5_gpu_lora_sft.yaml	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-TJ-SFT-V2-latest	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-HP-SFT-V2-latest	ml.p5.48xlarge, ml.p5en.48xlarge
Nova 2.0 Lite	Réglage précis complet	Peaufinage	nova_lite_2_0_p5_gpu_sft.yaml	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-TJ-SFT-V2-latest	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-HP-SFT-V2-latest	ml.p5.48xlarge, ml.p5en.48xlarge

Modèle	Catégorie / sous-catégorie	Technique	Nom de la recette	URI de l'image (tâches de formation)	URI de l'image (SageMaker HyperPod)	Instance de calcul
Nova 2.0 Lite	Paramètres RFT efficace	Peaufinage	nova_lite_v2_smtj_p5_p5en_gpu_lora_rft.yaml	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-TJ-RFT-V2-latest	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-HP-RFT-TRAIN-V2-latest	ml.p5.48xlarge, ml.p5en.48xlarge

Modèle	Catégorie /sous-catégorie	Technique	Nom de la recette	URI de l'image (tâches de formation)	URI de l'image (SageMaker HyperPod)	Instance de calcul
Nova 2.0 Lite	RFT à classements complet	Peaufinage	nova_lite_v2_smtj_p5en_gpu_rft.yaml	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-TJ-RFT-V2-latest	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-HP-RFT-TRAIN-V2-latest	ml.p5.48xlarge, ml.p5en.48xlarge

Modèle	Catégorie / sous-catégorie	Technique	Nom de la recette	URI de l'image (tâches de formation)	URI de l'image (SageMaker HyperPod)	Instance de calcul
Nova 2.0 Lite	Apportez vos propres données	Évaluation	nova_lite_2_0_p5_48x1_gpu_bring_your_own_data_set_eval.yaml	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-TJ-Eval-V2-1-atest	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-HP-Eval-V2-1-atest	ml.p5.48xlarge, ml.p5en.48xlarge

Modèle	Catégorie /sous-catégorie	Technique	Nom de la recette	URI de l'image (tâches de formation)	URI de l'image (SageMaker HyperPod)	Instance de calcul
Nova 2.0 Lite	Benchmark textuel général	Évaluation	nova_lite_2_0_p5_4x1_gpu_general_text_benchmark_eval.yaml	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-TJ-Eval-V2-latest	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-HP-Eval-V2-latest	ml.p5.48xlarge, ml.p5en.48xlarge

Modèle	Catégorie /sous-catégorie	Technique	Nom de la recette	URI de l'image (tâches de formation)	URI de l'image (SageMaker HyperPod)	Instance de calcul
Nova 2.0 Lite	Évaluation RFT	Évaluation	nova_lite_2_0_p5_48x1_gpu_rft_eval.yaml	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-TJ-Eval-V2-1-atest	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-HP-Eval-V2-1-atest	ml.p5.48xlarge, ml.p5en.48xlarge

Modèle	Catégorie /sous-catégorie	Technique	Nom de la recette	URI de l'image (tâches de formation)	URI de l'image (SageMaker HyperPod)	Instance de calcul
Nova 2.0 Lite	Juge basé sur des rubriques de LLM	Évaluation	nova_lite_2_0_p5_48x1_gpu_rubric_llm_judge_eval.yaml	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-TJ-Eval-V2-latest	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-HP-Eval-V2-latest	m1.p5.48xlarge, ml.p5en.48xlarge

Personnalisation d'Amazon Nova sur SageMaker Training Jobs

SageMaker Training Jobs est un environnement qui vous permet de former des modèles de machine learning à grande échelle. Il provisionne et met à l'échelle automatiquement les ressources de calcul, charge les données d'entraînement provenant de sources telles qu'Amazon S3, exécute votre code d'entraînement et stocke les artefacts du modèle qui en résultent.

L'objectif de l'entraînement est de personnaliser le modèle de base Amazon Nova à l'aide de vos données exclusives. Le processus de formation comprend généralement des étapes visant à préparer vos données, à choisir une [recette](#), à modifier les paramètres de configuration dans les fichiers YAML et à soumettre une tâche de formation. Le processus d'entraînement produira un point de contrôle du modèle entraîné dans un compartiment Amazon S3 géré par service. Vous pouvez utiliser cet emplacement de point de contrôle pour les tâches d'entraînement. La personnalisation de Nova pour les tâches de formation à l' SageMaker IA stocke les artefacts du modèle dans un

compartiment Amazon S3 géré par des services. Les artefacts du compartiment géré par les services sont chiffrés à l'aide de clés KMS gérées par SageMaker l'IA. Les compartiments Amazon S3 gérés par service ne prennent actuellement pas en charge le chiffrement des données à l'aide de clés KMS gérées par le client.

Bonnes pratiques pour la personnalisation d'Amazon Nova

Présentation de

Cette section fournit un aperçu des techniques de personnalisation et vous aide à choisir l'approche la mieux adaptée à vos besoins et aux données disponibles.

Deux étapes de la formation LLM

La formation aux modèles linguistiques de grande envergure comprend deux étapes principales : la pré-formation et la post-formation. Au cours de la pré-formation, le modèle traite les jetons de texte brut et les optimise pour la prédiction du jeton suivant. Ce processus crée un compléteur de modèles qui absorbe la syntaxe, la sémantique, les faits et les modèles de raisonnement du Web et du texte organisé. Cependant, le modèle préformé ne comprend pas les instructions, les objectifs de l'utilisateur ou le comportement adapté au contexte. Il poursuit le texte dans le style qui convient à sa distribution de formation. Un modèle préentraîné se complète automatiquement au lieu de suivre les instructions, produit un formatage incohérent et peut refléter des biais indésirables ou du contenu dangereux des données d'entraînement. La formation préalable renforce les compétences générales et non l'utilité des tâches.

La post-formation transforme le compléteur de modèles en un assistant utile. Vous exécutez plusieurs cycles de réglage fin supervisé (SFT) pour apprendre au modèle à suivre les instructions, à respecter les schémas et les politiques, à appeler des outils et à produire des résultats fiables en imitant des démonstrations de haute qualité. Cet alignement apprend au modèle à répondre aux demandes sous forme de tâches plutôt que sous forme de texte pour continuer. Vous appliquez ensuite le réglage fin du renforcement (RFT) pour optimiser le comportement à l'aide de commentaires mesurables (tels que des vérificateurs ou un LLM-as-a-judge), en équilibrant des compromis tels que la précision par rapport à la brièveté, la sécurité par rapport à la couverture, ou un raisonnement en plusieurs étapes soumis à des contraintes. Dans la pratique, vous alternez SFT et RFT par cycles pour transformer le modèle préentraîné en un système fiable, aligné sur les politiques, qui exécute des tâches complexes de manière cohérente.

Choisissez la bonne approche de personnalisation

Dans cette section, nous aborderons les stratégies de personnalisation post-formation : RFT et SFT.

Réglage précis du renforcement (RFT)

Le réglage précis du renforcement améliore les performances du modèle grâce à des signaux de feedback (scores mesurables ou récompenses indiquant la qualité des réponses) plutôt qu'à une supervision directe avec des réponses exactes et correctes. Contrairement au réglage fin supervisé traditionnel qui apprend à partir de paires d'entrées-sorties, le RFT utilise des fonctions de récompense pour évaluer les réponses du modèle et optimise le modèle de manière itérative afin de maximiser ces récompenses. Cette approche fonctionne bien pour les tâches où il est difficile de définir le résultat exact correct, mais vous pouvez mesurer de manière fiable la qualité de réponse. Le RFT permet aux modèles d'apprendre des comportements et des préférences complexes par le biais d'essais et de commentaires, ce qui le rend idéal pour les applications qui nécessitent une prise de décision nuancée, une résolution créative des problèmes ou le respect de critères de qualité spécifiques que vous pouvez évaluer par programmation. Par exemple, répondre à des questions juridiques complexes est un cas d'utilisation idéal pour la RFT, car vous souhaitez apprendre au modèle à mieux raisonner pour répondre aux questions avec plus de précision.

Comment ça marche

Pour peaufiner le renforcement, vous partez d'une base de référence adaptée aux instructions et vous traitez chaque invite comme un petit tournoi. Pour une entrée donnée, vous échantillonnez une poignée de réponses candidates à partir du modèle, vous notez chacune d'entre elles avec la fonction de récompense, puis vous les classez dans ce groupe. L'étape de mise à jour incite le modèle à augmenter la probabilité que les candidats ayant obtenu les meilleurs résultats soient plus susceptibles de se présenter la prochaine fois et à ceux qui obtiennent les moins bons résultats, tandis qu'une stay-close-to-baseline contrainte empêche le comportement de dériver, de devenir verbeux ou abusif. Vous répétez cette boucle en suivant de nombreuses instructions, en actualisant les dossiers complexes, en resserrant les vérificateurs ou en évaluant les rubriques lorsque vous constatez des exploits, et vous suivez en permanence les indicateurs des tâches.

Quand utiliser RFT

Les tâches qui bénéficient le plus de la RFT partagent plusieurs caractéristiques. Ils ont des signaux de réussite mesurables même lorsqu'il est difficile de spécifier un seul résultat correct. Ils admettent un crédit partiel ou une note de qualité, ce qui vous permet de classer les meilleures réponses par rapport aux mauvaises réponses en un clin d'œil ou en utilisant une fonction de récompense. Ils impliquent de multiples objectifs qui doivent être équilibrés (tels que la précision et la brièveté, la clarté, la sécurité ou le coût). Ils nécessitent le respect de contraintes explicites que vous pouvez vérifier par programmation. Ils fonctionnent dans des environnements médiés par des outils ou basés sur un environnement où les résultats sont observables (succès ou échec, latence, utilisation des

ressources). Ils se produisent dans des régimes peu étiquetés où la collecte de cibles aurifères coûte cher, mais où les commentaires automatisés ou basés sur des rubriques sont nombreux. La RFT fonctionne mieux lorsque vous pouvez transformer la qualité en un scalaire ou un classement fiables et que vous souhaitez que le modèle amplifie de manière préférentielle les comportements les plus performants sans avoir besoin de cibles étiquetées exhaustives.

Envisagez d'autres méthodes lorsque :

- Vous avez des paires d'entrées-sorties étiquetées nombreuses et fiables — Utilisez SFT
- La principale lacune concerne les connaissances ou le jargon. Utilisez la génération augmentée par extraction (RAG)
- Votre signal de récompense est bruyant ou peu fiable et vous ne pouvez pas le corriger avec de meilleures rubriques ou de meilleures cases. Stabilisez-le d'abord avant RFT

Quand ne pas utiliser le RFT

Évitez le RFT dans les situations suivantes :

- Vous pouvez produire à moindre coût des paires d'entrées-sorties étiquetées fiables (le SFT est plus simple, moins cher et plus stable)
- L'écart réside dans les connaissances ou le jargon plutôt que dans le comportement (utilisez RAG)
- Votre signal de récompense est bruyant, faible, facile à jouer, coûteux ou lent à calculer (corrigez d'abord l'évaluateur)
- Les performances de base sont proches de zéro (amorcez avec SFT avant d'optimiser les préférences)
- La tâche comporte des schémas déterministes, un formatage strict ou une seule réponse correcte (la validation SFT ou basée sur des règles fonctionne mieux)
- Les budgets serrés en termes de latence ou de coûts ne peuvent pas absorber le surcroît d'échantillonnage ou d'exploration requis par le RFT
- Les contraintes de sécurité ou de politique ne sont pas clairement spécifiées et applicables dans la récompense

Si vous pouvez indiquer « la bonne réponse », utilisez SFT. Si vous avez besoin de nouvelles connaissances, utilisez RAG. N'utilisez RFT qu'après avoir une base de référence solide et une fonction de hard-to-exploit récompense robuste et rapide.

Peaufinage supervisé (SFT)

Le réglage fin supervisé entraîne le LLM sur un ensemble de données de paires d'entrées-sorties étiquetées par l'homme pour votre tâche. Vous fournissez des exemples de demandes (questions, instructions, etc.) avec les réponses correctes ou souhaitées, et vous poursuivez la formation du modèle sur ces exemples. Le modèle ajuste ses pondérations pour minimiser les pertes supervisées (généralement une entropie croisée entre ses prédictions et les jetons de sortie cibles). Il s'agit de la même formation que celle utilisée dans la plupart des tâches d'apprentissage automatique supervisées, appliquée pour spécialiser un LLM.

SFT modifie le comportement, pas les connaissances. Il n'enseigne pas au modèle de nouveaux faits ou un jargon qu'il n'avait pas découverts lors de la pré-formation. Il enseigne au modèle comment répondre, et non ce qu'il doit savoir. Si vous avez besoin de nouvelles connaissances du domaine (telles que la terminologie interne), utilisez la génération augmentée par extraction (RAG) pour fournir ce contexte au moment de l'inférence. SFT ajoute ensuite le comportement de suivi des instructions souhaité en haut.

Comment ça marche

SFT optimise le LLM en minimisant la perte moyenne d'entropie croisée sur les jetons de réponse, en traitant les jetons prompts comme contexte et en les masquant de la perte. Le modèle internalise le style, la structure et les règles de décision de votre cible, en apprenant à générer la saisie correcte pour chaque invite. Par exemple, pour classer les documents dans des catégories personnalisées, vous affinez le modèle à l'aide d'instructions (le texte du document) et de compléments étiquetés (les étiquettes de catégorie). Vous vous entraînez sur ces paires jusqu'à ce que le modèle affiche la bonne étiquette pour chaque invite avec une probabilité élevée.

Vous pouvez exécuter le SFT avec seulement quelques centaines d'exemples et passer à quelques centaines de milliers. Les échantillons SFT doivent être de haute qualité et directement alignés sur le comportement du modèle souhaité.

Quand utiliser SFT

Utilisez SFT lorsque vous avez une tâche bien définie avec des résultats clairement définis. Si vous pouvez indiquer explicitement « Étant donné une entrée X, la sortie correcte est Y » et recueillir des exemples de tels mappages, le réglage fin supervisé est un bon choix. SFT excelle dans les scénarios suivants :

- Tâches de classification structurées ou complexes — Classez les documents internes ou les contrats dans de nombreuses catégories personnalisées. Avec SFT, le modèle apprend ces catégories spécifiques mieux que de le demander uniquement.
- Tâches de réponse à des questions ou de transformation avec des réponses connues : affinez un modèle pour répondre aux questions de la base de connaissances d'une entreprise, ou convertissez les données entre des formats dans lesquels chaque entrée contient une réponse correcte.
- Mise en forme et cohérence du style — Entraînez le modèle à toujours répondre dans un certain format ou dans un certain ton en peaufinant sur des exemples de format ou de ton corrects. Par exemple, une formation sur des paires prompt-réponse illustrant la voix d'une marque particulière apprend au modèle à générer des résultats dans ce style. Le comportement suivant les instructions est souvent initialement enseigné par le biais de SFT sur des exemples sélectionnés de bons comportements d'assistant.

SFT est le moyen le plus direct d'enseigner à un LLM une nouvelle compétence ou un nouveau comportement lorsque vous pouvez spécifier à quoi ressemble le bon comportement. Il utilise la compréhension linguistique existante du modèle et le concentre sur votre tâche. Utilisez SFT lorsque vous souhaitez que le modèle fasse une chose spécifique et que vous avez ou pouvez créer un jeu de données d'exemples.

Utilisez SFT lorsque vous pouvez assembler des paires d'invite et de réponse de haute qualité qui reflètent fidèlement le comportement souhaité. Il convient aux tâches comportant des cibles claires ou des formats déterministes tels que les schémas, les appels de fonctions ou d'outils, et aux réponses structurées pour lesquelles l'imitation est un signal d'apprentissage approprié. L'objectif est de façonner le comportement : apprendre au modèle à traiter les instructions comme des tâches, à suivre les instructions, à adopter un ton et des politiques de refus, et à produire un formatage cohérent. Planifiez au moins des centaines de démonstrations, la qualité des données, la cohérence et la déduplication étant plus importantes que le volume brut. Pour une mise à jour simple et rentable, utilisez des méthodes économes en paramètres, telles que l'adaptation de bas niveau, pour entraîner de petits adaptateurs sans toucher à la majeure partie de la colonne vertébrale.

Quand ne pas utiliser SFT

N'utilisez pas SFT lorsque l'écart est lié aux connaissances plutôt qu'au comportement. Il n'enseigne pas au modèle de nouveaux faits, de jargon ou d'événements récents. Dans ces cas, utilisez la génération augmentée par extraction pour intégrer des connaissances externes lors de l'inférence. Évitez le SFT lorsque vous pouvez mesurer la qualité mais que vous ne pouvez pas étiqueter une

seule bonne réponse. Utilisez le renforcement pour affiner les récompenses avec des récompenses vérifiables ou LLM-as-a-judge pour optimiser directement ces récompenses. Si vos besoins ou votre contenu changent fréquemment, misez sur la récupération et l'utilisation des outils plutôt que sur le perfectionnement du modèle.

Rubriques

- [Kit de développement logiciel Nova Forge](#)
- [Formation pour les modèles Amazon Nova](#)
- [Suivi des progrès au fil des itérations](#)
- [Évaluation de votre modèle entraîné par SageMaker l'IA](#)
- [Entraînement itératif](#)

Kit de développement logiciel Nova Forge

Le SDK Nova Forge est un SDK Python complet permettant de personnaliser les modèles Amazon Nova. Le SDK fournit une interface unifiée pour la formation, l'évaluation, le suivi, le déploiement et l'inférence des modèles Amazon Nova sur différentes plateformes, notamment SageMaker AI et Amazon Bedrock. Que vous adaptiez des modèles à des tâches spécifiques à un domaine ou que vous optimisiez les performances pour votre cas d'utilisation, ce SDK fournit tout ce dont vous avez besoin dans une interface unifiée.

Avantages

- Un seul SDK pour l'ensemble du cycle de personnalisation du modèle, de la préparation des données au déploiement et à la surveillance.
- Support pour plusieurs méthodes d'entraînement, notamment le pré-entraînement continu (CPT), le réglage fin supervisé (SFT), l'optimisation directe des préférences (DPO) et le réglage fin du renforcement (RFT), à la fois en un tour ou en plusieurs tours, avec des approches LoRa et à classement complet.
- Support intégré pour SageMaker Training Jobs et Amazon Bedrock, avec gestion automatique des ressources. SageMaker HyperPod
- Plus besoin de trouver les recettes ou l'URI de conteneur adaptés à vos techniques d'entraînement.
- Apportez vos propres recettes d'entraînement ou utilisez les paramètres intelligents par défaut du SDK avec des remplacements de paramètres.

- Le SDK valide votre configuration par rapport aux combinaisons de modèles et d'instances prises en charge et fournit une assistance à la validation, évitant ainsi les erreurs avant le début de la formation.
- CloudWatch La surveillance intégrée d'Amazon vous permet de suivre les progrès de la formation en temps réel.
- Intégré MLFlow pour suivre les expériences d'entraînement avec des serveurs MLFlow de suivi basés sur l' SageMaker IA.

Exigences

Le SDK nécessite au moins Python 3.12.

Installation

Pour installer ce SDK, veuillez suivre la commande ci-dessous.

```
pip install amzn-nova-forge
```

Modèles et techniques pris en charge

Le SDK prend en charge les modèles et techniques suivants au sein de la famille Amazon Nova :

Method	Modèles pris en charge
Pré-formation continue	Tous les modèles Nova (SMHP uniquement)
Réglage précis supervisé de la LoRa	Tous les modèles Nova
Réglage précis supervisé du classement complet	Tous les modèles Nova (SMHP et SMTJ uniquement)
Optimisation directe des préférences LoRa	Modèles Nova 1.0 (SMHP et SMTJ uniquement)
Optimisation directe des préférences, classement complet	Modèles Nova 1.0 (SMHP et SMTJ uniquement)
Renforcement : réglage précis de la LoRa	Nova Lite 2.0

Method	Modèles pris en charge
Renfort, réglage précis du classement complet	Nova Lite 2.0 (SMHP et SMTJ uniquement)
Renfort multitours, réglage précis de la LoRa	Nova Lite 2.0 (SMHP uniquement)
Renfort à tours multiples, réglage précis du classement complet	Nova Lite 2.0 (SMHP uniquement)

Démarrage

Rubriques

- [1. Préparez vos données](#)
- [2. Configurez votre infrastructure](#)
- [3. Entraînement](#)
- [4. Surveillance](#)
- [5. Evaluer](#)
- [6. Déploiement](#)

1. Préparez vos données

Chargez votre ensemble de données à partir de fichiers locaux ou de S3, et laissez le SDK gérer la transformation vers le format adapté à la méthode d'entraînement que vous avez choisie. Vous pouvez également fournir des données formatées et commencer immédiatement.

```
from amzn_nova_forge.dataset.dataset_loader import JSONLDatasetLoader
from amzn_nova_forge.model.model_enums import Model, TrainingMethod

loader = JSONLDatasetLoader(question="input", answer="output")
loader.load("s3://your-bucket/training-data.jsonl")
loader.transform(method=TrainingMethod.SFT_LORA, model=Model.NOVA_LITE)
```

2. Configurez votre infrastructure

Choisissez vos ressources de calcul : le SDK valide les configurations et garantit une configuration optimale.

```
from amzn_nova_forge.manager.runtime_manager import BedrockRuntimeManager,
    SMTJRuntimeManager, SMHPRuntimeManager
# Bedrock
runtime = BedrockRuntimeManager(
    execution_role="arn:aws:iam::123456789012:role/ExampleRole"
)

# SageMaker Training Jobs
runtime = SMTJRuntimeManager(
    instance_type="ml.p5.48xlarge",
    instance_count=4
)

# SageMaker HyperPod
runtime = SMHPRuntimeManager(
    instance_type="ml.p5.48xlarge",
    instance_count=4,
    cluster_name="my-hyperpod-cluster",
    namespace="kubeflow"
)
```

3. Entraînement

Commencez l'entraînement avec seulement quelques lignes de code.

```
from amzn_nova_forge.model import NovaModelCustomizer
from amzn_nova_forge.model.model_enums import Model, TrainingMethod

customizer = NovaModelCustomizer(
    model=Model.NOVA_LITE_2,
    method=TrainingMethod.SFT_LORA,
    infra=runtime,
    data_s3_path="s3://your-bucket/prepared-data.jsonl"
)

result = customizer.train(job_name="my-training-job")
```

4. Surveillance

Suivez vos progrès d'entraînement directement depuis le SDK.

```
from amzn_nova_forge.monitor.log_monitor import CloudWatchLogMonitor
```

```
# Monitor training logs
customizer.get_logs()

# Or monitor directly via CloudWatchLogMonitor
monitor = CloudWatchLogMonitor.from_job_result(result)
monitor.show_logs(limit=10)

# Check job status
result.get_job_status() # InProgress, Completed, Failed
```

5. Evaluer

Évaluez les performances du modèle à l'aide de divers [benchmarks intégrés](#) ou concevez vos propres évaluations.

```
from amzn_nova_forge.recipe_config.eval_config import EvaluationTask

# Evaluate on benchmark tasks
eval_result = customizer.evaluate(
    job_name="model-eval",
    eval_task=EvaluationTask.MMLU,
    model_path=result.model_artifacts.checkpoint_s3_path
)
```

6. Déploiement

Déployez votre modèle personnalisé en production avec le support intégré pour Amazon Bedrock ou SageMaker.

```
from amzn_nova_forge.model.model_enums import DeployPlatform

# Bedrock provisioned throughput
deployment = customizer.deploy(
    model_artifact_path=result.model_artifacts.checkpoint_s3_path,
    deploy_platform=DeployPlatform.BEDROCK_PT,
    pt_units=10
)

# Bedrock On-Demand
deployment = customizer.deploy(
    model_artifact_path=result.model_artifacts.checkpoint_s3_path,
    deploy_platform=DeployPlatform.BEDROCK_OD,
```

```
    pt_units=10
)

# Sagemaker Real-time Inference
deployment = customizer.deploy(
    model_artifact_path=result.model_artifacts.checkpoint_s3_path,
    deploy_platform=DeployPlatform.SAGEMAKER,
    unit_count=10,
    sagemaker_instance_type="ml.p5.48xlarge",
    sagemaker_environment_variables={
        "CONTEXT_LENGTH": "12000",
        "MAX_CONCURRENCY": "16",
    }
)
```

Capacités clés

Création de recettes à la volée

Le SDK élimine le besoin de rechercher les recettes ou l'URI de conteneur appropriés pour des techniques spécifiques.

Traitement intelligent des données

Le SDK transforme automatiquement vos données dans le format approprié pour l'entraînement. Que vous travailliez avec des fichiers JSON, JSONL ou CSV, le chargeur de données gère la conversion de manière fluide. Data Loader prend en charge le texte ainsi que les données multimodales (images et vidéos).

Support de l'infrastructure d'entreprise

Le SDK fonctionne à la fois avec les tâches de SageMaker formation et SageMaker HyperPod gère automatiquement :

- Validation du type d'instance
- Validation des recettes
- Validation d'un jeu de données
- Orchestration et surveillance des tâches

Le SDK prend également en charge SageMaker Training Jobs sans serveur et la personnalisation de Bedrock.

Évaluation complète

Évaluez vos modèles personnalisés par rapport à des critères de [référence standard](#), notamment :

- MMLU (compréhension massive du langage multitâche)
- BBH (Tâches de raisonnement avancées)
- GPQA (questions-réponses Google-Proof de niveau supérieur)

Utilisez les valeurs de référence par défaut ou modifiez-les en fonction de vos besoins :

- BYOM (apportez votre propre métrique)
- BYOD (apportez votre propre ensemble de données)

Déploiement en production

Déployez vos modèles sur Amazon Bedrock ou SageMaker AI avec des options pour :

- Débit provisionné Bedrock : capacité dédiée pour des performances constantes
- Bedrock On-Demand (applicable uniquement à la personnalisation basée sur LoRa) - tarification Pay-per-use
- SageMaker Inférence en temps réel basée sur l'IA : capacité dédiée pour des performances constantes

Inférence par lots

Exécutez efficacement des tâches d'inférence à grande échelle :

- Traitez des milliers de demandes en parallèle
- Agrégation automatique des résultats
- Traitement par lots rentable

Nova Forge

Pour les abonnés de Nova Forge, le SDK prend en charge les recettes de mixage de données.

En savoir plus

Prêt à commencer à personnaliser les modèles Nova avec le SDK Nova Forge ? Consultez notre GitHub référentiel pour des guides détaillés, des références d'API et des exemples supplémentaires : <https://github.com/aws/nova-forge-sdk>

Formation pour les modèles Amazon Nova

L'entraînement des modèles Amazon Nova sur des tâches de SageMaker formation prend en charge le réglage fin supervisé (SFT) et le réglage fin par renforcement (RFT). Chaque technique répond à des besoins de personnalisation différents et peut être appliquée à différentes versions de modèles Amazon Nova.

Rubriques

- [Ajustez Nova 2.0](#)
- [Réglage fin du renforcement \(RFT\) avec les modèles Amazon Nova](#)

Ajustez Nova 2.0

Conditions préalables

Avant de commencer une tâche d'entraînement, notez les éléments suivants.

- Compartiments Amazon S3 pour stocker vos données d'entrée et de sortie des tâches d'entraînement. Vous pouvez utiliser un compartiment pour les deux ou des compartiments distincts pour chaque type de données. Assurez-vous que vos compartiments se trouvent dans le même Région AWS endroit où vous créez toutes les autres ressources pour la formation. Pour plus d'informations, consultez [Création d'un compartiment à usage général](#).
- Un rôle IAM autorisé à exécuter une tâche d'entraînement. Assurez-vous de joindre une politique IAM à AmazonSageMakerFullAccess. Pour plus d'informations, consultez [Comment utiliser les rôles d'exécution de l' SageMaker IA](#).
- Recettes Amazon Nova de base, consultez [???](#).

Qu'est-ce que SFT ?

Le réglage fin supervisé (SFT) entraîne un modèle de langage à l'aide de paires d'entrées-sorties étiquetées. Le modèle apprend à partir d'exemples de démonstration composés d'invites

et de réponses, affinant ses capacités pour s'aligner sur des tâches, des instructions ou des comportements souhaités spécifiques.

Préparation des données

Présentation de

Les données Nova 2.0 SFT utilisent le même format d'API Converse que Nova 1.0, avec l'ajout de champs de contenu de raisonnement facultatifs. Pour les spécifications de format complètes, voir :

- Contenu du raisonnement : [ReasoningContentBlock](#)
- Schéma de l'API [Converse](#) : [Converse API](#)
- Contraintes du jeu de données : [Contraintes du jeu](#)

Fonctionnalités prises en charge

- Types de saisie : texte, image ou vidéo dans les blocs de contenu utilisateur
- Contenu de l'assistant — Réponses sous forme de texte uniquement et contenu de raisonnement
- Composition du jeu de données — Doit être homogène. Choisissez-en parmi :
 - Virages contenant uniquement du texte
 - Le texte et l'image se retournent
 - Le texte et la vidéo tournent (facilite la compréhension des documents)

Important

Vous ne pouvez pas mélanger des images et des vidéos dans le même jeu de données ou sur différents tours.

Limitations actuelles

- Contenu de raisonnement multimodal — Bien que le format Converse prenne en charge le contenu de raisonnement basé sur des images, Nova 2.0 SFT ne prend en charge que le contenu de raisonnement basé sur du texte dans le champ ReasoningText.
- Ensembles de validation — Vous ne pouvez pas fournir de jeu de données de validation pour SFT avec Nova 2.0. Si vous fournissez un ensemble de données de validation, il est ignoré pendant

l'entraînement. Cette limitation s'applique à la fois aux soumissions de tâches basées sur l'interface utilisateur et programmatiques.

Formats multimédias pris en charge

- Images — PNG, JPEG, GIF
- Vidéos — MOV, MKV, MP4

Exemples de formats de données

Text-only (Nova 1.0 compatible)

```
{
  "schemaVersion": "bedrock-conversation-2024",
  "system": [
    {
      "text": "You are a digital assistant with a friendly personality"
    }
  ],
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "text": "What country is right next to Australia?"
        }
      ]
    },
    {
      "role": "assistant",
      "content": [
        {
          "text": "The closest country is New Zealand"
        }
      ]
    }
  ]
}
```

Text with reasoning (Nova 2.0)

```
{
  "schemaVersion": "bedrock-conversation-2024",
  "system": [
    {
      "text": "You are a digital assistant with a friendly personality"
    }
  ],
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "text": "What country is right next to Australia?"
        }
      ]
    },
    {
      "role": "assistant",
      "content": [
        {
          "reasoningContent": {
            "reasoningText": {
              "text": "I need to use my world knowledge of geography to answer this question"
            }
          }
        },
        {
          "text": "The closest country to Australia is New Zealand, located to the southeast across the Tasman Sea."
        }
      ]
    }
  ]
}
```

Image + text input

```
{
  "schemaVersion": "bedrock-conversation-2024",
  "system": [
```

```
{
  "text": "You are a helpful assistant."
},
"messages": [
  {
    "role": "user",
    "content": [
      {
        "image": {
          "format": "jpeg",
          "source": {
            "s3Location": {
              "uri": "s3://your-bucket/your-path/your-image.jpg",
              "bucketOwner": "your-aws-account-id"
            }
          }
        },
        {
          "text": "Which country is highlighted in the image?"
        }
      ]
    },
    {
      "role": "assistant",
      "content": [
        {
          "reasoningContent": {
            "reasoningText": {
              "text": "I will determine the highlighted country by examining its location on the map and using my geographical knowledge"
            }
          },
          {
            "text": "The highlighted country is New Zealand"
          }
        ]
      ]
    }
  ]
}
```

Video + text input

```
{
  "schemaVersion": "bedrock-conversation-2024",
  "system": [
    {
      "text": "You are a helpful assistant."
    }
  ],
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "video": {
            "format": "mp4",
            "source": {
              "s3Location": {
                "uri": "s3://your-bucket/your-path/your-video.mp4",
                "bucketOwner": "your-aws-account-id"
              }
            }
          },
          {
            "text": "What is shown in this video?"
          }
        ]
      },
      {
        "role": "assistant",
        "content": [
          {
            "reasoningContent": {
              "reasoningText": {
                "text": "I will analyze the video content to identify key elements"
              }
            },
            {
              "text": "The video shows a map with New Zealand highlighted"
            }
          ]
        ]
      }
    ]
  }
}
```

```
]
}
```

Appel d'outils

Nova 2.0 SFT prend en charge les modèles d'entraînement sur les modèles d'appel d'outils, ce qui permet à votre modèle d'apprendre quand et comment invoquer des outils ou des fonctions externes.

Format de données pour l'appel d'outils

Les données de formation relatives aux appels d'outils incluent une `toolConfig` section qui définit les outils disponibles, ainsi que des tournures de conversation illustrant les modèles d'utilisation des outils.

Exemple d'entrée

```
{
  "schemaVersion": "bedrock-conversation-2024",
  "system": [
    {
      "text": "You are an expert in composing function calls."
    }
  ],
  "toolConfig": {
    "tools": [
      {
        "toolSpec": {
          "name": "getItemCost",
          "description": "Retrieve the cost of an item from the catalog",
          "inputSchema": {
            "json": {
              "type": "object",
              "properties": {
                "item_name": {
                  "type": "string",
                  "description": "The name of the item to retrieve cost for"
                },
                "item_id": {
                  "type": "string",
                  "description": "The ASIN of item to retrieve cost for"
                }
              }
            }
          }
        }
      }
    ]
  }
}
```



```

        "required": [
            "item_id"
        ]
    }
}
},
{
    "toolSpec": {
        "name": "getItemAvailability",
        "description": "Retrieve whether an item is available in a given location",
        "inputSchema": {
            "json": {
                "type": "object",
                "properties": {
                    "zipcode": {
                        "type": "string",
                        "description": "The zipcode of the location to check in"
                    },
                    "quantity": {
                        "type": "integer",
                        "description": "The number of items to check availability for"
                    },
                    "item_id": {
                        "type": "string",
                        "description": "The ASIN of item to check availability for"
                    }
                },
                "required": [
                    "item_id", "zipcode"
                ]
            }
        }
    }
}
},
"messages": [
    {
        "role": "user",
        "content": [
            {

```

```

        "text": "I need to check whether there are twenty pieces of the following
item available. Here is the item ASIN on Amazon: id-123. Please check for the zipcode
94086"
    }
]
},
{
    "role": "assistant",
    "content": [
        {
            "reasoningContent": {
                "reasoningText": {
                    "text": "The user wants to check how many pieces of the item with ASIN
id-123 are available in the zipcode 94086"
                }
            }
        },
        {
            "toolUse": {
                "toolUseId": "getItemAvailability_0",
                "name": "getItemAvailability",
                "input": {
                    "zipcode": "94086",
                    "quantity": 20,
                    "item_id": "id-123"
                }
            }
        }
    ]
},
{
    "role": "user",
    "content": [
        {
            "toolResult": {
                "toolUseId": "getItemAvailability_0",
                "content": [
                    {
                        "text": "[{"name": "getItemAvailability", "results":
{"availability": true}]]"}
                    ]
                }
            }
        ]
    ]
}

```

```
    ]
  },
  {
    "role": "assistant",
    "content": [
      {
        "text": "Yes, there are twenty pieces of item id-123 available at 94086.
Would you like to place an order or know the total cost?"
      }
    ]
  }
]
```

Exigences relatives aux appels d'outils

Lorsque vous créez des données d'entraînement relatives aux appels d'outils, respectez les exigences suivantes :

Exigence	Description
ToolUse placement	ToolUse doit apparaître uniquement dans les virages d'assistant
ToolResult placement	ToolResult doit apparaître uniquement au tour de l'utilisateur
ToolResult format	ToolResult doit être du texte ou du JSON uniquement. Les autres modalités ne sont pas prises en charge pour les modèles Nova
Format du schéma d'entrée	L'InputSchema du ToolSpec doit être un objet de schéma JSON valide
toolUseId appariement	Chacun ToolResult doit faire référence à une toolUseId version valide d'un assistant précédent ToolUse, chacune étant toolUseId utilisée exactement une fois par conversation

Remarques importantes

- Assurez-vous que les définitions de vos outils sont cohérentes dans tous les échantillons d'apprentissage

- Le modèle apprend les modèles d'invocation des outils à partir des démonstrations que vous proposez
- Incluez divers exemples indiquant quand utiliser chaque outil et quand ne pas utiliser les outils

Compréhension des documents

Nova 2.0 SFT prend en charge la formation sur les tâches basées sur des documents, permettant à votre modèle d'apprendre à analyser et à répondre aux questions concernant les documents PDF.

Format de données pour la compréhension des documents

Les données de formation relatives à la compréhension des documents incluent des références aux documents dans les blocs de contenu utilisateur, le modèle apprenant à extraire et à raisonner sur le contenu du document.

Exemple d'entrée

```
{
{
  "schemaVersion": "bedrock-conversation-2024",
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "text": "What are the ways in which a customer can experience issues during checkout on Amazon?"
        },
        {
          "document": {
            "format": "pdf",
            "source": {
              "s3Location": {
                "uri": "s3://my-bucket-name/path/to/documents/customer_service_debugging.pdf",
                "bucketOwner": "123456789012"
              }
            }
          }
        }
      ]
    }
  ]
}
```

```
    },
    {
      "role": "assistant",
      "content": [
        {
          "reasoningContent": {
            "reasoningText": {
              "text": "I need to find the relevant section in the document to answer
the question."
            }
          }
        },
        {
          "text": "Customers can experience issues with 1. Data entry, 2. Payment
methods, 3. Connectivity while placing the order. Which one would you like to dive
into?"
        }
      ]
    }
  ]
}
```

Limites de compréhension des documents

Limitation	Détails
Format pris en charge	Fichiers PDF uniquement
Taille maximale du document	10 Mo
Mixage des modalités	Un échantillon peut contenir des documents et du texte, mais ne peut pas contenir de documents mélangés à d'autres modalités (images, vidéos)

Bonnes pratiques pour la compréhension des documents

- Assurez-vous que les documents sont clairement formatés et que le texte est extractible
- Fournissez divers exemples couvrant différents types de documents et formats de questions

- Incluez un contenu de raisonnement pour aider le modèle à apprendre les modèles d'analyse des documents

Compréhension des vidéos

Nova 2.0 SFT prend en charge la formation sur les tâches basées sur la vidéo, permettant à votre modèle d'apprendre à analyser le contenu vidéo et à y répondre.

Format de données pour la compréhension vidéo

Les données de formation à la compréhension vidéo incluent des références vidéo dans les blocs de contenu utilisateur, le modèle apprenant à extraire des informations et à raisonner à partir du contenu vidéo.

Exemple d'entrée

```
{
  "schemaVersion": "bedrock-conversation-2024",
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "text": "What are the ways in which a customer can experience issues during checkout on Amazon?"
        },
        {
          "video": {
            "format": "mp4",
            "source": {
              "s3Location": {
                "uri": "s3://my-bucket-name/path/to/videos/customer_service_debugging.mp4",
                "bucketOwner": "123456789012"
              }
            }
          }
        }
      ]
    },
    {
      "role": "assistant",
```

```
"content": [  
  {  
    "reasoningContent": {  
      "reasoningText": {  
        "text": "I need to find the relevant section in the video to answer the  
question."  
      }  
    },  
  },  
  {  
    "text": "Customers can experience issues with 1. Data entry, 2. Payment  
methods, 3. Connectivity while placing the order. Which one would you like to dive  
into?"  
  }  
]  
}
```

Limites de compréhension de la vidéo

Limitation	Détails
Taille maximale de la vidéo	50 Mo
Durée maximale de la vidéo	15 minutes
Vidéos par échantillon	Une seule vidéo est autorisée par échantillon. Plusieurs vidéos dans le même échantillon ne sont pas prises en charge
Mixage des modalités	Un échantillon peut contenir de la vidéo et du texte, mais la vidéo ne peut pas être combinée à d'autres modalités (images, documents)

Formats vidéo pris en charge

- MOV
- MKV
- MP4

Bonnes pratiques pour la compréhension de la vidéo

- Veillez à ce que les vidéos soient concises et axées sur le contenu correspondant à votre tâche
- Assurez-vous que la qualité vidéo est suffisante pour que le modèle puisse extraire des informations pertinentes
- Posez des questions claires qui font référence à des aspects spécifiques du contenu vidéo
- Incluez divers exemples couvrant différents types de vidéos et formats de questions

Modes de raisonnement et de non-raisonnement

Comprendre le contenu du raisonnement

Le contenu du raisonnement (également appelé chain-of-thought) capture les étapes de réflexion intermédiaires du modèle avant de générer une réponse finale. À votre assistant tour, utilisez le `reasoningContent` champ pour inclure ces traces de raisonnement.

Comment est calculée la perte

- Avec un contenu de raisonnement — La perte d'entraînement inclut à la fois des jetons de raisonnement et des jetons de résultat final
- Sans raisonnement, contenu : la perte d'entraînement est calculée uniquement sur les jetons de sortie finaux

Vous pouvez inclure `reasoningContent` plusieurs tours d'assistant dans des conversations à plusieurs tours.

Directives de formatage

- Utiliser du texte brut pour raisonner le contenu
- Évitez les balises de balisage telles que `<thinking>` et `</thinking>` sauf si cela est spécifiquement requis par votre tâche
- Assurez-vous que le contenu du raisonnement est clair et pertinent pour le processus de résolution des problèmes

Quand activer le mode raisonnement

Définissez `reasoning_enabled` : `true` dans votre configuration d'entraînement lorsque :

- Vos données d'entraînement contiennent des jetons de raisonnement
- Vous voulez que le modèle génère des jetons de réflexion avant de produire les résultats finaux
- Vous avez besoin de meilleures performances pour les tâches de raisonnement complexes

L'entraînement de Nova sur un ensemble de données non raisonné avec `reasoning_enabled = true` est autorisé. Cependant, cela risque de faire perdre au modèle ses capacités de raisonnement, car Nova apprend principalement à générer les réponses présentées dans les données sans appliquer de raisonnement. Si vous souhaitez entraîner Nova sur un ensemble de données non raisonné tout en vous attendant à raisonner lors de l'inférence, vous pouvez désactiver le raisonnement pendant l'entraînement (`reasoning_enabled = false`) mais l'activer pour l'inférence. Bien que cette approche permette d'utiliser le raisonnement au moment de l'inférence, elle ne garantit pas une amélioration des performances par rapport à l'inférence sans raisonnement. En général, activez le raisonnement à la fois pour l'entraînement et pour l'inférence lorsque vous utilisez des ensembles de données de raisonnement, et désactivez-le pour les deux lorsque vous utilisez des ensembles de données sans raisonnement.

Réglez `reasoning_enabled: false` quand :

- Vos données d'entraînement ne contiennent pas de jetons de raisonnement
- Vous vous entraînez sur des tâches simples qui ne bénéficient pas d'étapes de raisonnement explicites
- Vous souhaitez optimiser la vitesse et réduire l'utilisation des jetons

Génération de données de raisonnement

Si votre jeu de données ne contient pas de traces de raisonnement, vous pouvez les créer à l'aide d'un modèle capable de raisonner tel que Nova Premier. Fournissez vos paires d'entrées-sorties au modèle et capturez son processus de raisonnement pour créer un ensemble de données augmenté par le raisonnement.

Utiliser des jetons de raisonnement pour l'entraînement

Lorsque le mode raisonnement est activé lors de l'entraînement, le modèle apprend à séparer le raisonnement interne de la réponse finale. Le processus de formation :

- Organise les données sous forme de triples : saisie, raisonnement et réponse

- Optimise à l'aide de la prédiction standard du jeton suivant les pertes causées à la fois par les jetons de raisonnement et de réponse
- Encourage le modèle à raisonner en interne avant de générer des réponses

Contenu de raisonnement efficace

Un contenu de raisonnement de haute qualité doit inclure :

- Réflexions et analyses intermédiaires
- Déductions logiques et étapes d'inférence
- Step-by-step approches de résolution de problèmes
- Liens explicites entre les étapes et les conclusions

Cela aide le modèle à développer la capacité de « réfléchir avant de répondre ».

Directives pour la préparation des jeux

Taille et qualité

- Taille recommandée : 2 000 à 10 000 échantillons
- Échantillons minimum : 200
- Priorité — La qualité prime sur la quantité. Assurez-vous que les exemples sont précis et bien annotés
- Alignement des applications — Le jeu de données doit refléter étroitement vos cas d'utilisation en production

Diversité

Incluez divers exemples qui :

- Couvrir la gamme complète des entrées attendues
- Représenter différents niveaux de difficulté
- Incluez des boîtiers et des variantes
- Évitez de trop vous adapter à des motifs étroits

Formatage de sortie

Spécifiez clairement le format de sortie souhaité dans les réponses de l'assistant :

- Structures JSON
- Tables
- Format CSV
- Formats personnalisés spécifiques à votre application

Conversations complexes

Pour les ensembles de données multitours, n'oubliez pas :

- La perte est calculée uniquement sur les tours de l'assistant, et non sur les tours de l'utilisateur
- Chaque réponse de l'assistant doit être correctement formatée
- Maintenez la cohérence à chaque tour de conversation

Liste de contrôle de qualité

- Taille de jeu de données suffisante (échantillons de 2 à 10 000)
- Des exemples variés couvrant tous les cas d'utilisation
- Formatage de sortie clair et cohérent
- Étiquettes et annotations précises
- Représentatif des scénarios de production
- Exempt de contradictions ou d'ambiguïtés

Téléchargement de vos données

Les ensembles de données doivent être téléchargés dans un compartiment accessible par les tâches de SageMaker formation. Pour plus d'informations sur la définition des autorisations appropriées, consultez la section [Conditions préalables](#).

Démarrage d'un poste de formation

Sélection des hyperparamètres et mise à jour de la recette

La configuration de Nova 2.0 est largement la même que celle de Nova 1.0. Une fois que les données d'entrée ont été téléchargées dans S3, utilisez la recette de [SageMaker HyperPod Recipes](#) sous le dossier Fine tuning. Pour Nova 2.0, voici quelques-uns des principaux hyperparamètres que vous pouvez mettre à jour en fonction du cas d'utilisation. Voici un exemple de la recette Nova 2.0 SFT PEFT. Pour l'URI de l'image du conteneur, `708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-TJ-SFT-V2-latest` utilisez-le pour exécuter une tâche de réglage fin SFT.

Veuillez utiliser la version 2.254.1 d' SageMaker AI PySDK pour une compatibilité stricte avec Nova Training. La mise à niveau du SDK vers la version 3.0 entraînera des modifications majeures. Support pour la version 3 d' SageMaker AI PySDK bientôt disponible.

Exemple de saisie

```
!pip install sagemaker==2.254.1
```

```
run:
  name: {peft_recipe_job_name}
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: {peft_model_name_or_path}
  data_s3_path: {train_dataset_s3_path} # SageMaker HyperPod (SMHP) only and not
compatible with SageMaker Training jobs. Note replace my-bucket-name with your real
bucket name for SMHP job
  replicas: 4                          # Number of compute instances for training, allowed
values are 4, 8, 16, 32
  output_s3_path: ""                   # Output artifact path (Hyperpod job-specific; not
compatible with standard SageMaker Training jobs). Note replace my-bucket-name with
your real bucket name for SMHP job

training_config:
  max_steps: 10                        # Maximum training steps. Minimal is 4.
  save_steps: 10                       # How many training steps the checkpoint will be
saved. Should be less than or equal to max_steps
  save_top_k: 1                        # Keep top K best checkpoints. Note supported only
for SageMaker HyperPod jobs. Minimal is 1.
  max_length: 32768                    # Sequence length (options: 8192, 16384, 32768
[default], 65536)
```

```

global_batch_size: 32          # Global batch size (options: 32, 64, 128)
reasoning_enabled: true        # If data has reasoningContent, set to true;
otherwise False

lr_scheduler:
  warmup_steps: 15             # Learning rate warmup steps. Recommend 15% of
max_steps                    #
  min_lr: 1e-6                 # Minimum learning rate, must be between 0.0 and 1.0

optim_config:                 # Optimizer settings
  lr: 1e-5                     # Learning rate, must be between 0.0 and 1.0
  weight_decay: 0.0            # L2 regularization strength, must be between 0.0
and 1.0
  adam_beta1: 0.9              # Exponential decay rate for first-moment estimates,
must be between 0.0 and 1.0
  adam_beta2: 0.95             # Exponential decay rate for second-moment
estimates, must be between 0.0 and 1.0

peft:                          # Parameter-efficient fine-tuning (LoRA)
  peft_scheme: "lora"          # Enable LoRA for PEFT
  lora_tuning:
    alpha: 64                  # Scaling factor for LoRA weights ( options: 32, 64,
96, 128, 160, 192),
    lora_plus_lr_ratio: 64.0

```

La recette contient également en grande partie les mêmes hyperparamètres que Nova 1.0. Les hyperparamètres les plus importants sont les suivants :

- **max_steps**— Le nombre d'étapes pendant lesquelles vous souhaitez exécuter la tâche. Généralement, pour une époque (une analyse complète de votre ensemble de données), le nombre d'étapes = nombre d'échantillons de données/taille globale du lot. Plus le nombre d'étapes est élevé et plus la taille globale de votre lot est petite, plus le travail sera long à exécuter.
- **reasoning_enabled**— Contrôle le mode de raisonnement de votre ensemble de données. Options :
 - **true**: Active le mode raisonnement (équivalent à un raisonnement élevé)
 - **false**: désactive le mode raisonnement

Remarque : Pour SFT, il n'existe aucun contrôle granulaire sur les niveaux d'effort de raisonnement. `reasoning_enabled: true` Le réglage permet une capacité de raisonnement complète.

- `peft.peft_scheme`— Le réglage sur « lora » permet un réglage précis basé sur le format PEFT. Le définir sur `null` (sans guillemets) permet d'affiner le classement complet.

Commencez le stage de formation

```
from sagemaker.pytorch import PyTorch

# define OutputDataConfig path
if default_prefix:
    output_path = f"s3://{bucket_name}/{default_prefix}/{sm_training_job_name}"
else:
    output_path = f"s3://{bucket_name}/{sm_training_job_name}"

output_kms_key = "<KMS key arn to encrypt trained model in Amazon-owned S3 bucket>" #
    optional, leave blank for Amazon managed encryption

recipe_overrides = {
    "run": {
        "replicas": instance_count, # Required
        "output_s3_path": output_path
    },
}

estimator = PyTorch(
    output_path=output_path,
    base_job_name=sm_training_job_name,
    role=role,
    disable_profiler=True,
    debugger_hook_config=False,
    instance_count=instance_count,
    instance_type=instance_type,
    training_recipe=training_recipe,
    recipe_overrides=recipe_overrides,
    max_run=432000,
    sagemaker_session=sagemaker_session,
    image_uri=image_uri,
    output_kms_key=output_kms_key,
    tags=[
        {'Key': 'model_name_or_path', 'Value': model_name_or_path},
    ]
)

print(f"\nsm_training_job_name:\n{sm_training_job_name}\n")
```

```
print(f"output_path:\n{output_path}")
```

```
from sagemaker.inputs import TrainingInput

train_input = TrainingInput(
    s3_data=train_dataset_s3_path,
    distribution="FullyReplicated",
    s3_data_type="Converse",
)

estimator.fit(inputs={"validation": val_input}, wait=False)
```

Note

La transmission d'un ensemble de données de validation n'est pas prise en charge pour le réglage fin supervisé de Nova 2.0.

Pour démarrer le travail :

- Mettez à jour la recette avec les chemins et les hyperparamètres de votre jeu de données
- Exécutez les cellules spécifiées dans le bloc-notes pour soumettre le travail de formation

Le bloc-notes gère la soumission des tâches et assure le suivi du statut.

Réglage fin du renforcement (RFT) avec les modèles Amazon Nova

Présentation de

Qu'est-ce que le RFT ?

Le réglage fin par renforcement (RFT) améliore les performances du modèle en s'entraînant sur les signaux de feedback (scores mesurables ou récompenses indiquant les performances du modèle) plutôt que sur les bonnes réponses exactes. Contrairement au réglage fin supervisé qui apprend à partir de paires d'entrées-sorties, le RFT utilise des fonctions de récompense pour évaluer les réponses du modèle et optimise le modèle de manière itérative afin de maximiser ces récompenses. Cette approche excelle lorsqu'il est difficile de définir la sortie exacte correcte, mais vous pouvez mesurer de manière fiable la qualité de réponse.

Quand utiliser le RFT

Utilisez RFT lorsque vous pouvez définir des critères de réussite clairs et mesurables, mais que vous avez du mal à fournir des résultats exacts pour la formation. C'est idéal pour :

- Tâches où la qualité est subjective ou multidimensionnelle (écriture créative, optimisation du code, raisonnement complexe)
- Scénarios comportant plusieurs solutions valides où certaines sont clairement meilleures que d'autres
- Applications nécessitant une amélioration itérative, une personnalisation ou le respect de règles métier complexes
- Cas où la collecte d'exemples étiquetés de haute qualité est coûteuse ou peu pratique

Meilleurs cas d'utilisation

La RFT excelle dans les domaines où la qualité de sortie peut être mesurée objectivement, mais où les réponses optimales sont difficiles à définir dès le départ :

- Résolution de problèmes mathématiques et génération de code
- Raisonnement scientifique et analyse de données structurées
- Tâches nécessitant un step-by-step raisonnement ou une résolution de problèmes en plusieurs étapes
- Applications équilibrant plusieurs objectifs (précision, efficacité, style)
- Scénarios dans lesquels le succès peut être vérifié par programmation par le biais de résultats d'exécution ou de mesures de performance

Modèles pris en charge

Nova Lite 2.0

Vue d'ensemble des formats de données

[Les données d'entraînement RFT doivent suivre le format OpenAI Reinforcement Fine-Tuning.](#)

Chaque exemple d'entraînement est un objet JSON contenant :

- Un messages tableau de tournants, d'utilisations system et user de rôles conversationnels
- Un `reference_answer` champ contenant le résultat attendu ou les critères d'évaluation pour le calcul de la récompense

Limitations actuelles

- Texte uniquement

Exemples de formats de données

Chaque exemple doit figurer sur une seule ligne de votre fichier JSONL, avec un objet JSON par ligne.

Chemistry problem

```
{
  "id": "chem-01",
  "messages": [
    {
      "role": "system",
      "content": "You are a helpful chemistry assistant"
    },
    {
      "role": "user",
      "content": "Calculate the molecular weight of caffeine (C8H10N4O2)"
    }
  ],
  "reference_answer": {
    "molecular_weight": 194.19,
    "unit": "g/mol",
    "calculation": "8(12.01) + 10(1.008) + 4(14.01) + 2(16.00) = 194.19"
  }
}
```

Math problem

```
{
  "id": "sample-001", // Optional
  "messages": [
    {
      "role": "system",
      "content": "You are a math tutor"
    },
    {
      "role": "user",
      "content": "Solve:  $2x + 5 = 13$ "
    }
  ]
}
```

```

    }
  ],
  "reference_answer": {
    "solution": "x = 4",
    "steps": ["2x = 13 - 5", "2x = 8", "x = 4"]
  }
}

```

Code problem

```

{
  "id": "code-002",
  "messages": [
    {
      "role": "system",
      "content": "You are a helpful programming assistant"
    },
    {
      "role": "user",
      "content": "Write a Python function that reverses a string without using
built-in reverse methods"
    }
  ],
  "reference_answer": {
    "code": "def reverse_string(s): \n    result = '' \n    for i in range(len(s)
- 1, -1, -1): \n        result += s[i] \n    return result",
    "test_cases": [
      {
        "input": "hello",
        "expected_output": "olleh"
      },
      {
        "input": "",
        "expected_output": ""
      },
      {
        "input": "a",
        "expected_output": "a"
      },
      {
        "input": "Python123",
        "expected_output": "321nohtyP"
      }
    ]
  }
}

```

```
    ],  
    "all_tests_pass": true  
  }  
}
```

Le `reference_answer` champ contient le résultat attendu ou les critères d'évaluation que votre fonction de récompense utilise pour évaluer la réponse du modèle. Il ne se limite pas aux sorties structurées : il peut contenir n'importe quel format permettant à votre fonction de récompense d'évaluer la qualité.

Recommandations relatives à la taille des jeux

Point de départ

- Minimum de 100 exemples de formation
- Minimum de 100 exemples d'évaluation

Approche axée sur l'évaluation

Avant d'investir dans une formation RFT à grande échelle, évaluez les performances de base de votre modèle :

- Haute performance (récompense supérieure à 95 %) — Le RFT n'est peut-être pas nécessaire : votre modèle fonctionne déjà bien
- Très mauvaise performance (0 % de récompense) — Passez d'abord à SFT pour établir les fonctionnalités de base
- Performances modérées — le RFT est probablement approprié

Commencer par un petit jeu de données vous permet de :

- Validez que votre fonction de récompense est exempte de bogues
- Confirmez que RFT est la bonne approche pour votre cas d'utilisation
- Identifiez et corrigez les problèmes à un stade précoce
- Testez le flux de travail avant de le mettre à l'échelle

Une fois validé, vous pouvez l'étendre à de plus grands ensembles de données pour améliorer encore les performances.

Caractéristiques des données d'entraînement efficaces

Clarté et cohérence

Les bons exemples de RFT nécessitent des données d'entrée claires et sans ambiguïté qui permettent un calcul précis des récompenses sur les différents modèles de sortie. Évitez le bruit dans vos données, notamment :

- Formatage incohérent
- Étiquettes ou instructions contradictoires
- Des instructions ambiguës
- Réponses de référence contradictoires

Toute ambiguïté induira le processus de formation en erreur et incitera le modèle à apprendre des comportements involontaires.

Diversité

Votre ensemble de données doit capturer toute la diversité des cas d'utilisation en production afin de garantir de solides performances réelles. Inclure :

- Différents formats d'entrée et boîtiers
- Cartographiez les modèles d'utilisation réels de la production à partir des journaux et des analyses des utilisateurs
- Échantillonner selon les types d'utilisateurs, les régions géographiques et les variations saisonnières
- Incluez des niveaux de difficulté allant des problèmes simples aux problèmes complexes

Considérations relatives aux fonctions de récompense

Concevez votre fonction de récompense pour un entraînement efficace :

- Exécuter en quelques secondes (et non en quelques minutes)
- Parallélisez efficacement avec Lambda
- Obtenez des scores cohérents et fiables
- Gérez les différents types de sorties de modèles avec élégance

Des fonctions de récompense rapides et évolutives permettent une itération rapide et une expérimentation rentable.

Propriétés supplémentaires

Le format de données RFT prend en charge les champs personnalisés au-delà des exigences de base du schéma (`messagesetreference_answer`). Cette flexibilité vous permet d'ajouter toutes les données supplémentaires dont votre fonction de récompense a besoin pour une évaluation correcte.

Note

Vous n'avez pas besoin de le configurer dans votre recette : le format de données prend en charge des champs supplémentaires par nature. Incluez-les simplement dans le JSON de vos données d'entraînement, et elles seront transmises à votre fonction de récompense metadata sur le terrain.

Propriétés supplémentaires communes

Exemples de champs de métadonnées :

- `task_id`— Identifiant unique pour le suivi
- `difficulty_level`— Indicateur de complexité du problème
- `domain`— Domaine ou catégorie
- `expected_reasoning_steps`— Nombre d'étapes de la solution

Exemple avec des propriétés supplémentaires

```
{
  "messages": [
    {
      "role": "system",
      "content": "You are a math tutor"
    },
    {
      "role": "user",
      "content": "Solve:  $2x + 5 = 13$ "
    }
  ],
  "reference_answer": {
```

```

    "solution": "x = 4",
    "steps": ["2x = 13 - 5", "2x = 8", "x = 4"]
  },
  "task_id": "algebra_001",
  "difficulty_level": "easy",
  "domain": "algebra",
  "expected_reasoning_steps": 3
}

```

Ces champs supplémentaires sont transmis à votre fonction de récompense lors de l'évaluation, ce qui permet une logique de notation sophistiquée adaptée à votre cas d'utilisation spécifique.

Configuration d'entraînement

Exemple de recette

```

# Note:
# This recipe can run on p5.48xlarge and p5en.48xlarge instance types.
run:
  name: "my-rft-run" # Unique run name (appears in logs/
artifacts).
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: nova-lite-2/prod
  data_s3_path: s3://<bucket>/<data file> # Training dataset in JSONL;
  replicas: 4
  reward_lambda_arn: arn:aws:lambda:<region>:<account-id>:function:<function-name>

## MLFlow configs
mlflow_tracking_uri: "" # Required for MLFlow
mlflow_experiment_name: "my-rft-experiment" # Optional for MLFlow. Note: leave this
field non-empty
mlflow_run_name: "my-rft-run" # Optional for MLFlow. Note: leave this field non-empty

## SMTJ GRPO Training specific configs
training_config:
  max_length: 8192 # Context window (tokens) for inputs
+prompt;
  global_batch_size: 16 # Total samples per optimizer step
  across all replicas (16/32/64/128/256).
  reasoning_effort: high # Enables reasoning mode high / low /
or null for non-reasoning

  rollout: # How responses are generated for GRPO/
advantage calc.

```

```

    advantage_strategy:
        number_generation: 2                    # N samples per prompt to estimate
advantages (variance vs cost).
    generator:
        max_new_tokens: 6000                  # Cap on tokens generated per sample
        set_random_seed: true                 # Seed generation for reproducibility
across runs.
        temperature: 1                       # Softmax temperature;
        top_k: 1                             # Sample only from top-K logits
    rewards:
        preset_reward_function: null          # Usage of preset reward functions
[exact_match]
    api_endpoint:
        lambda_arn: arn:aws:lambda:<region>:<account-id>:function:<function-name>
        lambda_concurrency_limit: 12          # Max concurrent Lambda invocations
(throughput vs. throttling).

    trainer:
        max_steps: 2                         # Steps to train for. One Step =
global_batch_size
        save_steps: 5
        test_steps: 1

    # RL parameters
    ent_coeff: 0.0                           # A bonus added to the policy loss that
rewards higher-output entropy.
    kl_loss_coef: 0.001                      # Weight on the KL penalty between the
actor (trainable policy) and a frozen reference model

    optim_config:                             # Optimizer settings
        lr: 5e-5                             # Learning rate
        weight_decay: 0.0                    # L2 regularization strength (0.0-1.0)
        adam_beta1: 0.9
        adam_beta2: 0.95

    peft:                                     # Parameter-efficient fine-tuning (LoRA)
        peft_scheme: "lora"                  # Enable LoRA for PEFT
        lora_tuning:
            alpha: 32
            lora_plus_lr_ratio: 64.0          # LoRA+ learning rate scaling factor (0.0-
100.0)

```

Formation RFT utilisant le LLM en tant que juge

Présentation de

Les grands modèles linguistiques (LLMs) sont de plus en plus utilisés comme juges dans les flux de travail de réglage fin par renforcement (RFT), fournissant des signaux de récompense automatisés qui guident l'optimisation des modèles. Dans cette approche, un LLM évalue les résultats du modèle par rapport à des critères spécifiques, qu'il s'agisse d'évaluer l'exactitude, la qualité, le respect du style ou l'équivalence sémantique, et attribue des récompenses qui stimulent le processus d'apprentissage par renforcement.

Cela est particulièrement utile pour les tâches où les fonctions de récompense traditionnelles sont difficiles à définir par programmation, par exemple pour déterminer si différentes représentations (comme « 1/3 », « 0,333 » et « un tiers ») sont sémantiquement équivalentes, ou pour évaluer des qualités nuancées telles que la cohérence et la pertinence. En utilisant des juges basés sur le LLM comme fonctions de récompense, vous pouvez adapter RFT à des domaines complexes sans nécessiter d'annotations humaines approfondies, ce qui permet une itération rapide et une amélioration continue de vos modèles dans divers cas d'utilisation, au-delà des problèmes d'alignement traditionnels.

Sélection du mode de raisonnement

Modes disponibles

- none — Aucun raisonnement (omettez le champ `reasoning_effort`)
- faible — Frais de raisonnement minimaux
- high — Capacité de raisonnement maximale (par défaut lorsque `reasoning_effort` est spécifié)

Note

Il n'existe pas d'option moyenne pour le RFT. Si le champ `reasoning_effort` est absent de votre configuration, le raisonnement est désactivé. Lorsque le raisonnement est activé, vous devez le définir sur 32768 `max_new_tokens` pour prendre en compte les résultats de raisonnement étendus.

Quand utiliser chaque mode

Utilisez un raisonnement raisonné pour :

- Tâches analytiques complexes
- Résolution de problèmes mathématiques
- Dédution logique en plusieurs étapes
- Tâches où la step-by-step réflexion apporte une valeur ajoutée

Utilisez `none` (omettez `reasoning_effort`) ou un raisonnement faible pour :

- Des requêtes factuelles simples
- Classifications directes
- Optimisation de la vitesse et des coûts
- Réponse simple aux questions

Compromis en termes de coûts et de performances

Les modes de raisonnement supérieurs augmentent :

- Durée et coût de la formation
- Latence et coût des inférences
- Capacité de modélisation pour les tâches de raisonnement complexes

Validation de votre juge LLM

Avant de déployer un modèle LLM-as-a-judge en production, vérifiez que les évaluations du modèle de juge correspondent au jugement humain. Cela implique :

- Mesurer les taux d'accord entre le juge LLM et les évaluateurs humains sur des échantillons représentatifs de votre tâche
- Veiller à ce que l'accord du LLM avec les humains atteigne ou dépasse les taux de concordance interhumains
- Identifier les biais potentiels dans le modèle de juge
- Renforcer l'assurance que le signal de récompense oriente votre modèle dans la direction prévue

Cette étape de validation permet de garantir que le processus d'évaluation automatisé produira des modèles répondant à vos critères de qualité de production.

Configuration Lambda pour LLM Judge

L'utilisation d'un LLM en tant que juge est une extension de l'utilisation des fonctions Lambda pour l'apprentissage par renforcement avec récompenses vérifiables (RLVR). Dans la fonction Lambda, vous appelez l'un des modèles hébergés sur Amazon Bedrock.

Exigences de configuration importantes :

Configuration	Exigence	Détails
Débit d'Amazon Bedrock	Quota suffisant	Assurez-vous que votre quota de débit pour le modèle Amazon Bedrock utilisé est suffisant pour votre charge de travail de formation
délai d'expiration Lambda	Délai d'attente prolongé	Configurez le délai d'expiration de votre fonction Lambda jusqu'à 15 minutes maximum. Le paramètre par défaut est de 3 secondes, ce qui est insuffisant pour les réponses du modèle Amazon Bedrock
Simultanéité Lambda	Concurrence accrue	Le Lambda est invoqué en parallèle pendant l'entraînement. Augmentez la simultanéité pour optimiser le débit disponible
Configuration de la recette	Correspond aux paramètres Lambda	La limite de simultanéité doit être configurée dans votre recette

Création et gestion d'emplois

Démarrage d'un poste de formation

Utilisez le modèle de carnet de tâches de SageMaker formation : <https://docs.aws.amazon.com/sagemaker/latest/dg/nova-fine-tuning-training-job.html#nova-model-training-jobs-notebook>

Exigences relatives à l'instance

Le conteneur prend en charge à la fois les formations Full-Rank et LoRa :

- Entraînement LoRa — instances 2/4/6/8 × p5.48xlarge ou p5en.48xlarge

- Entraînement complet : 2/4/6/8 × instances p5.48xlarge (obligatoire)

Formation en matière de surveillance

Les journaux d'entraînement incluent des mesures complètes à chaque étape. Principales catégories de mesures :

Métriques de récompenses

- `critic/rewards/mean`, `critic/rewards/max`, `critic/rewards/min` — Distribution de récompenses
- `val-score/rewards/mean@1` — Récompenses de validation

Comportement du modèle

- `actor/entropy` — Variation des politiques (plus élevée = plus exploratoire)

Santé de l'entraînement

- `actor/pg_loss` — Perte liée au gradient de politique
- `actor/pg_clipfrac` — Fréquence des mises à jour découpées
- `actor/grad_norm` — Ampleur du gradient

Caractéristiques de la réponse

- `prompt_length/mean`, `prompt_length/max`, `prompt_length/min` — Statistiques des jetons d'entrée
- `response_length/mean`, `response_length/max`, `response_length/min` — Statistiques des jetons de sortie
- `response/aborted_ratio` — Taux de génération incomplète (0 = tout est terminé)

Performances

- `perf/throughput` — Débit de formation
- `perf/time_per_step` — Durée par étape d'entraînement
- `timing_per_token_ms/*` — Délais de traitement par jeton

Utilisation des ressources

- `perf/max_memory_allocated_gb`, `perf/max_memory_reserved_gb` — Mémoire GPU
- `perf/cpu_memory_used_gb` — Mémoire du processeur

Utilisation de modèles affinés

Une fois l'entraînement terminé, le point de contrôle final du modèle est enregistré à l'emplacement de sortie que vous avez spécifié. Le chemin du point de contrôle est disponible en :

- Journaux d'entraînement
- `manifest.json` fichier dans l'emplacement de sortie Amazon S3 (défini par `output_s3_uri` dans votre bloc-notes)

Limites et meilleures pratiques

Limites

- Délai Lambda : les fonctions de récompense doivent être terminées dans les 15 minutes (pour éviter les processus incontrôlables et gérer les coûts)
- Un tour uniquement — Les conversations à plusieurs tours ne sont pas prises en charge
- Exigences en matière de données — Nécessite une diversité suffisante ; peine à obtenir de maigres récompenses (< 5 % d'exemples positifs)
- Coût de calcul : plus coûteux que le réglage fin supervisé
- Aucune donnée multimodale : seul le type de données texte est pris en charge

Bonnes pratiques

Commencez petit

- Commencez par 100 à 200 exemples
- Valider l'exactitude de la fonction de récompense
- Échelle progressive en fonction des résultats

Évaluation préalable à la formation

- Testez les performances du modèle de référence avant RFT

- Si les récompenses sont toujours de 0 %, utilisez d'abord SFT pour établir les capacités de base
- Si les récompenses sont supérieures à 95 %, le RFT peut être inutile

Surveiller l'entraînement

- Suivez les scores moyens des récompenses et leur distribution
- Attention au surajustement (les récompenses d'entraînement augmentent tandis que les récompenses de validation diminuent)
- Recherchez des modèles inquiétants :
 - Plafond des récompenses inférieur à 0,15
 - Variation croissante des récompenses au fil du temps
 - Baisse des performances de validation

Optimisez les fonctions de récompense

- Exécuter en quelques secondes (et non en quelques minutes)
- Minimiser les appels d'API externes
- Utilisez des algorithmes efficaces
- Mettre en œuvre une gestion appropriée des erreurs
- Profitez de la mise à l'échelle parallèle de Lambda

Stratégie d'itération

Si les récompenses ne s'améliorent pas :

- Ajuster le design des fonctions de récompense
- Augmenter la diversité des jeux de données
- Ajoutez des exemples plus représentatifs
- Vérifiez que les signaux de récompense sont clairs et cohérents

Capacités avancées : Nova Forge

Pour les utilisateurs nécessitant des fonctionnalités avancées allant au-delà des limites standard du RFT, Nova Forge est disponible sous forme de service d'abonnement payant offrant :

- Support de conversation à plusieurs tours
- Fonctions de récompense avec un temps d'exécution supérieur à 15 minutes
- Algorithmes et options de réglage supplémentaires
- Modifications de recettes d'entraînement personnalisées
- State-of-the-art Techniques d'IA

Nova Forge fonctionne SageMaker HyperPod et est conçu pour aider les entreprises clientes à créer leurs propres modèles de pointe.

Commandes et astuces utiles

Une collection de [scripts d'observabilité](#) est disponible pour aider à suivre l'état et la progression des tâches de formation.

Les scripts disponibles sont les suivants :

- Activation des notifications par e-mail pour les mises à jour du statut des tâches de formation
- Obtenir des estimations du temps de formation en fonction de la configuration des tâches
- Obtenir des estimations de la durée prévue de la formation pour les tâches en cours

Installation

Note

Veillez à actualiser vos AWS informations d'identification avant d'utiliser l'un des scripts suivants.

```
pip install boto3
git clone https://github.com/aws-samples/amazon-nova-samples.git
cd amazon-nova-samples/customization/SageMakerUilts/SageMakerJobsMonitoring/
```

Usage de base

```
# Enabling email notifications for training job status updates
python enable_sagemaker_job_notifs.py --email test@amazon.com test2@gmail.com --region
us-east-1 --platform SMTJ
```

```
Creating resources.....
```

```
Please check your email for a subscription confirmation email, and click 'Confirm  
subscription' to start receiving job status email notifications!
```

```
You'll receive the confirmation email within a few minutes.
```

```
# Obtaining training time estimates based on job configurations
```

```
python get_training_time_estimate.py
```

```
# Obtaining approximations for how long training is expected to take for in-progress  
jobs
```

```
python get-training-job-progress.py --region us-east-1 --job-name my-training-job --  
num-dataset-samples 1000
```

Cliquez [ici](#) pour plus de détails et d'exemples.

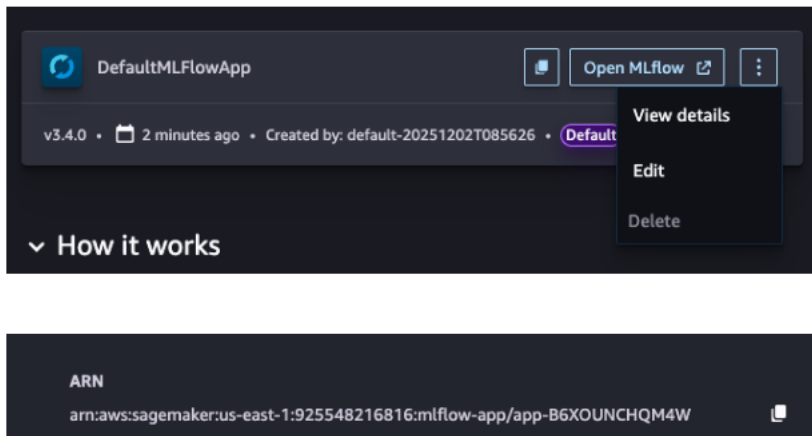
Suivi des progrès au fil des itérations

Vous pouvez suivre les métriques via MLflow.

Nova Customization - MLFlow configuration pour SageMaker HyperPod

Pour permettre à votre SageMaker HyperPod environnement de générer des métriques MLFlow, il est nécessaire de procéder à une configuration supplémentaire.

1. Ouvrez Amazon SageMaker AI
2. Sélectionnez un SageMaker studio
 - a. Si un profil a déjà été créé, sélectionnez « Open Studio ».
 - b. Si aucun profil n'est créé, sélectionnez « Créer un SageMaker domaine » pour en configurer un
3. Sélectionnez MLFlow. Si aucune MLFlow application n'a été créée, sélectionnez « Créer une MLFlow application »
4. Cliquez sur le copy/paste bouton ou sur l'élément de menu « Afficher les détails » de l'application ML Flow pour obtenir l'ARN. Vous en aurez besoin lorsque vous soumettrez votre offre de formation.



5. Dans le rôle d'exécution du HyperPod cluster, ajoutez la politique suivante. Cela permettra au HyperPod cluster d'appeler l' MLFlow API pour publier des métriques.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sagemaker-mlflow:*",
      "Resource": [
        "arn:aws:sagemaker:us-east-1:372836560492:mlflow-app/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:ListMlflowTrackingServers",
        "sagemaker:CallMlflowAppApi"
      ],
      "Resource": "*"
    }
  ]
}
```

Soumission d'une tâche via la CLI

Spécifiez 4 nouveaux paramètres de remplacement, soit dans la ligne de commande, soit dans la recette yaml.

1. `mlflow_tracking_uri`: L'ARN de l' MLFlow application
2. `mlflow_experiment_name`: nom de cette série de l'expérience
3. `mlflow_experiment_name`: nom de l'expérience dans laquelle les métriques seront stockées MLFlow
4. `mlflow_run_name`: nom de cette expérience

Ligne de commande

```
--override-parameters '{"recipes.run.mlflow_tracking_uri": "arn:aws:sagemaker:us-east-1:925548216816:mlflow-app/app-B6X0UNCHQM4W", "recipes.run.mlflow_experiment_name": "myuser-sft-lora-exp1", "recipes.run.mlflow_run_name": "myuser-sft-lora-exp1-202512181940"}'
```

yaml :

```
## Run config
run:
  mlflow_tracking_uri: "arn:aws:sagemaker:us-east-1:925548216816:mlflow-app/app-B6X0UNCHQM4W"
  mlflow_experiment_name: "myuser-sft-lora-exp1"
  mlflow_run_name: "myuser-sft-lora-exp1-202512181940"
```

Soumission d'une tâche via l'interface utilisateur de SageMaker Studio

MLFlow l'intégration est déjà intégrée à l'expérience utilisateur de SageMaker Studio. Lorsque vous soumettez une offre de formation, il vous suffit d'indiquer MLFlow l'instance d'application à utiliser.

1. Dans SageMaker Studio, accédez à Modèles > Nova 2.0 Lite > Personnaliser > Personnaliser avec l'interface utilisateur.
2. Développez la section Configuration avancée
3. Sélectionnez l' MLFlow application à laquelle vous souhaitez envoyer les statistiques d'entraînement. Vous pouvez également définir le nom de votre expérience et l'exécuter ici.

MLflow

MLflow Resource *

MLflow resource selected for experiment tracking.

DefaultMLFlowApp

Experiment name

Default name applied. Enter a new name if desired.

myuser-sft-lora-exp1

Experiment run

The run name is autogenerated by default. Specify a new name if desired.

myuser-sft-lora-exp1-1766499113356

Soumettre une offre d'emploi via le AWS AWS CLI

Si vous utilisez le AWS AWS CLI, vous devez créer une MLflow application et la transmettre comme entrée à la demande d'API de la tâche de formation.

```
mlflow_app_name="<enter your MLflow app name>"
role_arn="<enter your role ARN>"
bucket_name="<enter your bucket name>"
region="<enter your region>"

mlflow_app_arn=$(aws sagemaker create-mlflow-app \
  --name $mlflow_app_name \
  --artifact-store-uri "s3://$bucket_name" \
  --role-arn $role_arn \
  --region $region)
```

Création d'une MLflow application

Utilisation de l'interface utilisateur de Studio : si vous créez une tâche de formation via l'interface utilisateur de Studio, une MLflow application par défaut est créée automatiquement et sélectionnée par défaut dans les options avancées.

Utilisation de la CLI : si vous utilisez la CLI, vous devez créer une MLflow application et la transmettre comme entrée à la demande d'API de la tâche de formation.

```
mlflow_app_name="<enter your MLflow app name>"
```

```
role_arn="<enter your role ARN>"
bucket_name="<enter your bucket name>"
region="<enter your region>"

mlflow_app_arn=$(aws sagemaker create-mlflow-app \
  --name $mlflow_app_name \
  --artifact-store-uri "s3://$bucket_name" \
  --role-arn $role_arn \
  --region $region)
```

Accédez à l' MLflow application

À l'aide de la CLI : créez une URL pré-signée pour accéder à l'interface utilisateur de l' MLflow application :

```
aws sagemaker create-presigned-mlflow-app-url \
  --arn $mlflow_app_arn \
  --region $region \
  --output text
```

Utilisation de l'interface utilisateur de Studio : l'interface utilisateur de Studio affiche les indicateurs clés enregistrés MLflow et fournit un lien vers l'interface utilisateur de MLflow l'application.

Indicateurs clés à suivre

Surveillez ces indicateurs au fil des itérations pour évaluer les améliorations et suivre l'avancement des travaux :

Pour SFT

- Courbes de pertes d'entraînement
- Nombre d'échantillons consommés et temps de traitement des échantillons
- Précision des performances sur des ensembles de test conservés
- Conformité au format (par exemple, débit de sortie JSON valide)
- Perplexité quant aux données d'évaluation spécifiques à un domaine

Pour RFT

- Scores de récompense moyens au cours de l'entraînement

- Répartition des récompenses (pourcentage de réponses offrant des récompenses élevées)
- Tendances en matière de récompenses de validation (attention aux surajustements)
- Taux de réussite spécifiques aux tâches (par exemple, taux de réussite de l'exécution du code, précision des problèmes mathématiques)

Général

- Comparez les deltas de performance entre les itérations
- Scores d'évaluation humaine sur des échantillons représentatifs
- Métriques de production (en cas de déploiement itératif)

Déterminer quand arrêter

Arrêtez d'itérer lorsque :

- Plateaux de performance : l'entraînement supplémentaire n'améliore plus de manière significative les indicateurs cibles
- Le changement de technique est utile : si une technique plafonne, essayez de la changer (par exemple, SFT → RFT → SFT) pour dépasser les plafonds de performance
- Indicateurs cibles atteints : vos critères de réussite sont remplis
- Régression détectée : les nouvelles itérations dégradent les performances (voir les procédures de restauration ci-dessous)

Pour les procédures d'évaluation détaillées, reportez-vous à la section Évaluation.

Évaluation de votre modèle entraîné par SageMaker l'IA

Le but du processus d'évaluation est d'évaluer les performances du modèle entraîné par rapport à des points de référence ou à un jeu de données personnalisé. Le processus d'évaluation comprend généralement des étapes visant à créer une formule d'évaluation pointant vers le modèle entraîné, à spécifier des jeux de données et des métriques d'évaluation, à soumettre une tâche distincte pour l'évaluation et à évaluer par rapport à des points de référence standard ou à des données personnalisées. Le processus d'évaluation produira des métriques de performance stockées dans votre compartiment Amazon S3.

 Note

Le processus d'évaluation décrit dans cette rubrique est un processus hors ligne. Le modèle est testé par rapport à des points de référence fixes avec des réponses prédéfinies, plutôt que d'être évalué en temps réel ou par le biais d'interactions en direct avec les utilisateurs. Pour une évaluation en temps réel, vous pouvez tester le modèle après son déploiement sur Amazon Bedrock en appelant [Amazon Bedrock](#) Runtime. APIs

Rubriques

- [Conditions préalables](#)
- [Tâches de point de référence disponibles](#)
- [Configurations spécifiques à l'évaluation](#)
- [Exécution de tâches d'entraînement d'évaluation](#)
- [Évaluation et analyse des résultats d'évaluation](#)
- [Bonnes pratiques d'évaluation et dépannage](#)
- [Sous-tâches disponibles](#)
- [Évaluation du modèle de raisonnement](#)
- [Évaluation RFT](#)
- [Mettre en œuvre des fonctions de récompense](#)

Conditions préalables

Avant de commencer une tâche d'entraînement d'évaluation, notez les éléments suivants.

- Un modèle SageMaker Amazon Nova entraîné par l'IA dont vous souhaitez évaluer les performances.
- Recette de base Amazon Nova à des fins d'évaluation. Pour de plus amples informations, veuillez consulter [???](#).

Tâches de point de référence disponibles

Un exemple de package de code est disponible qui montre comment calculer les mesures de référence à l'aide de la fonctionnalité d'évaluation du modèle d' SageMaker intelligence artificielle

pour Amazon Nova. Pour accéder aux packages de code, consultez [Sample-nova-lighteval-custom-task](#).

Voici une liste des points de référence standard pris en charge. Vous pouvez spécifier les points de référence suivants dans le paramètre `eval_task`.

Points de référence disponibles pour l'évaluation des modèles

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
mmlu	Texte	Compréhension linguistique multitâche : teste les connaissances dans 57 domaines.	précision	zs_cot	Oui
mmlu_pro	Texte	MMLU – Sous-ensemble professionnel : se concentre sur les domaines professionnels tels que le droit, la médecine, la comptabilité et l'ingénierie.	précision	zs_cot	Non
bbh	Texte	Tâches de raisonnement avancées : ensemble de problèmes	précision	fs_cot	Oui

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
		difficiles qui mettent à l'épreuve des compétences cognitives et de résolution de problèmes de haut niveau.			
gpqa	Texte	Réponses aux questions de physique générale : évalue la compréhension des concepts de physique et les capacités de résolution de problèmes connexes.	précision	zs_cot	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
math	Texte	Résolution de problèmes mathématiques : mesure le raisonnement mathématique dans des domaines tels que l'algèbre, le calcul et les problèmes de mots.	exact_match	zs_cot	Oui
strong_reject	Texte	Tâche de contrôle qualité : teste la capacité du modèle à détecter et à rejeter le contenu inapproprié, préjudiciable ou incorrect.	deflection	zs	Oui

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
ifeval	Texte	Évaluation suivant les instructions : évalue la précision avec laquelle un modèle suit les instructions données et exécute les tâches conformément aux spécifications.	précision	zs	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
gen_qa	Multimodal (image)	Évaluation personnalisée du jeu de données : vous permet de fournir votre propre ensemble de données à des fins d'analyse comparative, en comparant les résultats du modèle aux réponses de référence avec des indicateurs tels que ROUGE et BLEU. gen_qa prend en charge l'inférence d'images pour les modèles basés sur Amazon Nova Lite ou Amazon Nova Pro.	tout	gen_qa	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
		Supporte également Bring-Your-Own Metrics Lambda. (Pour l'évaluation RFT, veuillez utiliser la recette d'évaluation RFT)			
mmmu	Multimodale	Compréhension multidisciplinaire et multimodale massive (MMMU) — Indice de référence de niveau universitaire comprenant des questions à choix multiples et ouvertes provenant de 30 disciplines.)	précision	zs_cot	Oui

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
llm_judge	Texte	LLM-as-a-Judge Comparaison des préférences — Utilise un modèle Nova Judge pour déterminer la préférence entre les réponses appariées (B par rapport à A) pour vos invites, en calculant la probabilité que B soit préféré à A.	tout	judge	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
mm_llm_judge	Multimodal (image)	Ce nouveau benchmark se comporte de la même manière que le modèle basé sur le texte ci-dessus llm_judge . La seule différence est qu'il prend en charge l'inférence d'image.	tout	judge	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
rubrique_ilm_judge	Texte	Rubric Judge est un modèle d' LLM-as-a-judgeévaluation amélioré basé sur Nova 2.0 Lite. Contrairement au modèle de juge original qui ne fournit que des verdicts de préférence, Rubric Judge génère de manière dynamique des critères d'évaluation personnalisés adaptés à chaque invite et attribue des scores granulaires sur plusieurs dimensions.	tout	judge	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
aime_2024	Texte	AIME 2024 - Problèmes de l'examen de mathématiques sur invitation américain testant le raisonnement mathématique avancé et la résolution de problèmes	exact_match	zs_cot	Non
calendar_planning	Texte	Natural Plan - Planification du calendrier, test des tâches, capacités de planification pour planifier des réunions sur plusieurs jours et plusieurs personnes	exact_match	fs	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
humaneval	Texte	HumanEval - Un ensemble de données de référence conçu pour évaluer les capacités de génération de code des grands modèles de langage	pass@1	zs	Non

Configurations spécifiques à l'évaluation

Vous trouverez ci-dessous une description des principaux composants de la formule et des conseils sur la manière de les modifier en fonction de vos cas d'utilisation.

Compréhension et modification de vos formules

Configuration générale de l'exécution

```
run:
  name: eval_job_name
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: nova-lite-2/prod # or s3://escrow_bucket/model_location
  replicas: 1
  data_s3_path: ""
  mlflow_tracking_uri: ""
  mlflow_experiment_name : ""
  mlflow_run_name : ""
```

- **name** : nom descriptif de votre tâche d'évaluation.
- **model_type** : spécifie la variante du modèle Nova à utiliser. Ne modifiez pas ce champ manuellement. Voici les options :

- `amazon.nova-micro-v1:40:128 km`
- `amazon.nova-lite-v1:30:300 km`
- `amazon.nova-pro-v1:30:300 km`
- `amazon.nova-2-lite-v 1:0:256 k`
- `model_name_or_path` : le chemin vers le modèle de base ou le chemin s3 pour le point de contrôle post-entraîné. Voici les options :
 - `nova-micro/prod`
 - `nova-lite/prod`
 - `nova-pro/prod`
 - `nova-lite-2/prod`
 - Chemin S3 pour le chemin de point de contrôle post-entraîné (`s3:customer-escrow-111122223333-smjtj-<unique_id>/<training_run_name>`)

 Note

Évaluation des modèles post-entraînés

Pour évaluer un modèle post-entraîné après une tâche d'entraînement Nova SFT, suivez ces étapes après avoir exécuté une tâche d'entraînement réussie. À la fin des journaux d'entraînement, vous verrez le message « L'entraînement est terminé ». Vous trouverez également un fichier `manifest.json` dans votre compartiment de sortie contenant l'emplacement de votre point de contrôle. Ce fichier sera situé dans un fichier `output.tar.gz` à votre emplacement S3 de sortie. Pour procéder à l'évaluation, utilisez ce point de contrôle en le définissant comme valeur pour `run.model_name_or_path` dans la configuration de votre formule.

- `replica`: le nombre d'instances de calcul à utiliser pour l'inférence distribuée (exécution de l'inférence sur plusieurs nœuds). Définissez `replica > 1` pour activer l'inférence multi-nœuds, ce qui accélère l'évaluation. Si `instance_count` les deux `replica` sont spécifiés, `instance_count` prévaut. Notez que les répliques multiples ne s'appliquent qu'aux tâches de formation à l' SageMaker IA, et non.
- `data_s3_path` : le chemin d'accès Amazon S3 du jeu de données d'entrée. Ce champ est obligatoire mais doit toujours être laissé vide.
- `mlflow_tracking_uri`: (Facultatif) L'emplacement du serveur de MLflow suivi (uniquement nécessaire sur SMHP)

- `mlflow_experiment_name`: (Facultatif) Nom de l'expérience pour regrouper les essais de machine learning associés
- `mlflow_run_name`: (Facultatif) Nom personnalisé pour un entraînement spécifique effectué dans le cadre d'une expérience

Configuration d'évaluation

```
evaluation:  
  task: mmlu  
  strategy: zs_cot  
  subtask: abstract_algebra  
  metric: accuracy
```

- `task` : spécifie le point de référence ou la tâche d'évaluation à utiliser. Les tâches prises en charge incluent :
 - `mmlu`
 - `mmlu_pro`
 - `bbh`
 - `gpqa`
 - `math`
 - `strong_reject`
 - `gen_qa`
 - `ifeval`
 - `mmmu`
 - `llm_judge`
 - `mm_llm_judge`
 - `rubric_llm_judge`
 - `aime_2024`
 - `calendar_scheduling`
 - `humaneval`
- `strategy` : définit l'approche d'évaluation.

- `zs_cot`: Chaîne de pensée zéro : une approche visant à susciter de grands modèles linguistiques qui encouragent le step-by-step raisonnement sans nécessiter d'exemples explicites.
- `fs_cot`: Chaîne de pensée en plusieurs points : approche qui fournit quelques exemples de step-by-step raisonnement avant de demander au modèle de résoudre un nouveau problème.
- `zs` : zéro coup, approche pour résoudre un problème sans aucun exemple d'entraînement préalable.
- `gen_qa` : stratégie spécifique pour apporter votre propre jeu de données.
- `jjudge`: Stratégie spécifique à Nova LLM en tant que juge `etmm_llm_judge`.
- `subtask`: Facultatif. Composantes spécifiques de la tâche d'évaluation. Pour obtenir la liste complète des sous-tâches, consultez [???](#).
 - Vérifiez les sous-tâches prises en charge dans Tâches de référence disponibles.
 - Ce champ devrait être supprimé s'il n'y a pas de points de référence pour les sous-tâches.
- `metric` : métrique d'évaluation à utiliser.
 - `accuracy` : pourcentage de bonnes réponses.
 - `exact_match` : pour un point de référence mathématique, renvoie le taux auquel les chaînes prédites en entrée correspondent exactement à leurs références.
 - `deflection` : pour un point de référence de rejet fort, renvoie une déflexion relative par rapport au modèle de base et aux métriques d'importance des différences.
 - `all`:

Pour `gen_qa`, apportez votre propre point de référence de jeu de données, renvoyez les métriques suivantes :

- `rouge1` : mesure le chevauchement des unigrammes (mots uniques) entre le texte généré et le texte de référence.
- `rouge2` : mesure le chevauchement des bigrammes (deux mots consécutifs) entre le texte généré et le texte de référence.
- `rougeL` : mesure la plus longue sous-séquence commune entre les textes, en tenant compte des lacunes dans la correspondance.
- `exact_match` : score binaire (0 ou 1) indiquant si le texte généré correspond exactement au texte de référence, caractère par caractère.
- `quasi_exact_match` : similaire à la correspondance exacte, mais plus indulgent, ignorant généralement les différences entre majuscules et minuscules, la ponctuation et les espaces.

- `f1_score` : moyenne harmonique de précision et de rappel, mesurant le chevauchement des mots entre les réponses prédites et les réponses de référence.
- `f1_score_quasi` : similaire à `f1_score` mais avec une correspondance plus souple, en utilisant une comparaison de texte normalisée qui ignore les différences mineures.
- `bleu` : mesure la précision des correspondances n-grammes entre le texte généré et le texte de référence, couramment utilisé dans l'évaluation des traductions.

Pour `llm_judge` et `etmm_llm_judge`, apportez votre propre référence de jeu de données, renvoyez les métriques suivantes :

- `a_scores` : nombre de victoires pour `response_A` sur les passes d'évaluation avant et arrière.
- `a_scores_stderr` : erreur standard de `response_A_scores` pour les jugements par paires.
- `b_scores` : mesure le nombre de victoires pour `response_B` sur les passes d'évaluation avant et arrière.
- `a_scores_stderr` : erreur standard de `response_B_scores` pour les jugements par paires.
- `ties` : nombre de jugements où `response_A` et `response_B` sont évaluées comme égales.
- `ties_stderr` : erreur standard de `ties` pour les jugements par paires.
- `inference_error` : nombre de jugements qui n'ont pas pu être correctement évalués.
- `score` : score agrégé basé sur les victoires obtenues lors des passes avant et arrière pour `response_B`.
- `score_stderr` : score agrégé basé sur les victoires obtenues lors des passes avant et arrière pour `response_B`.
- `inference_error_stderr` : erreur type du score agrégé pour les jugements par paires.
- `winrate` : probabilité que la réponse `response_B` soit préférée à `response_A`, calculée à l'aide de la probabilité de Bradley-Terry.
- `lower_rate` : limite inférieure (2,5 percentile) du taux de victoire estimé à partir d'un échantillonnage d'amorçage.
- `upper_rate` : limite supérieure (97,5e percentile) du taux de victoire estimé à partir d'un échantillonnage d'amorçage.

```
inference:
  max_new_tokens: 2048
  top_k: -1
  top_p: 1.0
  temperature: 0
  top_logprobs: 10
  reasoning_effort: null # options: low/high to enable reasoning or null to disable reasoning
```

- `max_new_tokens` : nombre maximum de jetons à générer. Doit être un nombre entier. (Non disponible pour le juge LLM)
- `top_k` : nombre de jetons présentant la probabilité la plus élevée à prendre en compte. Doit être un nombre entier.
- `top_p` : seuil de probabilité cumulé pour l'échantillonnage de jetons. Doit être une valeur à virgule flottante comprise entre 1,0 et 0,0.
- `temperature` : caractère aléatoire dans la sélection des jetons (plus élevé = plus aléatoire), conservez 0 pour que le résultat soit déterministe. Type flottant, la valeur minimum est 0.
- `top_logprobs`: Le nombre de meilleures probs logprobs à renvoyer dans la réponse d'inférence. Cette valeur doit être un entier compris entre 0 et 20. Les logprobs contiennent les jetons de sortie pris en compte et les probabilités de journalisation de chaque jeton de sortie renvoyé dans le contenu du message.
- `reasoning_effort`: contrôle le comportement de raisonnement pour les modèles capables de raisonner. Défini `reasoning_effort` uniquement lorsque `model_type` spécifie un modèle capable de raisonner (actuellement `amazon.nova-2-lite-v1:0:256k`). Les options disponibles sont nulles (valeur par défaut si elle n'est pas définie ; désactive le raisonnement), faibles ou élevées.

Format de sortie de probabilité logarithmique

Lorsque `top_logprobs` est configuré dans vos paramètres d'inférence, le résultat de l'évaluation inclut les probabilités logarithmiques au niveau des jetons dans les fichiers parquet. Chaque position de jeton contient un dictionnaire des meilleurs jetons candidats avec leurs probabilités logarithmiques dans la structure suivante :

```
{
  "Ġint": {"logprob_value": -17.8125, "decoded_value": " int"},
  "Ġthe": {"logprob_value": -2.345, "decoded_value": " the"}
```

```
}
```

Chaque entrée de jeton contient :

- `logprob_value`: valeur de probabilité logarithmique pour le jeton
- `decoded_value`: la représentation sous forme de chaîne décodée lisible par l'homme du jeton

Le jeton tokenizer brut est utilisé comme clé de dictionnaire pour garantir l'unicité, tout en `decoded_value` fournissant une interprétation lisible.

Exemples de formules d'évaluation

Amazon Nova propose quatre types de formules d'évaluation différents. Toutes les recettes sont disponibles dans le [GitHub référentiel de SageMaker HyperPod recettes](#).

Formules de point de référence de texte général

Ces formules vous permettent d'évaluer les capacités fondamentales des modèles Amazon Nova grâce à une suite complète de points de référence basés uniquement sur le texte.

Format de formule : `xxx_general_text_benchmark_eval.yaml`.

Apportez vos propres formules de point de référence de jeu de données

Ces formules vous permettent d'utiliser votre propre jeu de données à des fins d'analyse comparative et de comparer les résultats du modèle aux réponses de référence à l'aide de différents types de métriques.

Format de formule : `xxx_bring_your_own_dataset_eval.yaml`.

Apportez vos propres exigences en matière de jeux de données

Format de fichier :

- Fichier `gen_qa.jsonl` unique contenant des exemples d'évaluation. Le nom du fichier doit être exactement `gen_qa.jsonl`.
- Vous devez télécharger votre ensemble de données sur un emplacement S3 auquel les postes de formation en SageMaker IA peuvent accéder.
- Le fichier doit suivre le format de schéma requis pour le jeu de données général de questions-réponses.

Exigences relatives au format du schéma - Chaque ligne du `.jsonl` fichier doit être un objet JSON avec les champs suivants.

- Champs obligatoires.

`query` : chaîne contenant la question ou l'instruction nécessitant une réponse.

`response` : chaîne contenant le résultat attendu du modèle.

- Champs facultatifs.

`system` : chaîne contenant l'invite système qui définit le comportement, le rôle ou la personnalité du modèle d'IA avant qu'il ne traite la requête.

`images`: Tableau contenant une liste d'objets avec des attributs de données (chaînes d'image codées en Base64).

`metadata`: chaîne contenant les métadonnées associées à l'entrée à des fins de balisage.

Exemple d'entrée

```
{
  "system": "You are an English major with top marks in class who likes to give minimal word responses: ",
  "query": "What is the symbol that ends the sentence as a question",
  "response": "?"
}{
  "system": "You are a pattern analysis specialist who provides succinct answers: ",
  "query": "What is the next number in this series? 1, 2, 4, 8, 16, ?",
  "response": "32"
}{
  "system": "You have great attention to detail and follow instructions accurately: ",
  "query": "Repeat only the last two words of the following: I ate a hamburger today and it was kind of dry",
  "response": "of dry"
}{
  "system": "Image inference: ",
  "query": "What is the number in the image? Please just use one English word to answer.",
  "response": "two",
  "images": [
    {
      "data": "data:image/png;base64,iVBORw0KGgoA ..."
    }
  ]
}
```

```
}  
]  
}
```

Pour utiliser votre ensemble de données personnalisé, modifiez votre recette d'évaluation en ajoutant les champs obligatoires suivants sans modifier la configuration existante :

```
evaluation:  
  task: gen_qa  
  strategy: gen_qa  
  metric: all
```

Limites

- Un seul fichier `.jsonl` est autorisé par évaluation.
- Le fichier doit suivre strictement le schéma défini.

Apportez vos propres indicateurs

Vous pouvez utiliser vos propres indicateurs pour personnaliser entièrement le flux de travail d'évaluation de votre modèle grâce à des fonctionnalités de prétraitement, de post-traitement et de mesures personnalisées. Le prétraitement vous permet de traiter les données d'entrée avant de les envoyer au serveur d'inférence, tandis que le post-traitement vous permet de personnaliser le calcul des métriques et de renvoyer des métriques personnalisées en fonction de vos besoins.

Suivez ces étapes pour créer vos propres indicateurs grâce au SDK d'évaluation personnalisé.

1. Si vous ne l'avez pas encore fait, [créez une AWS Lambda fonction](#) dans votre Compte AWS premier.
2. Téléchargez le `nova-custom-eval-layer.zip` fichier prédéfini depuis le [GitHub référentiel](#). Vous pouvez utiliser ce SDK d'évaluation personnalisé open source Nova pour valider les charges utiles d'entrée et de sortie pour votre fonction personnalisée et fournir une interface unifiée pour l'intégration à l'évaluation des métriques « Bring your own » de Nova pendant l'entraînement.
3. Téléchargez la couche Lambda personnalisée à l'aide de la commande suivante :

```
aws lambda publish-layer-version \  
  --layer-name nova-custom-eval-layer \  
  --zip-file fileb://nova-custom-eval-layer.zip \  

```



```
--compatible-runtimes python3.12 python3.11 python3.10 python3.9
```

4. Ajoutez cette couche en tant que couche personnalisée à votre fonction Lambda, ainsi que la AWS couche requise : `AWSLambdaPowerToolsPythonV3-python312-arm64` (requis pour la pydantic dépendance).
5. Mettez à jour votre code Lambda à l'aide de l'exemple fourni, en modifiant le code selon les besoins. Cet exemple de code crée une fonction Lambda pour l'évaluation personnalisée de Nova avec des étapes de prétraitement et de post-traitement pour l'évaluation du modèle.

```
from nova_custom_evaluation_sdk.processors.decorators import preprocess,
    postprocess
from nova_custom_evaluation_sdk.lambda_handler import build_lambda_handler

@preprocess
def preprocessor(event: dict, context) -> dict:
    data = event.get('data', {})
    return {
        "statusCode": 200,
        "body": {
            "system": data.get("system"),
            "prompt": data.get("prompt", ""),
            "gold": data.get("gold", "")
        }
    }

@postprocess
def postprocessor(event: dict, context) -> dict:
    # data is already validated and extracted from event
    data = event.get('data', [])
    inference_output = data.get('inference_output', '')
    gold = data.get('gold', '')

    metrics = []
    inverted_accuracy = 0 if inference_output.lower() == gold.lower() else 1.0
    metrics.append({
        "metric": "inverted_accuracy_custom",
        "value": accuracy
    })

    # Add more metrics here

    return {
```

```

        "statusCode": 200,
        "body": metrics
    }

# Build Lambda handler
lambda_handler = build_lambda_handler(
    preprocessor=preprocessor,
    postprocessor=postprocessor
)

```

6. Accordez à Lambda l'accès à la tâche d'évaluation. Assurez-vous que le rôle d'exécution spécifié pour la tâche d'évaluation inclut une politique qui appelle votre fonction Lambda. Voici un exemple de politique.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LambdaAccess",
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": "arn:aws:lambda:us-east-1:111122223333:function:ExampleFunction",
      "Condition": {
        "StringLike": {
          "aws:PrincipalArn": "arn:aws:iam::111122223333:role/service-role/AmazonSageMaker-ExecutionRole-ARN"
        }
      }
    },
    {
      "Sid": "DenyNonAWSEventSourcesForLambda",
      "Effect": "Deny",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": "arn:aws:lambda:us-east-1:111122223333:function:ExampleFunction",

```

```
        "Condition": {
            "Null": {
                "lambda:EventSourceToken": false
            }
        }
    }
}
```

7. Passez en revue le schéma de charge utile Lambda. Le tableau suivant répertorie le schéma de demande et de réponse Lambda. Vous pouvez valider votre schéma à l'aide du SDK d'évaluation personnalisé Nova.

	Charge utile de la demande Lambda	Charge utile de réponse Lambda
Préproc seur	<pre>{ "process_type": "preproce ss", "data": { "system": "You are a helpful assistant", "prompt": "What is 2+2?", "gold": "4" } }</pre>	<pre>{ "statusCode": 200, "body": { "system": "You are a helpful assistant that can substitute * for addition", "prompt": "What is 2*2?", "gold": "4" } }</pre>
Post- proc esseur	<pre>{ "process_type": "postproc ess", "data": { "prompt": "What is 2+2?", "inference_output": "2+2=4", "gold": "4" } }</pre>	<pre>{ "statusCode": 200, "body": [{"metric": "accuracy", "value": 1.0}, {"metric": "f1_score", "value": 1.0}, {"metric": "exact_match", "value": 1}, {"metric": "length_ratio", "value": 0.8}] }</pre>

	Charge utile de la demande Lambda	Charge utile de réponse Lambda
		}

8. Modifiez le fichier de recette. Voici un exemple.

```
processor:
  lambda_arn: arn:aws:lambda:us-east-1:111122223333:function:name
  lambda_type: "custom_metrics"
  preprocessing:
    enabled: true
  postprocessing:
    enabled: true
  aggregation: average
```

- `lambda_arn`: Amazon Resource Name (ARN) de votre fonction Lambda qui gère le prétraitement et le post-traitement.
- `lambda_type`: « `custom_metrics` » ou « `rft` ».
- `preprocessing`: s'il faut activer les opérations de prétraitement personnalisées.
- `postprocessing`: s'il faut activer les opérations de post-traitement personnalisées.
- `aggregation`: fonction d'agrégation intégrée (options valides : `min`, `max`, `moyenne`, `somme`).

Limites

- `Bring your own metrics` ne s'applique qu'aux ensembles de données de saisie de texte.
- Les ensembles de données d'entrée multimodaux ne sont pas pris en charge.
- L'étape de prétraitement ne traite pas le champ de métadonnées.

Formules de point de référence Nova LLM-juge

Nova LLM-juge est une fonctionnalité d'évaluation des modèles qui vous permet de comparer la qualité des réponses d'un modèle aux réponses d'un modèle de référence à l'aide d'un jeu de données personnalisé. Il accepte un jeu de données contenant des invites, des réponses de base et des réponses de challenger, puis utilise un modèle Nova Judge pour fournir une métrique du taux de victoire basée sur la probabilité de [Bradley-Terry](#) par le biais de comparaisons par paires. Format de formule : `xxx_llm_judge_eval.yaml`.

Exigences relatives à le jeu de données Nova LLM

Format de fichier :

- Fichier `llm_judge.jsonl` unique contenant des exemples d'évaluation. Le nom du fichier doit être exactement `llm_judge.jsonl`.
- Vous devez télécharger votre ensemble de données sur un emplacement S3 auquel les postes de formation en SageMaker IA peuvent accéder.
- Le fichier doit respecter le format de schéma requis pour le jeu de données `llm_judge`.
- Le jeu de données d'entrée doit garantir que tous les enregistrements ont une longueur de contexte inférieure à 12 000.

Format de schéma : chaque ligne du fichier `.jsonl` doit être un objet JSON avec les champs suivants.

- Champs obligatoires.

`prompt` : chaîne contenant l'invite de réponse générée.

`response_A` : chaîne contenant la réponse de base.

`response_B` : chaîne contenant la réponse alternative comparée à la réponse de base.

Exemple d'entrée

```
{
  "prompt": "What is the most effective way to combat climate change?",
  "response_A": "The most effective way to combat climate change is through a combination of transitioning to renewable energy sources and implementing strict carbon pricing policies. This creates economic incentives for businesses to reduce emissions while promoting clean energy adoption.",
  "response_B": "We should focus on renewable energy. Solar and wind power are good. People should drive electric cars. Companies need to pollute less."
}
{
  "prompt": "Explain how a computer's CPU works",
  "response_A": "CPU is like brain of computer. It does math and makes computer work fast. Has lots of tiny parts inside.",
  "response_B": "A CPU (Central Processing Unit) functions through a fetch-execute cycle, where instructions are retrieved from memory, decoded, and executed through its"
```

```
arithmetic logic unit (ALU). It coordinates with cache memory and registers to process
data efficiently using binary operations."
}
{
  "prompt": "How does photosynthesis work?",
  "response_A": "Plants do photosynthesis to make food. They use sunlight and water. It
happens in leaves.",
  "response_B": "Photosynthesis is a complex biochemical process where plants convert
light energy into chemical energy. They utilize chlorophyll to absorb sunlight,
combining CO2 and water to produce glucose and oxygen through a series of chemical
reactions in chloroplasts."
}
```

Pour utiliser votre jeu de données personnalisé, modifiez votre formule d'évaluation avec les champs obligatoires suivants, sans modifier le contenu :

```
evaluation:
  task: llm_judge
  strategy: judge
  metric: all
```

Limites

- Un seul fichier `.jsonl` est autorisé par évaluation.
- Le fichier doit suivre strictement le schéma défini.
- Les modèles Nova Judge sont les mêmes quelles que soient les spécifications micro/lite/pro.
- Les modèles de juges personnalisés ne sont actuellement pas pris en charge.

Nova LLM en tant que juge pour les recettes de référence multimodales (image)

Nova LLM Judge pour multimodal (image), abréviation de Nova MM_LLM Judge, est une fonctionnalité d'évaluation de modèle qui vous permet de comparer la qualité des réponses d'un modèle à celles d'un modèle de référence à l'aide d'un ensemble de données personnalisé. Il accepte un ensemble de données contenant des instructions, des réponses de base et des réponses aux challengers, ainsi que des images sous forme de chaîne codée en Base64, puis utilise un modèle Nova Judge pour fournir une métrique de taux de victoire basée sur la probabilité de [Bradley-Terry](#) par le biais de comparaisons par paires. Format de formule : `xxx_mm_llm_judge_eval.yaml`.

Exigences relatives à le jeu de données Nova LLM

Format de fichier :

- Fichier `mm_llm_judge.jsonl` unique contenant des exemples d'évaluation. Le nom du fichier doit être exact `mm_llm_judge.jsonl`.
- Vous devez télécharger votre ensemble de données sur un emplacement S3 où les professionnels de formation à l' SageMaker IA peuvent y accéder.
- Le fichier doit respecter le format de schéma requis pour le jeu de données `mm_llm_judge`.
- L'ensemble de données en entrée doit garantir que tous les enregistrements ont une longueur de contexte inférieure à 12 k, à l'exception de l'attribut de l'image.

Format de schéma : chaque ligne du fichier `.jsonl` doit être un objet JSON avec les champs suivants.

- Champs obligatoires.

`prompt` : chaîne contenant l'invite de réponse générée.

`images`: Tableau contenant une liste d'objets avec des attributs de données (les valeurs sont des chaînes d'image codées en Base64).

`response_A` : chaîne contenant la réponse de base.

`response_B` : chaîne contenant la réponse alternative comparée à la réponse de base.

Exemple d'entrée

Pour des raisons de lisibilité, l'exemple suivant inclut de nouvelles lignes et une indentation, mais dans le jeu de données actuel, chaque enregistrement doit figurer sur une seule ligne.

```
{
  "prompt": "What is in the image?",
  "images": [
    {
      "data": "data:image/jpeg;base64,/9j/2wBDAAQDAwQDAwQEAwQFBAQFBgo..."
    }
  ],
  "response_A": "a dog.",
  "response_B": "a cat.",
}
```

```
"prompt": "How many animals are in each of the images?",
"images": [
  {
    "data": "data:image/jpeg;base64,/9j/2wBDAAQDAwQDAwQEAwQFBAQFBgo..."
  },
  {
    "data": "data:image/jpeg;base64,/DKEafe3gihn..."
  }
],
"response_A": "The first image contains one cat and the second image contains one dog",
"response_B": "The first image has one animal and the second has one animal"
}
```

Pour utiliser votre jeu de données personnalisé, modifiez votre formule d'évaluation avec les champs obligatoires suivants, sans modifier le contenu :

```
evaluation:
  task: mm_llm_judge
  strategy: judge
  metric: all
```

Limites

- Un seul fichier `.jsonl` est autorisé par évaluation.
- Le fichier doit suivre strictement le schéma défini.
- Les modèles Nova MM Judge ne prennent en charge que les références d'image.
- Les modèles Nova MM Judge sont les mêmes pour les spécifications d'Amazon Nova Micro, Amazon Nova Lite et Amazon Nova Pro.
- Les modèles de juges personnalisés ne sont actuellement pas pris en charge.
- L'URI de l'image Amazon S3 n'est pas prise en charge.
- L'ensemble de données en entrée doit garantir que tous les enregistrements ont une longueur de contexte inférieure à 12 k, à l'exception de l'attribut images.

Évaluation des points de contrôle du CPT (formation continue)

L'évaluation sur des modèles CPT (préentraînement en continu) peut s'avérer plus difficile que sur des modèles ayant subi un réglage précis supervisé (SFT), car les modèles CPT ne sont généralement pas capables de suivre les instructions. Au lieu de suivre les instructions, le modèle

CPT fonctionnera comme des modèles de complétion, ce qui signifie qu'il tentera uniquement de continuer le modèle fourni avec la séquence de jetons d'entrée. Compte tenu de cette limite, les ensembles de données d'évaluation classiques peuvent ne pas fonctionner correctement en raison du format « questions-réponses » des entrées. Au lieu de répondre à la question, le modèle essaiera simplement de continuer avec la même question. Cependant, en formatant les ensembles de données d'une manière spécifique pour proposer aux modèles un style de complétion, nous pouvons mieux comprendre les performances du modèle.

Suivez les étapes ci-dessous pour comprendre comment effectuer une évaluation sur un modèle préentraîné en continu à l'aide du flux de travail d'évaluation de Nova Forge.

Préparation et formatage des jeux de données

L'évaluation d'un modèle CPT tire parti du flux de travail [Bring Your Own Dataset](#) déjà fourni dans le cadre de l'expérience d'évaluation du modèle Nova Forge. Cependant, les requêtes au sein de l'ensemble de données doivent être formatées dans un format purement « complet », car les modèles CPT ne répondront pas aux questions standard de la même manière qu'un modèle SFT.

Une autre limitation fréquente importante des modèles qui ont subi uniquement le CPT est leur incapacité à générer des jetons STOP ou de fin de séquence. Cela signifie que le modèle continuera à générer des jetons jusqu'à ce qu'il soit arrêté de force (comme avec le paramètre `max_new_tokens`). Compte tenu de cette limite, la meilleure pratique consiste à évaluer les modèles à l'aide de réponses à jeton unique (telles que des choix multiples) afin de s'assurer que le modèle ne continue pas à générer des résultats indésirables inutiles après avoir été invités.

Par exemple, un ensemble de données d'évaluation classique (tel que MMLU, GPQA, MATH, etc.) peut demander au modèle une question telle que :

```
Early settlements and high population density along coastlines and rivers are  
best attributed to which of the following?
```

```
A: "Poor climate conditions"
```

```
B: "Limited forest cover"
```

```
C: "Cars"
```

```
D: "Access to trade routes"
```

```
(Expected answer is D.)
```

Cependant, un modèle CPT ne comprendrait pas comment répondre correctement à cette question en raison du manque de précision des instructions qui suivent. Par conséquent, les modèles CPT doivent être demandés dans un style de complétion, tel que :

Early settlements and high population density along coastlines and rivers are best attributed to which of the following?

A: Poor climate conditions

B: Limited forest cover

C: Cars

D: Access to trade routes

The correct answer to this question is option

(Expected answer is D.)

Après vérification des inférences, les probs de log de sortie générés par le modèle fourniront des informations indiquant si le modèle traitait correctement l'entrée et générait la bonne réponse. Il n'est pas garanti que le modèle produira toujours la réponse exacte (dans ce cas, la lettre D) attendue, mais elle doit être présente dans les logprobs si le modèle fonctionne correctement.

Autre exemple d'invite de style de complétion à choix non multiple :

The capital of France is

(Expected answer of Paris)

Nous nous attendrions à ce que le modèle produise une réponse « Paris » ou qu'il voie le jeton correspondant à « Paris » quelque part dans la sortie de logprobs.

Formatage des jeux

L'évaluation du CPT tire parti du flux de travail [Bring Your Own Dataset](#) existant. Les données doivent être formatées au format « réponse à la requête » dans un fichier JSONL séparé par de nouvelles lignes.

Exemple de jeu de données contenant 4 entrées :

```
{"query": "The capital of France is", "response": "Paris"}
{"query": "2 + 2 =", "response": "4"}
{"query": "The mitochondria is the powerhouse of the", "response": "cell"}
{"query": "What is the largest planet?\nA: Mars\nB: Jupiter\nC: Saturn\nD: Earth\n\nAnswer:", "response": "B"}
```

Chaque ligne doit contenir :

- query: le texte qui vous invite à terminer

- `response`: L'achèvement escompté (vérité sur le terrain)

Le modèle recevra une saisie de texte brut sans formatage du chat. Les modèles CPT ne sont généralement pas encore entraînés à utiliser des jetons spéciaux et ne répondent pas correctement aux modèles de chat. Ainsi, lorsque vous demandez au modèle, SEULE la chaîne fournie dans la requête sera envoyée au modèle (avec un [BOS] jeton supplémentaire ajouté automatiquement au modèle).

Configuration de la recette

Voici un exemple de recette configurée pour évaluer un modèle CPT :

```
run:
  name: cpt_eval_job
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: s3://bucket/path/to/cpt-checkpoint/

evaluation:
  task: gen_qa # Required for CPT - bring your own dataset
  strategy: gen_qa
  metric: all # Returns rouge1, rouge2, rougeL, exact_match, f1_score, bleu

inference:
  checkpoint_is_instruction_tuned: "false" # Required for CPT checkpoints
  top_logprobs: 5 # Change to desired amount of logprobs to calculate
  max_new_tokens: 1 # Keep low for completion tasks
  temperature: 0.0
```

Principaux changements pour l'évaluation du CPT :

- `checkpoint_is_instruction_tuned`: "false"

Il s'agit d'un nouveau paramètre ajouté spécifiquement pour prendre en charge les opérations d'évaluation sur les points de contrôle du CPT. La définition de `checkpoint_is_instruction_tuned` sur `false` désactivera le modèle de chat Amazon Nova par défaut qui enveloppe normalement l'invite de saisie.

- `top_logprobs`: 5

Les probabilités de journalisation (logprobs) révèlent la répartition de la confiance du modèle entre les jetons suivants possibles, ce qui vous aide à évaluer si le modèle a atteint les objectifs attendus

lors de la pré-formation. Généralement, si le modèle fonctionne comme prévu, nous devrions voir la réponse attendue (c'est-à-dire « A », « B », etc.) sous la forme d'un jeton de sortie généré ou d'un jeton dans les logprobs.

- `max_new_tokens`: 1

Les modèles CPT n'ont généralement pas encore été entraînés à générer des jetons spéciaux « d'arrêt » ou de « fin de séquence » pour indiquer quand arrêter l'inférence. Cela signifie que le modèle continuera généralement à générer de nouveaux jetons jusqu'à ce que la longueur maximale des jetons soit atteinte, ce qui entraîne des inférences inutiles. En général, le moyen le plus efficace de demander le modèle est de limiter les `max_new_tokens` à 1 et de fournir une invite permettant d'évaluer le modèle sur une seule réponse (comme une question à choix multiples). La définition de `max_new_tokens` sur 1 garantit qu'aucun jeton indésirable supplémentaire ne sera généré.

Paramètres clés

- `checkpoint_is_instruction_tuned` : doit être défini sur (ou en tant que booléen "`false`") pour désactiver les modèles de chat `false`
- `top_logprobs` : 5, recommandé pour voir comment le modèle apprend pendant le CPT
- `tâche` : Doit être `gen_qa` - Les modèles CPT ne peuvent pas utiliser de benchmarks suivant les instructions tels que MMLU ou MATH
- `stratégie` : Doit être `gen_qa`
- `max_new_tokens` : Il est recommandé de rester faible (1-5) car les modèles CPT effectuent la complétion et non la génération

Exécution de tâches d'entraînement d'évaluation

Commencez une tâche de formation à l'aide de l'exemple de bloc-notes suivant. Veuillez vous référer au bloc-notes ci-dessous à titre d'exemple pour exécuter la tâche de formation à l'évaluation. Pour plus d'informations, voir [Utiliser un estimateur SageMaker basé sur l'IA pour exécuter une tâche de formation](#).

Tableaux de référence

Avant d'exécuter le bloc-notes, reportez-vous aux tableaux de référence suivants pour sélectionner l'URI de l'image et les configurations d'instance.

Sélection de l'URI d'image

Formule	URI de l'image
URI de l'image d'évaluation	708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-TJ-Eval-V2-latest

Sélection du type et du nombre d'instances

Modèle	Type de tâche	Type d'instance	Nombre d'instances recommandées	Nombre d'instances autorisées
Amazon Nova Micro	Évaluation (SFT/DPO)	g5.12xlarge	1	1 à 16
Amazon Nova Lite	Évaluation (SFT/DPO)	g5.12xlarge	1	1 à 16
Amazon Nova Pro	Évaluation (SFT/DPO)	p5.48xlarge	1	1 à 16

Exemple de bloc-notes

L'exemple de bloc-notes ci-après montre comment exécuter une tâche d'entraînement d'évaluation.

```
# install python SDK

# Do not use sagemaker v3, as sagemaker v3 introduced breaking changes

!pip install sagemaker==2.254.1

import os
import sagemaker,boto3
from sagemaker.inputs import TrainingInput
from sagemaker.pytorch import PyTorch

sagemaker_session = sagemaker.Session()
role = sagemaker.get_execution_role()
```

```

# Download recipe from https://github.com/aws/sagemaker-hyperpod-recipes/tree/main/
recipes_collection/recipes/evaluation/nova to local
# Assume the file name be `recipe.yaml`

# Populate parameters
# input_s3_uri = "s3://<path>/input/" # (Optional) Only used for multi-modal dataset or
bring your own dataset s3 location
output_s3_uri= "s3://<path>/output/" # Output data s3 location, a zip containing
metrics json and tensorboard metrics files will be stored to this location
instance_type = "instance_type" # ml.g5.16xlarge as example
instance_count = 1 # The number of instances for inference (set instance_count > 1 for
multi-node inference to accelerate evaluation)
job_name = "your job name"
recipe_path = "recipe path" # ./recipe.yaml as example
image_uri = "708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-TJ-
Eval-V2-latest" # Do not change
output_kms_key = "<KMS key arn to encrypt trained model in Amazon-owned S3 bucket>" #
optional, leave blank for Amazon managed encryption

# (Optional) To bring your own dataset and LLM judge for evaluation
# evalInput = TrainingInput(
# s3_data=input_s3_uri,
# distribution='FullyReplicated',
# s3_data_type='S3Prefix'
#)

estimator = PyTorch(
    output_path=output_s3_uri,
    base_job_name=job_name,
    role=role,
    instance_type=instance_type,
    instance_count=instance_count,
    training_recipe=recipe_path,
    sagemaker_session=sagemaker_session,
    image_uri=image_uri,
    output_kms_key=output_kms_key
)
estimator.fit()

# If input dataset exist, pass in inputs
# estimator.fit(inputs={"train": evalInput})

```

Évaluation et analyse des résultats d'évaluation

Une fois votre tâche d'évaluation terminée avec succès, vous pouvez évaluer et analyser les résultats en suivant les étapes suivantes.

Pour évaluer et analyser les résultats, procédez comme suit.

1. Comprenez la structure de l'emplacement de sortie. Les résultats sont stockés dans l'emplacement de sortie Amazon S3 que vous avez spécifié sous forme de fichier compressé :

```
s3://your-bucket/output/benchmark-name/
### job_name/
    ### output/
        ### output.tar.gz
```

2. Téléchargez le fichier `output.tar.gz` depuis votre compartiment. Extrayez le contenu pour le révéler.

```
run_name/
### eval_results/
|   ### results_[timestamp].json
#   ### inference_output.jsonl (only present for gen_qa)
|   ### details/
|       ### model/
|           ### <execution-date-time>/
|               ###details_<task_name>_#_<datetime>.parquet
### tensorboard_results/
    ### eval/
        ### events.out.tfevents.[timestamp]
```

- `results_[timestamp].json` : fichier JSON des métriques de sortie
 - `details_<task_name>_#_<datetime>.parquet` : fichier de sortie d'inférence (sauf pour `strong_reject`)
 - `events.out.tfevents.[timestamp]` - fichier TensorBoard de sortie
 - `inference_output.jsonl` : fichier de sortie d'inférence nettoyé (uniquement pour les tâches `gen_qa`)
3. Afficher les résultats dans TensorBoard. Pour visualiser vos métriques d'évaluation :
 1. Chargez le dossier extrait dans un compartiment S3

2. Naviguez vers l' SageMaker IA TensorBoard
 3. Sélectionnez vos « dossiers S3 »
 4. Ajoutez le chemin du dossier S3
 5. Patientez jusqu'à ce que la synchronisation soit terminée
4. Analysez les sorties d'inférence. Toutes les tâches d'évaluation, à l'exception de `llm_judge` et `strong_reject`, comporteront les champs d'analyse suivants dans le résultat d'inférence.
- `full_prompt`- L'invite utilisateur complète envoyée au modèle utilisé pour la tâche d'évaluation.
 - `gold`- Le champ qui contient la ou les bonnes réponses spécifiées par l'ensemble de données.
 - `metrics`- Le champ qui contient les métriques évaluées sur la base de l'inférence individuelle. Les valeurs nécessitant une agrégation n'auraient pas de valeur sur les sorties d'inférence individuelles.
 - `predictions`- Le champ qui contient la liste des résultats du modèle pour l'invite donnée.
 - `pred_logits`- Le champ qui contient les jetons de sortie considérés et les probabilités de journalisation de chaque jeton de sortie renvoyé dans le contenu du message.

En consultant ces champs, vous pouvez déterminer la cause des différences entre les métriques et comprendre le comportement des modèles personnalisés.

Pour `llm_judge`, le fichier de sortie d'inférence contient les champs suivants sous le champ des métriques par paire d'évaluations.

- `forward_output` : préférences brutes de juge lors de l'évaluation dans l'ordre (`response_A`, `response_B`).
- `backward_output` : préférences brutes de juge lors de l'évaluation dans l'ordre inverse (`response_B`, `response_A`).
- `Pairwise metrics` : métriques calculés par paire d'évaluations avant et arrière, y compris `a_scores`, `b_scores`, `ties`, `inference-score` et `score`.

 Note

Les métriques agrégées telles que `winrate` ne sont disponibles que dans les fichiers de résultats récapitulatifs, et non par jugement individuel.

Pour `gen_qa`, le fichier `inference_output.jsonl` contient les champs suivants pour chaque objet JSON :

- `invite` : le message final soumis au modèle
- `inférence` : la sortie d'inférence brute du modèle
- `gold` - La réponse cible de l'ensemble de données en entrée
- `metadata` - La chaîne de métadonnées de l'ensemble de données en entrée, si elle est fournie

Bonnes pratiques d'évaluation et dépannage

Bonnes pratiques

Voici une liste de bonnes pratiques pour le processus d'évaluation.

- Organisez vos chemins de sortie par modèle et par type de référence.
- Maintenez des conventions de dénomination cohérentes pour faciliter le suivi.
- Enregistrez les résultats extraits dans un emplacement sécurisé.
- Surveillez l'état de TensorBoard synchronisation pour un chargement de données réussi.

Résolution des problèmes

Vous pouvez utiliser un groupe de CloudWatch journaux `/aws/sagemaker/TrainingJobs` pour former les journaux d'erreurs des tâches.

Défaillance du cœur du moteur

Problème :

Si vous observez :

```
RuntimeError: Engine core initialization failed.
```

Cause :

Bien qu'il s'agisse d'une erreur générale pouvant avoir plusieurs causes, elle se produit généralement lorsqu'il existe une incompatibilité entre le point de contrôle du modèle que vous essayez de charger

et le type de modèle spécifié. Par exemple, vous souhaitez évaluer un point de contrôle affiné du modèle Nova 2.0 lite, mais le type de modèle que vous fournissez est de type 1.0. Par exemple `amazon.nova-micro-v1:0:128k`

Le mappage correct doit être

```
model_type: amazon.nova-2-lite-v1:0:256k
model_name_or_path: nova-lite-2/prod # or s3://escrow_bucket/model_location
```

Prévention :

Vérifiez qu'`model_name_or_path` est mappé vers la droite `model_type` avant de soumettre le travail d'évaluation.

Sous-tâches disponibles

La liste suivante répertorie les sous-tâches disponibles pour l'évaluation de modèles dans plusieurs domaines, notamment MMLU (Massive Multitask Language Understanding), BBH (Big Bench Hard), MATH et MMMU (Massive Multi-discipline Multimodal Understanding). Ces sous-tâches vous permettent d'évaluer les performances de votre modèle par rapport à des capacités et à des domaines de connaissances spécifiques.

MMLU

```
MMLU_SUBTASKS = [
    "abstract_algebra",
    "anatomy",
    "astronomy",
    "business_ethics",
    "clinical_knowledge",
    "college_biology",
    "college_chemistry",
    "college_computer_science",
    "college_mathematics",
    "college_medicine",
    "college_physics",
    "computer_security",
    "conceptual_physics",
    "econometrics",
    "electrical_engineering",
    "elementary_mathematics",
    "formal_logic",
```

```
"global_facts",  
"high_school_biology",  
"high_school_chemistry",  
"high_school_computer_science",  
"high_school_european_history",  
"high_school_geography",  
"high_school_government_and_politics",  
"high_school_macroeconomics",  
"high_school_mathematics",  
"high_school_microeconomics",  
"high_school_physics",  
"high_school_psychology",  
"high_school_statistics",  
"high_school_us_history",  
"high_school_world_history",  
"human_aging",  
"human_sexuality",  
"international_law",  
"jurisprudence",  
"logical_fallacies",  
"machine_learning",  
"management",  
"marketing",  
"medical_genetics",  
"miscellaneous",  
"moral_disputes",  
"moral_scenarios",  
"nutrition",  
"philosophy",  
"prehistory",  
"professional_accounting",  
"professional_law",  
"professional_medicine",  
"professional_psychology",  
"public_relations",  
"security_studies",  
"sociology",  
"us_foreign_policy",  
"virology",  
"world_religions"
```

```
]
```

BBH

```
BBH_SUBTASKS = [  
    "boolean_expressions",  
    "causal_judgement",  
    "date_understanding",  
    "disambiguation_qa",  
    "dyck_languages",  
    "formal_fallacies",  
    "geometric_shapes",  
    "hyperbaton",  
    "logical_deduction_five_objects",  
    "logical_deduction_seven_objects",  
    "logical_deduction_three_objects",  
    "movie_recommendation",  
    "multistep_arithmetic_two",  
    "navigate",  
    "object_counting",  
    "penguins_in_a_table",  
    "reasoning_about_colored_objects",  
    "ruin_names",  
    "salient_translation_error_detection",  
    "snarks",  
    "sports_understanding",  
    "temporal_sequences",  
    "tracking_shuffled_objects_five_objects",  
    "tracking_shuffled_objects_seven_objects",  
    "tracking_shuffled_objects_three_objects",  
    "web_of_lies",  
    "word_sorting"  
]
```

Math

```
MATH_SUBTASKS = [  
    "algebra",  
    "counting_and_probability",  
    "geometry",  
    "intermediate_algebra",  
    "number_theory",  
    "prealgebra",  
    "precalculus",  
]
```

MMMU

```
MATH_SUBTASKS = [  
    "Accounting",  
    "Agriculture",  
    "Architecture_and_Engineering",  
    "Art",  
    "Art_Theory",  
    "Basic_Medical_Science",  
    "Biology",  
    "Chemistry",  
    "Clinical_Medicine",  
    "Computer_Science",  
    "Design",  
    "Diagnostics_and_Laboratory_Medicine",  
    "Economics",  
    "Electronics",  
    "Energy_and_Power",  
    "Finance",  
    "Geography",  
    "History",  
    "Literature",  
    "Manage",  
    "Marketing",  
    "Materials",  
    "Math",  
    "Mechanical_Engineering",  
    "Music",  
    "Pharmacy",  
    "Physics",  
    "Psychology",  
    "Public_Health",  
    "Sociology",
```

Évaluez vos modèles Nova personnalisés à l'aide de diverses méthodes et mesures d'évaluation.

Dans cette section :

- [Évaluation du modèle de raisonnement](#)
- [Évaluation RFT](#)
- [Mettre en œuvre des fonctions de récompense](#)

Évaluation du modèle de raisonnement

Présentation de

Le support des modèles de raisonnement permet une évaluation à l'aide de modèles Nova capables de raisonner qui effectuent un raisonnement interne explicite avant de générer des réponses finales. Cette fonctionnalité utilise le contrôle au niveau de l'API via le `reasoning_effort` paramètre pour activer ou désactiver dynamiquement la fonctionnalité de raisonnement, améliorant potentiellement la qualité des réponses pour les tâches analytiques complexes.

Modèles pris en charge

- `amazon.nova-2-lite-v 1:0:256 k`

Configuration de la recette

Activez le raisonnement en ajoutant le `reasoning_effort` paramètre à la `inference` section de votre recette :

```
run:
  name: reasoning-eval-job-name                # [MODIFIABLE] Unique
  identifier for your evaluation job
  model_type: amazon.nova-2-lite-v1:0:256k      # [FIXED] Must be a reasoning-
supported model
  model_name_or_path: nova-lite-2/prod          # [FIXED] Path to model
  checkpoint or identifier
  replicas: 1                                   # [MODIFIABLE] Number of
replicas for SageMaker Training job
  data_s3_path: ""                             # [MODIFIABLE] Leave empty for
SageMaker Training job; optional for SageMaker HyperPod job
  output_s3_path: ""                           # [MODIFIABLE] Output path for
SageMaker HyperPod job (not compatible with SageMaker Training jobs)

evaluation:
  task: mmlu                                    # [MODIFIABLE] Evaluation task

  strategy: zs_cot                             # [MODIFIABLE] Evaluation
strategy
  metric: accuracy                             # [MODIFIABLE] Metric
calculation method

inference:
```

```

reasoning_effort: high # [MODIFIABLE] Enables
reasoning mode; options: low/high or null to disable
max_new_tokens: 32768 # [MODIFIABLE] Maximum tokens
to generate, recommended value when reasoning_effort set to high
top_k: -1 # [MODIFIABLE] Top-k sampling
parameter
top_p: 1.0 # [MODIFIABLE] Nucleus
sampling parameter
temperature: 0 # [MODIFIABLE] Sampling
temperature (0 = deterministic)

```

Utilisation du paramètre reasoning_effort

Le `reasoning_effort` paramètre contrôle le comportement de raisonnement des modèles capables de raisonner.

Conditions préalables

- Compatibilité des modèles : définit `reasoning_effort` uniquement lorsqu'un modèle capable de raisonner est spécifié (actuellement) `amazon.nova-2-lite-v1:0:256k`
- Gestion des erreurs — L'utilisation `reasoning_effort` avec des modèles non pris en charge échouera avec `ConfigValidationError: "Reasoning mode is enabled but model '{model_type}' does not support reasoning. Please use a reasoning-capable model or disable reasoning mode."`

Options disponibles

Option	Comportement	Limite de jetons	Cas d'utilisation
null (par défaut)	Désactive le mode raisonnement	N/A	Évaluation standard sans frais de raisonnement
bas	Permet de raisonner avec des contraintes	4 000 jetons pour le raisonnement interne	Scénarios nécessitant un raisonnement concis ; optimisations en termes de rapidité et de coût
haut	Permet de raisonner sans contraintes	Aucune limite symbolique au raisonnement interne	Problèmes complexes nécessitant une analyse

Option	Comportement	Limite de jetons	Cas d'utilisation
			et un step-by-step raisonnement approfondis
Méthode d'entraînement	Options disponibles	Comment configurer	
SFT (réglage fin supervisé)	Élevé ou désactivé uniquement	Utilisez <code>reasoning_enabled</code> : <code>true</code> (high) ou <code>false</code> (off)	
RFT (réglage fin du renforcement)	Faible, haut ou éteint	Utilisez <code>reasoning_effort</code> : <code>low</code> ou <code>reasoning_effort</code> : <code>high</code> . Omettez le champ à désactiver.	
Évaluation	Faible, haut ou éteint	Utilisez <code>reasoning_effort</code> : <code>low</code> ou <code>reasoning_effort</code> : <code>high</code> . Utilisez <code>null</code> pour désactiver.	

Quand activer le raisonnement

Utiliser le mode raisonnement (**low** ou **high**) pour

- Tâches complexes de résolution de problèmes (mathématiques, énigmes logiques, codage)
- Questions analytiques en plusieurs étapes nécessitant un raisonnement intermédiaire
- Tâches pour lesquelles des explications ou des step-by-step réflexions détaillées améliorent la précision
- Scénarios dans lesquels la qualité de réponse est privilégiée par rapport à la vitesse

Utiliser le mode non raisonné (**null** ou omettre un paramètre) pour

- Questions et réponses simples ou questions factuelles
- Tâches d'écriture créative
- Lorsque des temps de réponse plus rapides sont essentiels
- Analyse comparative des performances où les frais de raisonnement doivent être exclus
- Optimisation des coûts lorsque le raisonnement n'améliore pas les performances des tâches

Résolution des problèmes

Erreur : « Le mode raisonnement est activé mais le modèle ne prend pas en charge le raisonnement »

Cause : le `reasoning_effort` paramètre est défini sur une valeur non nulle, mais la valeur spécifiée `model_type` ne permet pas de raisonner.

Résolution :

- Vérifiez que le type de votre modèle est `amazon.nova-2-lite-v1:0:256k`
- Si vous utilisez un modèle différent, passez à un modèle capable de raisonner ou supprimez le `reasoning_effort` paramètre de votre recette

Évaluation RFT

Qu'est-ce que l'évaluation RFT ?

L'évaluation RFT vous permet d'évaluer les performances de votre modèle à l'aide de fonctions de récompense personnalisées avant, pendant ou après l'entraînement par renforcement. Contrairement aux évaluations standard qui utilisent des métriques prédéfinies, RFT Evaluation vous permet de définir vos propres critères de réussite grâce à une fonction Lambda qui note les résultats du modèle en fonction de vos besoins spécifiques.

Pourquoi évaluer avec RFT ?

L'évaluation est cruciale pour déterminer si le processus de réglage précis des RL a :

- Alignement amélioré du modèle avec votre cas d'utilisation spécifique et vos valeurs humaines
- Fonctionnalités du modèle maintenues ou améliorées pour les tâches clés
- Effets secondaires imprévus évités tels qu'une baisse de la factualité, une verbosité accrue ou une dégradation des performances sur d'autres tâches
- Répond à vos critères de réussite personnalisés tels que définis par votre fonction de récompense

Quand utiliser l'évaluation RFT

Utilisez l'évaluation RFT dans les scénarios suivants :

- Avant la formation RFT : établissez des mesures de référence sur votre jeu de données d'évaluation

- Pendant la formation RFT : surveillez la progression de la formation avec des points de contrôle intermédiaires
- Après la formation RFT : vérifiez que le modèle final répond à vos exigences
- Comparaison de modèles : évaluez plusieurs versions de modèles à l'aide de critères de récompense cohérents

Note

Utilisez l'évaluation RFT lorsque vous avez besoin de mesures personnalisées et spécifiques à un domaine. Pour une évaluation à usage général (précision, perplexité, BLEU), utilisez des méthodes d'évaluation standard.

Exigences de format de données

Structure des données d'entrée

Les données d'entrée d'évaluation RFT doivent suivre le format OpenAI Reinforcement Fine-Tuning. Chaque exemple est un objet JSON contenant :

- **messages**— Gamme de tournures de conversation `system` et de rôles `user`
- **reference_answer**— Résultats attendus ou données de base utilisées par votre fonction de récompense pour marquer des points

Exemple de format de données

```
{
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": "Solve for x. Return only JSON like {\"x\": <number>}. Equation: 2x + 5 = 13"
        }
      ]
    }
  ],
  "reference_answer": "x = 4"
}
```

```
"reference_answer": {
  "x": 4
}
}
```

Limitations actuelles

- Texte uniquement : aucune entrée multimodale (images, audio, vidéo) n'est prise en charge
- Conversations à tour unique : ne prend en charge que les messages d'un seul utilisateur (pas de dialogues à plusieurs tours)
- Format JSON : les données d'entrée doivent être au format JSONL (un objet JSON par ligne)
- Sorties du modèle : L'évaluation est effectuée sur les complétions générées à partir du modèle spécifié

Préparation de votre recette d'évaluation

Exemple de bloc-notes

Pour un exemple complet, voir [Carnets d'évaluation](#).

Exemple de configuration de recette

```
run:
  name: nova-lite-rft-eval-job
  model_type: amazon.nova-lite-v1:0:300k
  model_name_or_path: s3://escrow_bucket/model_location # [MODIFIABLE] S3 path to your
  model or model identifier
  replicas: 1 # [MODIFIABLE] For SageMaker Training jobs only; fixed for HyperPod jobs

  data_s3_path: "" # [REQUIRED FOR HYPERPOD] Leave empty for SageMaker Training jobs
  and use TrainingInput in sagemaker python SDK
  output_s3_path: "" # [REQUIRED] Output artifact S3 path for evaluation results

evaluation:
  task: rft_eval # [FIXED] Do not modify
  strategy: rft_eval # [FIXED] Do not modify
  metric: all # [FIXED] Do not modify

# Inference Configuration
inference:
  max_new_tokens: 8192 # [MODIFIABLE] Maximum tokens to generate
```

```

top_k: -1 # [MODIFIABLE] Top-k sampling parameter
top_p: 1.0 # [MODIFIABLE] Nucleus sampling parameter
temperature: 0 # [MODIFIABLE] Sampling temperature (0 = deterministic)
top_logprobs: 0 # [MODIFIABLE] Set between 1-20 to enable logprobs output

# =====
# Bring Your Own Reinforcement Learning Environment
# =====
rl_env:
  reward_lambda_arn: arn:aws:lambda:<region>:<account_id>:function:<reward-function-
name>

```

Fonctions de récompense prédéfinies

Deux fonctions de récompense prédéfinies (`prime_code` et `prime_math`) sont disponibles sous forme de couche Lambda pour une intégration facile à vos fonctions Lambda RFT.

Présentation de

Ces fonctions prédéfinies fournissent des capacités out-of-the-box d'évaluation pour :

- `prime_code`— Génération de code et évaluation de l'exactitude
- `prime_math`— Évaluation du raisonnement mathématique et de la résolution de problèmes

Configuration rapide

1. [Téléchargez la couche Lambda depuis les nova-custom-eval-sdk versions.](#)
2. Publiez une couche Lambda à l'aide de l'interface de ligne de commande (AWS CLI) :

```

aws lambda publish-layer-version \
  --layer-name preset-function-layer \
  --description "Preset reward function layer with dependencies" \
  --zip-file fileb://universal_reward_layer.zip \
  --compatible-runtimes python3.9 python3.10 python3.11 python3.12 \
  --compatible-architectures x86_64 arm64

```

3. Ajoutez la couche à votre fonction Lambda dans AWS Management Console (préset-function-layers sélectionnez la couche personnalisée et ajoutez AWSSDKPandas-Python 312 pour les dépendances numpy).
4. Importez et utilisez dans votre code Lambda :

```
from prime_code import compute_score # For code evaluation
from prime_math import compute_score # For math evaluation
```

fonction prime_code

Évalue les tâches de génération de code Python en exécutant le code par rapport à des cas de test et en mesurant l'exactitude.

Exemple de format de jeu de données en entrée

```
{"messages":[{"role":"user","content":"Write a function that returns the sum of
two numbers."}], "reference_answer":{"inputs":["3\n5", "10\n-2", "0\n0"], "outputs":
["8", "8", "0"]}}
{"messages":[{"role":"user","content":"Write a function to check if a number
is even."}], "reference_answer":{"inputs":["4", "7", "0", "-2"], "outputs":
["True", "False", "True", "True"]}}
```

Fonctions principales

- Extraction automatique du code à partir de blocs de code Markdown
- Détection des fonctions et tests basés sur les appels
- Exécution des scénarios de test avec protection contre le délai d'expiration
- Validation de syntaxe et contrôles de compilation
- Signalement détaillé des erreurs avec retraçage

fonction prime_math

Évalue les capacités de raisonnement mathématique et de résolution de problèmes à l'aide d'un support mathématique symbolique.

Format d'entrée

```
{"messages":[{"role":"user","content":"What is the derivative of x^2 +
3x?."}], "reference_answer":"2*x + 3"}
```

Fonctions principales

- Évaluation mathématique symbolique à l'aide de SymPy
- Formats de réponse multiples (LaTeX, texte brut, symbolique)
- Vérification de l'équivalence mathématique
- Normalisation et simplification des expressions

Exigences de format de données

Pour l'évaluation du code

- Entrées : tableau d'arguments de fonction (types appropriés : entiers, chaînes, etc.)
- Sorties : tableau des valeurs de retour attendues (types appropriés : booléens, nombres, etc.)
- Code : Doit être en Python avec des définitions de fonctions claires

Pour l'évaluation des mathématiques

- Réponse de référence : expression mathématique ou valeur numérique
- Réponse : Il peut s'agir d'un LaTeX, d'un texte brut ou d'une notation symbolique
- Équivalence : vérifiée symboliquement, pas seulement pour la correspondance des chaînes

Bonnes pratiques

- Utilisez les types de données appropriés dans les cas de test (entiers contre chaînes, booléens contre « Vrai »)
- Fournir des signatures de fonction claires en cas de problèmes de code
- Inclure les cas extrêmes dans les entrées de test (zéro, nombres négatifs, entrées vides)
- Formater les expressions mathématiques de manière cohérente dans les réponses de référence
- Testez votre fonction de récompense avec des exemples de données avant le déploiement

Gestion des erreurs

Les deux fonctions incluent une gestion robuste des erreurs pour :

- Erreurs de compilation dans le code généré
- Exceptions d'exécution pendant l'exécution
- Données d'entrée mal formées

- Scénarios de temporisation pour des boucles infinies
- Expressions mathématiques non valides

Création de votre fonction de récompense

Exigences relatives à l'ARN Lambda

Votre ARN Lambda doit respecter le format suivant :

```
"arn:aws:lambda:*:*:function:*SageMaker*"
```

Si le Lambda ne possède pas ce schéma de dénomination, la tâche échouera avec cette erreur :

```
[ERROR] Unexpected error: lambda_arn must contain one of: ['SageMaker', 'sagemaker', 'Sagemaker'] when running on SMHP platform (Key: lambda_arn)
```

Format de demande Lambda

Votre fonction Lambda reçoit des données au format suivant :

```
[
  {
    "id": "sample-001",
    "messages": [
      {
        "role": "user",
        "content": [
          {
            "type": "text",
            "text": "Do you have a dedicated security team?"
          }
        ]
      },
      {
        "role": "nova_assistant",
        "content": [
          {
            "type": "text",
            "text": "As an AI developed by Company, I don't have a dedicated security team..."
          }
        ]
      }
    ]
  }
]
```

```
    ]
  }
],
"reference_answer": {
  "compliant": "No",
  "explanation": "As an AI developed by Company, I do not have a traditional
security team..."
}
}
]
```

Note

La structure du message inclut le contenu tableau imbriqué, correspondant au format des données d'entrée. Le dernier message avec un rôle `nova_assistant` contient la réponse générée par le modèle.

Format de réponse Lambda

Votre fonction Lambda doit renvoyer des données au format suivant :

```
[
  {
    "id": "sample-001",
    "aggregate_reward_score": 0.75,
    "metrics_list": [
      {
        "name": "accuracy",
        "value": 0.85,
        "type": "Metric"
      },
      {
        "name": "fluency",
        "value": 0.90,
        "type": "Reward"
      }
    ]
  }
]
```

Champs de réponse

- `id`— Doit correspondre à l'identifiant de l'échantillon saisi
- `aggregate_reward_score`— Note globale (généralement de 0,0 à 1,0)
- `metrics_list`— Ensemble de mesures individuelles avec :
 - `name`— Identifiant métrique (par exemple, « précision », « fluidité »)
 - `value`— Score métrique (généralement de 0,0 à 1,0)
 - `type`— Soit « métrique » (pour les rapports), soit « récompense » (utilisée dans le cadre de la formation)

Autorisations IAM

Autorisations requises

Votre rôle SageMaker d'exécution doit être autorisé à appeler votre fonction Lambda. Ajoutez cette politique à votre rôle SageMaker d'exécution :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": "arn:aws:lambda:region:account-id:function:function-name"
    }
  ]
}
```

Rôle d'exécution Lambda

Le rôle d'exécution de votre fonction Lambda nécessite des autorisations d'exécution Lambda de base :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:*"
}
]
```

Si votre fonction Lambda accède à d'autres AWS services (par exemple, S3 pour les données de référence, DynamoDB pour la journalisation), ajoutez ces autorisations au rôle d'exécution Lambda.

Exécution du travail d'évaluation

1. Préparez vos données — Formatez vos données d'évaluation conformément aux exigences de format de données et téléchargez votre fichier JSONL sur S3 : `s3://your-bucket/eval-data/eval_data.jsonl`
2. Configurer votre recette — Mettez à jour l'exemple de recette avec votre configuration :
 - Réglez `model_name_or_path` selon l'emplacement de votre modèle
 - Réglez `lambda_arn` sur l'ARN de votre fonction de récompense
 - Réglez `output_s3_path` sur l'emplacement de sortie souhaité
 - Ajustez `inference` les paramètres selon les besoins

Enregistrez la recette sous `rft_eval_recipe.yaml`

3. Exécuter l'évaluation — Exécute le travail d'évaluation à l'aide du bloc-notes fourni : [Carnets d'évaluation](#)
4. Suivez les progrès — Suivez votre travail d'évaluation grâce à :
 - SageMaker Console : vérifier l'état des tâches et les journaux
 - CloudWatch Journaux : Afficher les journaux d'exécution détaillés
 - Lambda Logs : résolution des problèmes liés à la fonction de récompense

Comprendre les résultats de l'évaluation

Format de sortie

La tâche d'évaluation produit les résultats vers l'emplacement S3 que vous avez spécifié au format JSONL. Chaque ligne contient les résultats de l'évaluation pour un échantillon :

```
{
  "id": "sample-001",
  "aggregate_reward_score": 0.75,
  "metrics_list": [
    {
      "name": "accuracy",
      "value": 0.85,
      "type": "Metric"
    },
    {
      "name": "fluency",
      "value": 0.90,
      "type": "Reward"
    }
  ]
}
```

Note

La sortie du Job d'évaluation RFT est identique au format Lambda Response. Le service d'évaluation passe en revue la réponse de votre fonction Lambda sans modification, garantissant ainsi la cohérence entre vos calculs de récompenses et les résultats finaux.

Interprétation des résultats

Score de récompense agrégé

- Plage : généralement de 0,0 (pire) à 1,0 (meilleur), mais cela dépend de votre implémentation
- Objectif : numéro unique résumant les performances globales
- Utilisation : comparer les modèles, suivre les améliorations au fil de la formation

Métriques individuelles

- Type de métrique : métriques informatives pour l'analyse
- Type de récompense : mesures utilisées lors de l'entraînement RFT
- Interprétation : des valeurs plus élevées indiquent généralement de meilleures performances (sauf si vous concevez des métriques inverses)

Critères de performance

Ce qui constitue une « bonne » performance dépend de votre cas d'utilisation :

Fourchette de scores	Interprétation	Action
0,8 - 1,0	Excellente	Modèle prêt pour le déploiement
0,6 - 0,8	Bon	Des améliorations mineures peuvent être bénéfiques
0,4 - 0,6	équitable	Une amélioration significative est nécessaire
0,0 - 0,4	Pauvre	Passez en revue les données d'entraînement et la fonction de récompense

Important

Il s'agit de directives générales. Définissez vos propres seuils en fonction des exigences commerciales, des performances du modèle de référence, des contraintes spécifiques au domaine et de l'analyse coûts-avantages de la formation continue.

Résolution des problèmes

Problèmes courants

Problème	Cause	Solution
délai d'expiration Lambda	Calcul complexe des récompenses	Augmenter le délai Lambda ou optimiser la fonction
Autorisation refusée	Autorisations IAM manquantes	Vérifier que SageMaker le rôle peut invoquer Lambda

Problème	Cause	Solution
Scores incohérents	Fonction de récompense non déterministe	Utilisez des graines fixes ou une logique déterministe
Résultats manquants	Erreurs Lambda non détectées	Ajoutez une gestion complète des erreurs dans Lambda

Liste de contrôle du débogage

- Vérifiez que les données d'entrée suivent le format correct avec des tableaux de contenu imbriqués
- Vérifiez que l'ARN Lambda est correct et que la fonction est déployée
- Vérifiez les autorisations IAM pour SageMaker → Invocation Lambda
- Vérifiez CloudWatch les journaux pour détecter les erreurs Lambda
- Valider que la réponse Lambda correspond au format attendu

Bonnes pratiques

- Commencez simplement : commencez par les fonctions de récompense de base et répétez
- Testez Lambda séparément : utilisez les événements de test Lambda avant l'évaluation complète
- Valider sur un petit ensemble de données : exécuter l'évaluation sur le sous-ensemble avant le jeu de données complet
- Contrôle des versions : suivez les versions des fonctions de récompense parallèlement aux versions des modèles
- Surveillez les coûts : les appels Lambda et le temps de calcul ont une incidence sur les coûts
- Journaliser intensivement : utilisez des instructions d'impression dans Lambda pour le débogage
- Définissez les délais de manière appropriée : équilibre entre patience et coût
- Métriques du document : définissez clairement ce que mesure chaque métrique

Étapes suivantes

Après avoir terminé l'évaluation RFT :

- Si les résultats sont satisfaisants : déployer le modèle en production

- Si des améliorations sont nécessaires :
 - Ajuster la fonction de récompense
 - Collectez davantage de données d'entraînement
 - Modifier les hyperparamètres d'entraînement
 - Exécutez des itérations de formation RFT supplémentaires
- Surveillance continue : réévaluez périodiquement avec de nouvelles données

Mettre en œuvre des fonctions de récompense

Présentation de

La fonction de récompense (également appelée scorer ou grader) est l'élément central qui évalue les réponses du modèle et fournit des signaux de feedback pour l'entraînement. Elle doit être implémentée en tant que fonction Lambda qui accepte les réponses du modèle et renvoie des scores de récompense.

Format d'interface

Votre fonction de récompense doit accepter et renvoyer les données au format suivant :

Exemple d'échantillon d'entrée pour la formation

```
{
  "messages": [
    {
      "role": "user",
      "content": "Do you have a dedicated security team?"
    }
  ],
  "reference_answer": {
    "compliant": "No",
    "explanation": "As an AI developed by Company, I do not have a traditional security team..."
  }
}
```

Exemple de charge utile pour le Reward Lambda

Le conteneur transforme automatiquement vos données avant de les envoyer à votre fonction Lambda en :

1. Génération d'un modèle de réponse pour chaque invite
2. Ajouter le tour de l'assistant (réponse générée) au tableau des messages
3. Ajouter un id champ unique pour le suivi

Votre fonction Lambda recevra les données dans ce format transformé :

```
{
  "id": "123",
  "messages": [
    {
      "role": "user",
      "content": "Do you have a dedicated security team?"
    },
    {
      "role": "assistant",
      "content": "As an AI developed by Amazon, I don not have a dedicated
security team..."
    }
  ],
  # Following section will be same as your training dataset sample
  "reference_answer": {
    "compliant": "No",
    "explanation": "As an AI developed by Company, I do not have a traditional
security team..."
  }
}
```

Contrat Reward Lambda

```
def lambda_handler(event, context):
    return lambda_grader(event)

def lambda_grader(samples: list[dict]) -> list[dict]:
    """
    Args:
        samples: List of dictionaries in OpenAI format

    Example input:
    {
        "id": "123",
        "messages": [
```

```
        {
            "role": "user",
            "content": "Do you have a dedicated security team?"
        },
        {
            "role": "assistant",
            "content": "As an AI developed by Company, I don nott have a
dedicated security team..."
        }
    ],
    # This section will be same as your training dataset
    "reference_answer": {
        "compliant": "No",
        "explanation": "As an AI developed by Company, I do not have a
traditional security team..."
    }
}

Returns:
List of dictionaries with reward scores:
{
    "id": str,                                # Same id as input sample
    "aggregate_reward_score": float,          # Overall score for the sample
    "metrics_list": [                         # OPTIONAL: Component scores
        {
            "name": str,                      # Name of the component score
            "value": float,                   # Value of the component score
            "type": str                       # "Reward" or "Metric"
        }
    ]
}
"""
```

Champs d'entrée et de sortie

Champs de saisie

Champ	Description	Informations complémentaires
id	Identifiant unique pour l'échantillon	Répertorié dans la sortie. Format de chaîne

Champ	Description	Informations complémentaires
messages	Historique des discussions ordonné au format OpenAI	Tableau d'objets de message
messages [].role	Intervenant du message	Valeurs communes : « utilisateur », « assistant », « système »
messages [].content	Contenu textuel du message	Chaîne de texte brut
**métadonnées	Informations sous forme libre pour faciliter le classement	Objet ; champs facultatifs transmis à partir des données d'entraînement

Champs de sortie

Champ	Description	Informations complémentaires
id	Identifiant identique à celui de l'échantillon d'entrée	Doit correspondre à l'entrée
score de récompense agrégé	Note globale pour l'échantillon	Valeur flottante (par exemple, 0,0-1,0 ou plage définie par la tâche)
liste_métriques	Scores des composants qui constituent l'agrégat	Tableau d'objets métriques

Contraintes techniques

- Limite de délai d'exécution : 15 minutes maximum par appel Lambda
- Concurrence — Doit gérer les demandes `rollout_worker_replicas * 64` simultanées
- Fiabilité — Doit mettre en œuvre une gestion appropriée des erreurs et renvoyer des scores valides de manière cohérente
- Performances — Optimisation pour une exécution rapide (quelques secondes, et non quelques minutes) afin de permettre un entraînement efficace

Bonnes pratiques

- Minimiser les appels d'API externes
- Utiliser des algorithmes et des structures de données efficaces
- Implémenter une logique de nouvelle tentative pour les échecs transitoires
- Mettre en cache les calculs réutilisables
- Effectuez des tests approfondis avant l'entraînement pour garantir une exécution sans bogue

Utilisation de fonctions de récompense personnalisées

Implémentez des fonctions de récompense personnalisées lorsque vous avez des critères d'évaluation spécifiques à une tâche :

- Définissez les critères d'évaluation — Déterminez ce qui constitue une bonne réponse à votre tâche
- Implémenter la fonction Lambda — Créez une fonction Lambda selon le format de l'interface
- Testez localement : validez que votre fonction renvoie des scores corrects pour les échantillons d'entrées
- Déployer vers AWS : déployez votre Lambda et notez l'ARN
- Configurer la recette — Ajoutez l'ARN Lambda au champ de votre recette `reward_lambda_arn`
- Test avec un petit ensemble de données — Exécutez RFT avec un minimum de données pour vérifier l'intégration

Autorisations IAM

Autorisations requises

Votre rôle SageMaker d'exécution doit être autorisé à appeler votre fonction Lambda. Ajoutez cette politique à votre rôle SageMaker d'exécution :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ]
    }
  ]
}
```

```
    ],  
    "Resource": "arn:aws:lambda:region:account-id:function:function-name"  
  }  
]  
}
```

Rôle d'exécution Lambda

Le rôle d'exécution de votre fonction Lambda nécessite des autorisations d'exécution Lambda de base :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "logs:CreateLogGroup",  
        "logs:CreateLogStream",  
        "logs:PutLogEvents"  
      ],  
      "Resource": "arn:aws:logs:*:*:*"  
    }  
  ]  
}
```

Autorisations supplémentaires : si votre fonction Lambda accède à d'autres AWS services (par exemple, S3 pour les données de référence, DynamoDB pour la journalisation), ajoutez ces autorisations au rôle d'exécution Lambda.

Exemple : fonction de récompense LLM As a Judge

Cet exemple montre comment utiliser les modèles Amazon Bedrock comme juges pour évaluer les réponses des modèles en les comparant aux réponses de référence. Ce modèle Lambda fournit un cadre permettant aux clients de mettre en œuvre des appels à Amazon Bedrock pour des demandes d'inférence afin de traiter les évaluations des juges. La fonction Lambda maintient le même input/output contrat que les autres fonctions de récompense.

Mise en œuvre

Cette fonction Lambda met en œuvre un processus d'évaluation en deux étapes : elle `lambda_handler` extrait les réponses du modèle et les réponses de référence des échantillons

entrants, puis la fonction `lambda_graded` appelle Amazon Bedrock pour évaluer la similitude sémantique entre elles. La mise en œuvre inclut une gestion robuste des erreurs avec des tentatives automatiques pour les échecs transitoires et prend en charge des formats de réponse de référence flexibles (formats de chaîne et de dictionnaire structuré).

Détails de mise en œuvre :

- Logique de réessai : implémente un ralentissement exponentiel (1 s, 2, 4 s) pour limiter les exceptions afin de gérer les limites de débit de l'API Bedrock
- Gestion des erreurs : renvoie un score de 0,0 en cas d'échec des évaluations plutôt que de déclencher des exceptions
- Notation déterministe : utilise la température = 0,0 pour garantir des scores cohérents entre les évaluations
- Format de référence flexible : gère automatiquement les réponses de référence sous forme de chaîne et de dictionnaire
- Limitation des scores : garantit que tous les scores se situent dans une plage valide [0,0, 1,0]
- Indépendant du modèle : modifiez `JUDGE_MODEL_ID` pour utiliser n'importe quel modèle Amazon Bedrock (Nova, Llama, Mistral, etc.)

```
"""
LLM Judge Lambda POC - Working implementation using Amazon Bedrock
"""

import json
import time
import boto3

bedrock_runtime = boto3.client('bedrock-runtime', region_name='us-east-1')
JUDGE_MODEL_ID = "anthropic.claude-3-5-sonnet-20240620-v1:0"
SYSTEM_PROMPT = "You must output ONLY a number between 0.0 and 1.0. No explanations, no text, just the number."

JUDGE_PROMPT_TEMPLATE = """Compare the following two responses and rate how similar they are on a scale of 0.0 to 1.0, where:
- 1.0 means the responses are semantically equivalent (same meaning, even if worded differently)
- 0.5 means the responses are partially similar
- 0.0 means the responses are completely different or contradictory
```

Response A: {response_a}

Response B: {response_b}

Output ONLY a number between 0.0 and 1.0. No explanations."""

```
def lambda_graded(response_a: str, response_b: str, max_retries: int = 3) -> float:
    """Call Bedrock to compare responses and return similarity score."""
    prompt = JUDGE_PROMPT_TEMPLATE.format(response_a=response_a, response_b=response_b)

    for attempt in range(max_retries):
        try:
            response = bedrock_runtime.converse(
                modelId=JUDGE_MODEL_ID,
                messages=[{"role": "user", "content": [{"text": prompt}]}],
                system=[{"text": SYSTEM_PROMPT}],
                inferenceConfig={"temperature": 0.0, "maxTokens": 10}
            )
            print(f"Bedrock call successful: {response}")
            output = response['output']['message']['content'][0]['text'].strip()
            score = float(output)
            print(f"Score parsed: {score}")
            return max(0.0, min(1.0, score))

        except Exception as e:
            if "ThrottlingException" in str(e) and attempt < max_retries - 1:
                time.sleep(2 ** attempt)
            else:
                print(f"Bedrock call failed: {e}")
                return None

    return None

def lambda_handler(event, context):
    """AWS Lambda handler - processes samples from RFTEvalInvoker."""
    try:
        samples = event if isinstance(event, list) else [event]
        results = []

        for sample in samples:
            sample_id = sample.get("id", "unknown")
            messages = sample.get("messages", [])
```

```

# Extract assistant response (response A)
response_a = ""
for msg in messages:
    if msg.get("role") in ["assistant", "nova_assistant"]:
        response_a = msg.get("content", "")
        break

# Extract reference answer from root level (no longer in metadata)
reference_answer = sample.get("reference_answer", "")

# Handle both string and dict reference_answer formats
if isinstance(reference_answer, dict):
    # If reference_answer is a dict, extract the explanation or compliant
    response_b = reference_answer.get("explanation",
reference_answer.get("compliant", ""))
else:
    response_b = reference_answer

if not response_a or not response_b:
    results.append({
        "id": sample_id,
        "aggregate_reward_score": 0.0,
        "metrics_list": [{"name": "similarity_score", "value": 0.0, "type":
"Metric"}]
    })
    continue

# Get similarity score
score = lambda_graded(response_a, response_b)

results.append({
    "id": sample_id,
    "aggregate_reward_score": score,
    "metrics_list": [
        {
            "name": "similarity_score",
            "value": score,
            "type": "Metric"
        }
    ]
})

```

```
        return {"statusCode": 200, "body": json.dumps(results)}

    except Exception as e:
        print(f"Error: {e}")
        return {"statusCode": 500, "body": json.dumps({"error": str(e)})}
```

Format d'entrée

Le Lambda reçoit le même format de saisie que les autres fonctions de récompense :

```
{
  "id": "sample-001",
  "messages": [
    {
      "role": "user",
      "content": "Do you have a dedicated security team?"
    },
    {
      "role": "assistant",
      "content": "As an AI developed by Amazon, I don't have a dedicated security
team..."
    }
  ],
  "reference_answer": {
    "compliant": "No",
    "explanation": "As an AI developed by Company, I do not have a traditional
security team..."
  },
  "my_custom_field": "custom_value"
}
```

Format de sortie

```
{
  "id": "sample-001",
  "aggregate_reward_score": 0.85,
  "metrics_list": [
    {
      "name": "similarity_score",
      "value": 0.85,
      "type": "Metric"
    }
  ]
}
```

```
}
```

Considérations relatives au déploiement

Vous devrez peut-être également ajuster le modèle d'invite et les paramètres d'inférence en fonction des capacités du modèle que vous avez choisi et du format d'API.

- **Autorisations IAM** : le rôle d'exécution Lambda doit être autorisé pour le `bedrock:InvokeModel` modèle que vous avez choisi
- **Délai d'expiration** : définissez le délai Lambda à au moins 60 secondes pour tenir compte de la latence de l'API Bedrock et des nouvelles tentatives
- **Région** : Déployez dans une région où le modèle Bedrock que vous avez choisi est disponible
- **Coût** : surveillez l'utilisation de l'API Bedrock car chaque évaluation effectue un appel d'API par échantillon
- **Débit** : pour les évaluations à grande échelle, demandez des quotas Bedrock accrus pour éviter les ralentissements

Augmenter le débit du substrat rocheux

Si vous rencontrez un ralentissement pendant l'évaluation, augmentez les quotas de votre modèle Bedrock :

- Accédez à la console AWS Service Quotas
- Recherchez « Bedrock » et sélectionnez votre région
- Trouvez le quota pour le modèle que vous avez choisi (par exemple, « Invocations par minute pour Claude 3.5 Sonnet »)
- Cliquez sur « Demander une augmentation du quota » et spécifiez le débit souhaité
- Justifiez l'augmentation (par exemple, « charge de travail d'évaluation RFT »)

La logique de relance intégrée au Lambda gère les ralentissements occasionnels, mais les évaluations soutenues de volumes élevés nécessitent des augmentations de quotas appropriées.

Politique IAM requise :

```
{
```



```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "bedrock:InvokeModel"
    ],
    "Resource": "arn:aws:bedrock:*::foundation-model/*"
  }
]
```

Entraînement itératif

La formation itérative est une approche systématique visant à affiner les modèles au moyen de plusieurs cycles de formation, au cours desquels chaque cycle s'appuie sur le point de contrôle précédent en corrigeant les faiblesses spécifiques découvertes lors de l'évaluation. Cette méthode permet d'améliorer de manière ciblée les performances des modèles en incorporant des exemples sélectionnés qui traitent des modes de défaillance, en s'adaptant aux exigences changeantes et en validant les améliorations de manière progressive plutôt que de s'engager dans un seul long cycle de formation. Le processus suit généralement des modèles tels que le SFT (Supervised Fine-Tuning) suivi du RFT (Reward-based Fine-Tuning), avec des points de contrôle stockés dans des compartiments S3 séquestres AWS gérés qui peuvent être référencés pour les itérations d'entraînement suivantes tout en maintenant la cohérence du type de modèle et de la technique d'entraînement tout au long du pipeline.

Pour plus d'informations, consultez [???](#).

Personnalisation d'Amazon Nova sur SageMaker HyperPod

Vous pouvez personnaliser les modèles Amazon Nova, y compris les modèles Amazon Nova 2.0 améliorés, à l'aide des [recettes Amazon Nova](#) et les entraîner sur Hyperpod. Une recette est un fichier de configuration YAML qui fournit des informations à l' SageMaker IA sur la manière d'exécuter votre tâche de personnalisation de modèle. SageMaker HyperPod prend en charge deux types de services : Forge et non-Forge.

Hyperpod propose un calcul haute performance avec des instances GPU optimisées et un stockage Amazon FSx for Lustre, une surveillance robuste grâce à l'intégration d'outils tels que TensorBoard la gestion flexible des points de contrôle pour une amélioration itérative, un déploiement fluide sur

Amazon Bedrock à des fins d'inférence et une formation distribuée multi-nœuds efficace et évolutive, le tout fonctionnant ensemble pour fournir aux entreprises un environnement sécurisé, performant et flexible leur permettant d'adapter les modèles Amazon Nova à leurs besoins commerciaux spécifiques.

La personnalisation d'Amazon Nova permet de SageMaker HyperPod stocker les artefacts du modèle, y compris les points de contrôle du modèle, dans un compartiment Amazon S3 géré par des services. Les artefacts du compartiment géré par les services sont chiffrés à l'aide de clés gérées par l' SageMaker IA AWS KMS . Les compartiments Amazon S3 gérés par service ne prennent actuellement pas en charge le chiffrement des données à l'aide de clés KMS gérées par le client. Vous pouvez utiliser cet emplacement de point de contrôle pour les tâches d'évaluation ou l'inférence Amazon Bedrock.

La tarification standard peut s'appliquer aux instances de calcul, au stockage Amazon S3 et FSx à Lustre. Pour plus de détails sur les tarifs, consultez la [tarification d'Hyperpod](#), la [tarification d'Amazon S3](#) et la [tarification FSx de Lustre](#).

Exigences de calcul pour les modèles Amazon Nova 2

Les tableaux suivants résument les exigences informatiques SageMaker HyperPod et la formation aux métiers de formation à l' SageMaker IA pour les modèles Amazon Nova 2.

Exigences de formation pour Nova 2

Technique d'entraînement	Instances minimales	Type d'instance	Nombre de GPU	Remarques	Modèles pris en charge
ST (LoRa)	4	P5.48 x large	16	Réglage précis efficace en fonction des paramètres	Nova 2 Lite
SFT (classement complet)	4	P5.48 x large	32	Réglage complet du modèle	Nova 2 Lite
RFT sur les emplois	2	P5.48 x large	16	Fonctions de récompens	Nova 2 Lite

Technique d'entraînement	Instances minimales	Type d'instance	Nombre de GPU	Remarques	Modèles pris en charge
SageMaker de formation (LoRa)				e personnalisées dans votre AWS environnement	
RFT sur les emplois SageMaker de formation (classement complet)	4	P5.48 x large	32	Longueur de contexte 32 Ko	Nova 2 Lite
RFT activé SageMaker HyperPod	8	P5.48 x large	64	Longueur de contexte par défaut : 8192	Nova 2 Lite
CPT	4	P5.48 x large	16	Traite environ 400 millions de jetons par instance et par jour	Nova 2 Lite

Pour optimiser vos flux de travail de personnalisation de modèles Amazon Nova sur Hyperpod, suivez ces bonnes pratiques recommandées pour une formation efficace, une gestion des ressources et un déploiement réussi des modèles.

Bonnes pratiques pour la personnalisation d'Amazon Nova

Présentation de

Cette section fournit un aperçu des techniques de personnalisation et vous aide à choisir l'approche la mieux adaptée à vos besoins et aux données disponibles.

Deux étapes de la formation LLM

La formation aux modèles linguistiques de grande envergure comprend deux étapes principales : la pré-formation et la post-formation. Au cours de la pré-formation, le modèle traite les jetons de texte brut et les optimise pour la prédiction du jeton suivant. Ce processus crée un compléteur de modèles qui absorbe la syntaxe, la sémantique, les faits et les modèles de raisonnement du Web et du texte organisé. Cependant, le modèle préformé ne comprend pas les instructions, les objectifs de l'utilisateur ou le comportement adapté au contexte. Il poursuit le texte dans le style qui convient à sa distribution de formation. Un modèle préentraîné se complète automatiquement au lieu de suivre les instructions, produit un formatage incohérent et peut refléter des biais indésirables ou du contenu dangereux des données d'entraînement. La formation préalable renforce les compétences générales et non l'utilité des tâches.

La post-formation transforme le compléteur de modèles en un assistant utile. Vous exécutez plusieurs cycles de réglage fin supervisé (SFT) pour apprendre au modèle à suivre les instructions, à respecter les schémas et les politiques, à appeler des outils et à produire des résultats fiables en imitant des démonstrations de haute qualité. Cet alignement apprend au modèle à répondre aux demandes sous forme de tâches plutôt que sous forme de texte pour continuer. Vous appliquez ensuite le réglage fin du renforcement (RFT) pour optimiser le comportement à l'aide de commentaires mesurables (tels que des vérificateurs ou un LLM-as-a-judge), en équilibrant des compromis tels que la précision par rapport à la brièveté, la sécurité par rapport à la couverture, ou un raisonnement en plusieurs étapes soumis à des contraintes. Dans la pratique, vous alternez SFT et RFT par cycles pour transformer le modèle préentraîné en un système fiable, aligné sur les politiques, qui exécute des tâches complexes de manière cohérente.

Choisissez la bonne approche de personnalisation

Dans cette section, nous aborderons les stratégies de personnalisation après la formation : RFT et SFT.

Réglage précis du renforcement (RFT)

Le réglage précis du renforcement améliore les performances du modèle grâce à des signaux de feedback (scores mesurables ou récompenses indiquant la qualité des réponses) plutôt qu'à une supervision directe avec des réponses exactes et correctes. Contrairement au réglage fin supervisé traditionnel qui apprend à partir de paires d'entrées-sorties, le RFT utilise des fonctions de récompense pour évaluer les réponses du modèle et optimise le modèle de manière itérative afin de maximiser ces récompenses. Cette approche fonctionne bien pour les tâches où il est difficile de définir le résultat exact correct, mais vous pouvez mesurer de manière fiable la qualité de réponse.

Le RFT permet aux modèles d'apprendre des comportements et des préférences complexes par le biais d'essais et de commentaires, ce qui le rend idéal pour les applications qui nécessitent une prise de décision nuancée, une résolution créative des problèmes ou le respect de critères de qualité spécifiques que vous pouvez évaluer par programmation. Par exemple, répondre à des questions juridiques complexes est un cas d'utilisation idéal pour la RFT, car vous souhaitez apprendre au modèle à mieux raisonner pour répondre aux questions avec plus de précision.

Comment ça marche

Pour peaufiner le renforcement, vous partez d'une base de référence adaptée aux instructions et vous traitez chaque invite comme un petit tournoi. Pour une entrée donnée, vous échantillonnez une poignée de réponses candidates à partir du modèle, vous notez chacune d'entre elles avec la fonction de récompense, puis vous les classez dans ce groupe. L'étape de mise à jour incite le modèle à augmenter la probabilité que les candidats ayant obtenu les meilleurs scores soient plus susceptibles de se présenter la prochaine fois et à ceux qui obtiennent les moins bons résultats, tandis qu'une stay-close-to-baseline contrainte empêche le comportement de dériver, de devenir verbeux ou abusif. Vous répétez cette boucle en suivant de nombreuses instructions, en actualisant les dossiers complexes, en resserrant les vérificateurs ou en évaluant les rubriques lorsque vous constatez des exploits, et vous suivez en permanence les indicateurs des tâches.

Quand utiliser le RFT

Les tâches qui bénéficient le plus de la RFT partagent plusieurs caractéristiques. Ils ont des signaux de réussite mesurables même lorsqu'il est difficile de spécifier un seul résultat correct. Ils admettent un crédit partiel ou une note de qualité, ce qui vous permet de classer les meilleures réponses par rapport aux mauvaises réponses en un clin d'œil ou en utilisant une fonction de récompense. Ils impliquent de multiples objectifs qui doivent être équilibrés (tels que la précision et la brièveté, la clarté, la sécurité ou le coût). Ils nécessitent le respect de contraintes explicites que vous pouvez vérifier par programmation. Ils fonctionnent dans des environnements médiés par des outils ou basés sur un environnement où les résultats sont observables (succès ou échec, latence, utilisation des ressources). Ils se produisent dans des régimes peu étiquetés où la collecte de cibles aurifères coûte cher, mais où les commentaires automatisés ou basés sur des rubriques sont nombreux. La RFT fonctionne mieux lorsque vous pouvez transformer la qualité en un scalaire ou un classement fiables et que vous souhaitez que le modèle amplifie de manière préférentielle les comportements les plus performants sans avoir besoin de cibles étiquetées exhaustives.

Envisagez d'autres méthodes lorsque :

- Vous avez des paires d'entrées-sorties étiquetées nombreuses et fiables — Utilisez SFT

- La principale lacune concerne les connaissances ou le jargon. Utilisez la génération augmentée par extraction (RAG)
- Votre signal de récompense est bruyant ou peu fiable et vous ne pouvez pas le corriger avec de meilleures rubriques ou de meilleures cases. Stabilisez-le d'abord avant RFT

Quand ne pas utiliser le RFT

Évitez le RFT dans les situations suivantes :

- Vous pouvez produire à moindre coût des paires d'entrées-sorties étiquetées fiables (le SFT est plus simple, moins cher et plus stable)
- L'écart réside dans les connaissances ou le jargon plutôt que dans le comportement (utilisez RAG)
- Votre signal de récompense est bruyant, faible, facile à jouer, coûteux ou lent à calculer (corrigez d'abord l'évaluateur)
- Les performances de base sont proches de zéro (amorcez avec SFT avant d'optimiser les préférences)
- La tâche comporte des schémas déterministes, un formatage strict ou une seule réponse correcte (la validation SFT ou basée sur des règles fonctionne mieux)
- Les budgets serrés en termes de latence ou de coûts ne peuvent pas absorber le surcroît d'échantillonnage ou d'exploration requis par le RFT
- Les contraintes de sécurité ou de politique ne sont pas clairement spécifiées et applicables dans la récompense

Si vous pouvez indiquer « la bonne réponse », utilisez SFT. Si vous avez besoin de nouvelles connaissances, utilisez RAG. N'utilisez RFT qu'après avoir une base de référence solide et une fonction de hard-to-exploit récompense robuste et rapide.

Peaufinage supervisé (SFT)

Le réglage fin supervisé entraîne le LLM sur un ensemble de données de paires d'entrées-sorties étiquetées par l'homme pour votre tâche. Vous fournissez des exemples de demandes (questions, instructions, etc.) avec les réponses correctes ou souhaitées, et vous poursuivez la formation du modèle sur ces exemples. Le modèle ajuste ses pondérations pour minimiser les pertes supervisées (généralement une entropie croisée entre ses prédictions et les jetons de sortie cibles). Il s'agit de la même formation que celle utilisée dans la plupart des tâches d'apprentissage automatique supervisées, appliquée pour spécialiser un LLM.

SFT modifie le comportement, pas les connaissances. Il n'enseigne pas au modèle de nouveaux faits ou un jargon qu'il n'avait pas découverts lors de la pré-formation. Il enseigne au modèle comment répondre, et non ce qu'il doit savoir. Si vous avez besoin de nouvelles connaissances du domaine (telles que la terminologie interne), utilisez la génération augmentée par extraction (RAG) pour fournir ce contexte au moment de l'inférence. SFT ajoute ensuite le comportement de suivi des instructions souhaité en haut.

Comment ça marche

SFT optimise le LLM en minimisant la perte moyenne d'entropie croisée sur les jetons de réponse, en traitant les jetons prompts comme contexte et en les masquant de la perte. Le modèle internalise le style, la structure et les règles de décision de votre cible, en apprenant à générer la saisie correcte pour chaque invite. Par exemple, pour classer des documents dans des catégories personnalisées, vous affinez le modèle à l'aide d'instructions (le texte du document) et de compléments étiquetés (les étiquettes de catégorie). Vous vous entraînez sur ces paires jusqu'à ce que le modèle affiche la bonne étiquette pour chaque invite avec une probabilité élevée.

Vous pouvez exécuter le SFT avec seulement quelques centaines d'exemples et passer à quelques centaines de milliers. Les échantillons SFT doivent être de haute qualité et directement alignés sur le comportement du modèle souhaité.

Quand utiliser SFT

Utilisez SFT lorsque vous avez une tâche bien définie avec des résultats clairement définis. Si vous pouvez indiquer explicitement « Étant donné une entrée X, la sortie correcte est Y » et recueillir des exemples de tels mappages, le réglage fin supervisé est un bon choix. SFT excelle dans les scénarios suivants :

- Tâches de classification structurées ou complexes — Classez les documents internes ou les contrats dans de nombreuses catégories personnalisées. Avec SFT, le modèle apprend ces catégories spécifiques mieux que de le demander uniquement.
- Tâches de réponse à des questions ou de transformation avec des réponses connues : affinez un modèle pour répondre aux questions de la base de connaissances d'une entreprise, ou convertissez les données entre des formats dans lesquels chaque entrée contient une réponse correcte.
- Mise en forme et cohérence du style — Entraînez le modèle à toujours répondre dans un certain format ou dans un certain ton en peaufinant sur des exemples de format ou de ton corrects. Par exemple, une formation sur des paires prompt-réponse illustrant la voix d'une marque particulière apprend au modèle à générer des résultats dans ce style. Le comportement suivant les instructions

est souvent initialement enseigné par le biais de SFT sur des exemples sélectionnés de bons comportements d'assistant.

SFT est le moyen le plus direct d'enseigner à un LLM une nouvelle compétence ou un nouveau comportement lorsque vous pouvez spécifier à quoi ressemble le bon comportement. Il utilise la compréhension linguistique existante du modèle et le concentre sur votre tâche. Utilisez SFT lorsque vous souhaitez que le modèle fasse une chose spécifique et que vous avez ou pouvez créer un jeu de données d'exemples.

Utilisez SFT lorsque vous pouvez assembler des paires d'invite et de réponse de haute qualité qui reflètent fidèlement le comportement souhaité. Il convient aux tâches comportant des cibles claires ou des formats déterministes tels que les schémas, les appels de fonctions ou d'outils, et aux réponses structurées pour lesquelles l'imitation est un signal d'apprentissage approprié. L'objectif est de façonner le comportement : apprendre au modèle à traiter les instructions comme des tâches, à suivre les instructions, à adopter un ton et des politiques de refus, et à produire un formatage cohérent. Planifiez au moins des centaines de démonstrations, la qualité des données, la cohérence et la déduplication étant plus importantes que le volume brut. Pour une mise à jour simple et rentable, utilisez des méthodes économes en paramètres, telles que l'adaptation de bas niveau, pour entraîner de petits adaptateurs sans toucher à la majeure partie de la colonne vertébrale.

Quand ne pas utiliser SFT

N'utilisez pas SFT lorsque l'écart est lié aux connaissances plutôt qu'au comportement. Il n'enseigne pas au modèle de nouveaux faits, de jargon ou d'événements récents. Dans ces cas, utilisez la génération augmentée par extraction pour intégrer des connaissances externes lors de l'inférence. Évitez le SFT lorsque vous pouvez mesurer la qualité mais que vous ne pouvez pas étiqueter une seule bonne réponse. Utilisez le renforcement pour affiner les récompenses avec des récompenses vérifiables ou LLM-as-a-judge pour optimiser directement ces récompenses. Si vos besoins ou votre contenu changent fréquemment, misez sur la récupération et l'utilisation des outils plutôt que sur le perfectionnement du modèle.

Rubriques

- [Kit de développement logiciel Nova Forge](#)
- [Création d'un cluster SageMaker HyperPod EKS avec un groupe d'instances restreint \(RIG\)](#)
- [Guide des commandes Amazon SageMaker HyperPod Essential](#)
- [Accès à Nova Forge et configuration pour](#)
- [Formation pour les modèles Amazon Nova](#)

- [Évaluation de votre modèle entraîné](#)
- [Suivi des HyperPod tâches avec MLflow](#)

Kit de développement logiciel Nova Forge

Le SDK Amazon Nova Forge est un SDK Python complet qui fournit une interface de programmation unifiée pour le cycle de vie complet de personnalisation des modèles Amazon Amazon Nova. Le SDK simplifie la personnalisation des modèles en proposant une API unique et cohérente pour la formation, l'évaluation, le suivi, le déploiement et l'inférence sur les plateformes Amazon et SageMaker Amazon Bedrock.

Pour de plus amples informations, veuillez consulter [Kit de développement logiciel Nova Forge](#).

Création d'un cluster SageMaker HyperPod EKS avec un groupe d'instances restreint (RIG)

Pour personnaliser un modèle sur Hyperpod, l'infrastructure nécessaire doit être configurée. Pour plus de détails sur la configuration d'un cluster SageMaker HyperPod EKS avec un groupe d'instances restreint (RIG), consultez l'[atelier](#), qui fournit une présentation détaillée du processus de configuration.

Guide des commandes Amazon SageMaker HyperPod Essential

Amazon SageMaker HyperPod fournit des fonctionnalités de ligne de commande étendues pour gérer les flux de travail de formation. Ce guide couvre les commandes essentielles pour les opérations courantes, de la connexion à votre cluster au suivi de l'avancement des tâches.

Conditions préalables

Avant d'utiliser ces commandes, assurez-vous d'avoir effectué la configuration suivante :

- SageMaker HyperPod cluster avec RIG créé (généralement dans us-east-1)
- Compartiment de sortie Amazon S3 créé pour les artefacts d'entraînement
- Rôles IAM configurés avec les autorisations appropriées
- Données d'entraînement téléchargées au format JSONL correct
- FSx pour que la synchronisation avec Lustre soit terminée (vérifiez dans les journaux du cluster lors de la première tâche)

Rubriques

- [Installation de Recipe CLI](#)
- [Connexion à votre cluster](#)
- [Démarrage d'un poste de formation](#)
- [Vérification de l'état du travail](#)
- [Surveillance des journaux des tâches](#)
- [Lister les offres d'emploi actives](#)
- [Annulation d'une tâche](#)
- [Exécution d'une tâche d'évaluation](#)
- [Problèmes courants](#)

Installation de Recipe CLI

Accédez à la racine de votre référentiel de recettes avant d'exécuter la commande d'installation.

Utilisez le référentiel Hyperpodrerecipes si vous utilisez des techniques de personnalisation autres que Forge. Pour la personnalisation basée sur Forge, reportez-vous au référentiel de recettes spécifique à la forge.

Exécutez les commandes suivantes pour installer la SageMaker HyperPod CLI :

Note

Assurez-vous que vous n'êtes pas dans un environnement conda, anaconda, miniconda actif ou dans un autre environnement virtuel

Si c'est le cas, veuillez quitter l'environnement en utilisant :

- `conda deactivate` pour les environnements conda, anaconda et miniconda
- `deactivate` pour les environnements virtuels Python

Si vous utilisez une technique de personnalisation autre que Forge, sagemaker-hyperpod-recipes téléchargez-la comme indiqué ci-dessous :

```
git clone -b release_v2 https://github.com/aws/sagemaker-hyperpod-cli.git
```

```
cd sagemaker-hyperpod-cli
pip install -e .
cd ..
root_dir=$(pwd)
export PYTHONPATH=${root_dir}/sagemaker-hyperpod-cli/src/hyperpod_cli/
sagemaker_hyperpod_recipes/launcher/nemo/nemo_framework_launcher/launcher_scripts:
$PYTHONPATH
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh
rm -f ./get_helm.sh
```

Si vous êtes abonné à Forge, vous devriez télécharger les recettes en utilisant le processus mentionné ci-dessous.

```
mkdir NovaForgeHyperpodCLI
cd NovaForgeHyperpodCLI
aws s3 cp s3://nova-forge-c7363-206080352451-us-east-1/v1/ ./ --recursive
pip install -e .

curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh
rm -f ./get_helm.sh
```

Tip

Pour utiliser un [nouvel environnement virtuel](#) avant de l'exécuter `pip install -e .`, exécutez :

- `python -m venv nova_forge`
- `source nova_forge/bin/activate`
- Votre ligne de commande s'affichera désormais (nova_forge) au début de votre invite
- Cela garantit qu'il n'y a pas de dépendances concurrentes lors de l'utilisation de la CLI

Objectif : Pourquoi le faisons-nous `pip install -e .` ?

Cette commande installe la SageMaker HyperPod CLI en mode modifiable, ce qui vous permet d'utiliser des recettes mises à jour sans avoir à les réinstaller à chaque fois. Cela vous permet également d'ajouter de nouvelles recettes que la CLI peut automatiquement récupérer.

Connexion à votre cluster

Connectez la SageMaker HyperPod CLI à votre cluster avant d'exécuter des tâches :

```
export AWS_REGION=us-east-1 && SageMaker HyperPod connect-cluster --cluster-name  
<your-cluster-name> --region us-east-1
```

Important

Cette commande crée un fichier de contexte (`/tmp/hyperpod_context.json`) requis par les commandes suivantes. Si vous voyez un message d'erreur concernant ce fichier introuvable, réexécutez la commande `connect`.

Conseil de pro : Vous pouvez configurer davantage votre cluster pour qu'il utilise toujours l'`kubeflow` espace de noms en ajoutant l'`--namespace kubeflow` argument à votre commande comme suit :

```
export AWS_REGION=us-east-1 && \  
hyperpod connect-cluster \  
--cluster-name <your-cluster-name> \  
--region us-east-1 \  
--namespace kubeflow
```

Cela vous évite d'avoir à ajouter la commande `-n kubeflow` dans chaque commande lorsque vous interagissez avec vos tâches.

Démarrage d'un poste de formation

Note

Si vous exécutez PPO/RFT des tâches, assurez-vous d'ajouter des paramètres de sélection d'étiquettes `src/hyperpod_cli/sagemaker_hyperpod_recipes/recipes_collection/cluster/k8s.yaml` afin que tous les modules soient planifiés sur le même nœud.

```
label_selector:
  required:
    sagemaker.amazonaws.com/instance-group-name:
      - <rig_group>
```

Lancez une tâche de formation à l'aide d'une recette avec des remplacements de paramètres facultatifs :

```
hyperpod start-job -n kubeflow \
--recipe fine-tuning/nova/nova_1_0/nova_micro/SFT/nova_micro_1_0_p5_p4d_gpu_lora_sft \
--override-parameters '{
"instance_type": "ml.p5.48xlarge",
  "container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-
HP-SFT-latest"
}'
```

Résultat attendu :

```
Final command: python3 <path_to_your_installation>/NovaForgeHyperpodCLI/src/
hyperpod_cli/sagemaker_hyperpod_recipes/main.py recipes=fine-tuning/nova/
nova_micro_p5_gpu_sft cluster_type=k8s cluster=k8s base_results_dir=/local/home/
<username>/results cluster.pullPolicy="IfNotPresent" cluster.restartPolicy="OnFailure"
  cluster.namespace="kubeflow" container="708977205387.dkr.ecr.us-east-1.amazonaws.com/
nova-fine-tune-repo:HP-SFT-DATAMIX-latest"
```

```
Prepared output directory at /local/home/<username>/results/<job-name>/k8s_templates
Found credentials in shared credentials file: ~/.aws/credentials
Helm script created at /local/home/<username>/results/<job-name>/<job-name>_launch.sh
Running Helm script: /local/home/<username>/results/<job-name>/<job-name>_launch.sh
```

```
NAME: <job-name>
LAST DEPLOYED: Mon Sep 15 20:56:50 2025
NAMESPACE: kubeflow
STATUS: deployed
REVISION: 1
TEST SUITE: None
Launcher successfully generated: <path_to_your_installation>/NovaForgeHyperpodCLI/src/
hyperpod_cli/sagemaker_hyperpod_recipes/launcher/nova/k8s_templates/SFT

{
```

```
"Console URL": "https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/cluster-management/<your-cluster-name>"
}
```

Vérification de l'état du travail

Surveillez vos tâches en cours à l'aide de kubectl :

```
kubectl get pods -o wide -w -n kubeflow | (head -n1 ; grep <your-job-name>)
```

Comprendre le statut des pods

Le tableau suivant explique les statuts courants des pods :

Statut	Description
Pending	Pod accepté mais pas encore programmé sur un nœud, ou en attente d'extraction des images du conteneur
Running	Pod lié à un nœud avec au moins un conteneur en cours d'exécution ou en cours de démarrage
Succeeded	Tous les conteneurs ont été terminés avec succès et ne redémarreront pas
Failed	Tous les conteneurs se sont arrêtés, dont au moins un s'est terminé par un échec
Unknown	L'état du pod ne peut pas être déterminé (généralement en raison de problèmes de communication entre nœuds)
CrashLoopBackOff	Défaillance répétée du conteneur ; Kubernetes recule après les tentatives de redémarrage
ImagePullBackOff / ErrImagePull	Impossible d'extraire l'image du conteneur du registre

Statut	Description
OOMKilled	Conteneur fermé pour dépassement des limites de mémoire
Completed	Job ou pod terminé avec succès (fin du travail par lots)

Tip

Utilisez le `-w` drapeau pour suivre les mises à jour de l'état du pod en temps réel. Appuyez `Ctrl+C` pour arrêter de regarder.

Surveillance des journaux des tâches

Vous pouvez consulter vos journaux de trois manières :

En utilisant CloudWatch

Vos journaux sont disponibles dans votre AWS compte qui contient l'Hyperpodcluster sous.

CloudWatch Pour les consulter dans votre navigateur, rendez-vous sur la CloudWatch page d'accueil de votre compte et recherchez le nom de votre cluster. Par exemple, si votre cluster était appelé, `my-hyperpod-rig` le groupe de journaux aurait le préfixe suivant :

- Groupe de journaux : `/aws/sagemaker/Clusters/my-hyperpod-rig/{UUID}`
- Une fois que vous êtes dans le groupe de journaux, vous pouvez trouver votre journal spécifique à l'aide de l'ID d'instance du nœud, tel que `-hyperpod-i-00b3d8a1bf25714e4`.
 - `i-00b3d8a1bf25714e4` ici représente le nom de la machine Hyperpodfriendly sur laquelle s'exécute votre tâche de formation. Rappelez-vous comment, dans la `kubectl get pods -o wide -w -n kubeflow | (head -n1 ; grep my-cpt-run)` sortie de commande précédente, nous avons capturé une colonne appelée `NODE`.
 - Le nœud « maître » exécuté était dans ce cas exécuté sur `hyperpod-i-00b3d8a1bf25714e4` et nous allons donc utiliser cette chaîne pour sélectionner le groupe de journaux à afficher. Sélectionnez celui qui dit `SagemakerHyperPodTrainingJob/rig-group/[NODE]`

Utilisation d' CloudWatch Insights

Si vous avez le nom de votre poste à portée de main et que vous ne souhaitez pas suivre toutes les étapes ci-dessus, vous pouvez simplement rechercher tous les journaux `/aws/sagemaker/Clusters/my-hyperpod-rig/{UUID}` ci-dessous pour trouver le journal individuel.

CPT :

```
fields @timestamp, @message, @logStream, @log
| filter @message like /(?!i)Starting CPT Job/
| sort @timestamp desc
| limit 100
```

Pour terminer le travail, remplacer `Starting CPT Job` par `CPT Job completed`

Ensuite, vous pouvez cliquer sur les résultats et choisir celui qui indique « Epoch 0 », car ce sera votre nœud principal.

À l'aide du AWS CLI

Vous pouvez choisir de suivre vos journaux à l'aide de la AWS CLI. Avant de le faire, veuillez vérifier votre version d'aws cli à l'aide de `aws --version`. Il est également recommandé d'utiliser ce script utilitaire qui facilite le suivi des journaux en direct sur votre terminal

pour V1 :

```
aws logs get-log-events \
--log-group-name /aws/sagemaker/YourLogGroupName \
--log-stream-name YourLogStream \
--start-from-head | jq -r '.events[].message'
```

pour V2 :

```
aws logs tail /aws/sagemaker/YourLogGroupName \
--log-stream-name YourLogStream \
--since 10m \
--follow
```

Lister les offres d'emploi actives

Afficher toutes les tâches exécutées dans votre cluster :

```
hyperpod list-jobs -n kubeflow
```


Exemple de sortie :

```
{
  "jobs": [
    {
      "Name": "test-run-nhgza",
      "Namespace": "kubeflow",
      "CreationTime": "2025-10-29T16:50:57Z",
      "State": "Running"
    }
  ]
}
```

Annulation d'une tâche

Arrêtez une tâche en cours d'exécution à tout moment :

```
hyperpod cancel-job --job-name <job-name> -n kubeflow
```

Trouver le nom de votre poste

Option 1 : À partir de votre recette

Le nom de la tâche est spécifié dans le run bloc de votre recette :

```
run:
  name: "my-test-run"                # This is your job name
  model_type: "amazon.nova-micro-v1:0:128k"
  ...
```

Option 2 : depuis la commande list-jobs

Utilisez `hyperpod list-jobs -n kubeflow` et copiez le Name champ à partir de la sortie.

Exécution d'une tâche d'évaluation

Évaluez un modèle entraîné ou un modèle de base à l'aide d'une recette d'évaluation.

Conditions préalables

Avant d'exécuter des tâches d'évaluation, assurez-vous d'avoir :

- Vérifiez l'URI Amazon S3 depuis le `manifest.json` fichier de votre tâche de formation (pour les modèles entraînés)
- Ensemble de données d'évaluation chargé sur Amazon S3 dans le bon format
- Afficher le chemin Amazon S3 pour les résultats de l'évaluation

Commande

Exécutez la commande suivante pour démarrer une tâche d'évaluation :

```
hyperpod start-job -n kubeflow \  
  --recipe evaluation/nova/nova_2_0/nova_lite/  
nova_lite_2_0_p5_48xl_gpu_bring_your_own_dataset_eval \  
  --override-parameters '{  
    "instance_type": "p5.48xlarge",  
    "container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-  
HP-Eval-latest",  
    "recipes.run.name": "<your-eval-job-name>",  
    "recipes.run.model_name_or_path": "<checkpoint-s3-uri>",  
    "recipes.run.output_s3_path": "s3://<your-bucket>/eval-results/",  
    "recipes.run.data_s3_path": "s3://<your-bucket>/eval-data.jsonl"  
  }'
```

Descriptions des paramètres :

- `recipes.run.name`: nom unique pour votre tâche d'évaluation
- `recipes.run.model_name_or_path`: URI Amazon S3 depuis `manifest.json` ou chemin du modèle de base (par exemple, `nova-micro/prod`)
- `recipes.run.output_s3_path`: emplacement Amazon S3 pour les résultats de l'évaluation
- `recipes.run.data_s3_path`: emplacement de votre ensemble de données d'évaluation sur Amazon S3

Conseils :

- Recettes spécifiques au modèle : chaque taille de modèle (micro, lite, pro) possède sa propre recette d'évaluation
- Évaluation du modèle de base : utilisez les chemins du modèle de base (par exemple, `nova-micro/prod`) au lieu du point de contrôle URIs pour évaluer les modèles de base

Format des données d'évaluation

Format d'entrée (JSONL) :

```
{
  "metadata": "{key:4, category:'apple'}",
  "system": "arithmetic-patterns, please answer the following with no other words: ",
  "query": "What is the next number in this series? 1, 2, 4, 8, 16, ?",
  "response": "32"
}
```

Format de sortie :

```
{
  "prompt": "[{'role': 'system', 'content': 'arithmetic-patterns, please answer the following with no other words: '}, {'role': 'user', 'content': 'What is the next number in this series? 1, 2, 4, 8, 16, ?'}]",
  "inference": "['32']",
  "gold": "32",
  "metadata": "{key:4, category:'apple'}"
}
```

Descriptions des champs :

- **prompt**: entrée formatée envoyée au modèle
- **inference**: réponse générée par le modèle
- **gold**: réponse correcte attendue de l'ensemble de données en entrée
- **metadata**: métadonnées facultatives transmises depuis l'entrée

Problèmes courants

- **ModuleNotFoundError**: No module named 'nemo_launcher', vous devrez peut-être ajouter des éléments `nemo_launcher` à votre chemin python en fonction de l'endroit où `hyperpod_cli` il est installé. Exemple de commande :

```
export PYTHONPATH=<path_to_hyperpod_cli>/sagemaker-hyperpod-cli/src/hyperpod_cli/
sagemaker_hyperpod_recipes/launcher/nemo/nemo_framework_launcher/launcher_scripts:
$PYTHONPATH
```

- `FileNotFoundException: [Errno 2] No such file or directory: '/tmp/hyperpod_current_context.json'` indique que vous n'avez pas exécuté la commande Hyperpod Connect Cluster.
- Si votre tâche n'est pas planifiée, vérifiez si la sortie de votre SageMaker HyperPod CLI contient cette section avec les noms des tâches et d'autres métadonnées. Sinon, réinstallez helm chart en exécutant :

```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh
rm -f ./get_helm.sh
```

Accès à Nova Forge et configuration pour

Pour configurer Amazon Nova Forge afin de l'utiliser avec Jobs, vous devez :

- Abonnez-vous à Amazon Nova Forge
- Configuration d'un cluster

Rubriques

- [Abonnez-vous à Amazon Nova Forge](#)
- [Configuration de l'infrastructure](#)
- [IA responsable](#)

Abonnez-vous à Amazon Nova Forge

Pour accéder aux fonctionnalités d'Amazon Nova Forge, procédez comme suit :

1. Vérifiez l'accès de l'administrateur au AWS compte.
2. Accédez à la console SageMaker AI et [demandez l'accès à Amazon Nova Forge](#).
3. Attendez que l'équipe Amazon Nova vous envoie un e-mail de confirmation une fois la demande d'abonnement approuvée.

4. Marquez votre rôle d'exécution avec le `forge-subscription` tag. Cette balise est requise pour accéder aux fonctionnalités et aux points de contrôle d'Amazon Nova Forge. Ajoutez la balise suivante à votre rôle d'exécution :

- Clé : `forge-subscription`
- Valeur : `true`

 Note

Les fonctionnalités standard d'Amazon Nova restent disponibles sans abonnement Forge. Amazon Nova Forge est conçu pour créer des modèles Frontier personnalisés avec contrôle et flexibilité pendant toutes les phases de formation des modèles.

Configuration de l'infrastructure

Une fois votre abonnement Amazon Nova Forge approuvé, configurez l'infrastructure nécessaire pour utiliser les fonctionnalités compatibles avec Forge. Pour obtenir des instructions détaillées sur la création d'un cluster EKS avec un groupe d'instances restreint (RIG), suivez les instructions de [l'atelier](#).

IA responsable

Paramètres de modération du contenu : les clients Amazon Nova Forge ont accès aux paramètres de modération du contenu (CCMS) personnalisables pour les modèles Amazon Nova Lite 1.0 et Pro 1.0. Le CCMS vous permet d'ajuster les contrôles de modération du contenu pour les aligner sur les exigences spécifiques de votre entreprise tout en maintenant les garanties essentielles et responsables liées à l'IA. Pour déterminer si votre cas d'utilisation professionnelle est éligible au CCMS, contactez votre responsable de compte Amazon Web Services.

Amazon Nova Forge fournit une boîte à outils d'intelligence artificielle responsable qui inclut des données d'entraînement, des critères d'évaluation et des contrôles d'exécution pour vous aider à aligner vos modèles sur les directives d'Amazon Nova en matière d'IA responsable.

Données de formation : La catégorie « RAI » du mixage de données contient des cas et des scénarios mettant l'accent sur les principes de l'IA responsable, les considérations de sécurité et le déploiement responsable des technologies. Utilisez-les pour aligner vos modèles de manière responsable pendant la période continue de pré-entraînement.

Évaluations : des tâches de référence sont disponibles pour tester la capacité de votre modèle à détecter et à rejeter le contenu inapproprié, préjudiciable ou incorrect. Utilisez ces évaluations pour mesurer la différence entre les performances du modèle de base et celles de votre modèle personnalisé.

Formation pour les modèles Amazon Nova

L'entraînement des modèles Amazon Nova SageMaker HyperPod prend en charge plusieurs techniques, notamment la pré-formation continue (CPT), le réglage fin supervisé (SFT) et le réglage fin par renforcement (RFT). Chaque technique répond à des besoins de personnalisation différents et peut être appliquée à différentes versions de modèles Amazon Nova.

Rubriques

- [Pré-entraînement continu \(CPT\)](#)
- [Peaufinage supervisé \(SFT\)](#)
- [Réglage précis du renforcement \(RFT\) activé SageMaker HyperPod](#)

Pré-entraînement continu (CPT)

La pré-formation continue (CPT) est une technique de formation qui prolonge la phase de pré-formation d'un modèle de base en l'exposant à du texte supplémentaire non étiqueté provenant de domaines ou de corpus spécifiques. Contrairement au réglage fin supervisé, qui nécessite des paires d'entrées-sorties étiquetées, le CPT s'entraîne sur des documents bruts pour aider le modèle à acquérir une connaissance plus approfondie de nouveaux domaines, à apprendre la terminologie et les modèles d'écriture spécifiques à un domaine, et à s'adapter à des types de contenu ou à des domaines particuliers.

Cette approche est particulièrement utile lorsque vous disposez de volumes importants (des dizaines de milliards de jetons) de données textuelles spécifiques à un domaine, telles que des documents juridiques, de la littérature médicale, de la documentation technique ou du contenu commercial propriétaire, et que vous souhaitez que le modèle développe une fluidité native dans ce domaine. Généralement, après la phase CPT, le modèle doit subir des étapes supplémentaires de réglage des instructions pour lui permettre d'utiliser les connaissances nouvellement acquises et d'effectuer des tâches utiles.

Modèles pris en charge

Le CPT est disponible pour les modèles Amazon Nova suivants :

- Nova 1.0 (Micro, Lite, Pro)
- Nova 2.0 (Lite)

Quand utiliser Nova 1.0 par rapport à Nova 2.0

La gamme de modèles Amazon Nova offre plusieurs points de fonctionnement en termes de rapport prix/performances afin d'optimiser la précision, la vitesse et le coût.

Choisissez Nova 2.0 lorsque vous avez besoin des éléments suivants :

- Capacités de raisonnement avancées pour les tâches analytiques complexes
- Performances supérieures en matière de codage, de mathématiques et de résolution de problèmes scientifiques
- Support d'une plus longue durée de contexte
- De meilleures performances multilingues

Note

Le modèle le plus grand n'est pas toujours meilleur. Tenez compte du compromis coût-performance et des exigences spécifiques de votre entreprise lorsque vous choisissez entre les modèles Nova 1.0 et Nova 2.0.

CPT sur Nova 2.0

Amazon Nova Lite 2.0 est un modèle de raisonnement conçu sur des ensembles de données plus volumineux et plus diversifiés que Nova Lite 1.0. Bien qu'il s'agisse d'un modèle plus grand, Nova Lite 2.0 permet une inférence plus rapide que Nova Lite 1.0 tout en offrant des capacités de raisonnement améliorées, des longueurs de contexte plus longues et des performances multilingues améliorées.

CPT on Nova 2.0 vous permet d'étendre ces fonctionnalités avancées avec les données spécifiques à votre domaine, ce qui permet au modèle de développer une expertise approfondie dans des domaines spécialisés tout en conservant ses capacités de raisonnement et d'analyse supérieures.

Exemple de recette CPT

Voici un exemple de recette pour le CPT. Vous pouvez trouver cette recette et d'autres dans le référentiel de [recettes](#).

```

# Note:
# This recipe can run on p5.48xlarge
# Run config
run:
  name: "my-cpt-run"                # A descriptive name for your training
  job
  model_type: "amazon.nova-2-lite-v1:0:256k" # Model variant specification, do not
  change
  model_name_or_path: "nova-lite-2/prod"      # Base model path, do not change
  replicas: 8                                # Number of compute instances for
  training, allowed values are 4, 8, 16, 32
  data_s3_path: ""                        # Customer data paths
  validation_data_s3_path: ""            # Customer validation data paths
  output_s3_path: ""                     # Output artifact path, job-specific
  configuration - not compatible with standard SageMaker Training Jobs
  mlflow_tracking_uri: ""                # Required for MLFlow
  mlflow_experiment_name: "my-cpt-experiment" # Optional for MLFlow. Note: leave this
  field non-empty
  mlflow_run_name: "my-cpt-run"          # Optional for MLFlow. Note: leave this
  field non-empty

## Training specific configs
training_config:
  task_type: cpt
  max_length: 8192                    # Maximum context window size (tokens)
  global_batch_size: 256              # Global batch size, allowed values are
  32, 64, 128, 256.

  trainer:
    max_steps: 10                     # The number of training steps to run
  total
    val_check_interval: 10            # The number of steps between running
  validation. Integer count or float percentage
    limit_val_batches: 2              # Batches of the validation set to use
  each trigger

  model:
    hidden_dropout: 0.0               # Dropout for hidden states, must be
  between 0.0 and 1.0
    attention_dropout: 0.0            # Dropout for attention weights, must
  be between 0.0 and 1.0

  optim:

```



```

optimizer: adam
lr: 1e-5                                # Learning rate
name: distributed_fused_adam            # Optimizer algorithm, do not change
adam_w_mode: true                       # Enable AdamW mode
eps: 1e-06                             # Epsilon for numerical stability
weight_decay: 0.0                      # L2 regularization strength, must be
between 0.0 and 1.0
adam_beta1: 0.9                        # Beta1 for Adam optimizer
adam_beta2: 0.95                       # Beta2 for Adam optimizer
sched:
  warmup_steps: 10                     # Learning rate warmup steps
  constant_steps: 0                    # Steps at constant learning rate
  min_lr: 1e-6                         # Minimum learning rate, must be lower
than lr

```

Préparation des données pour le CPT on 2.0

Exigences de format de données

Les ensembles de données de formation et de validation doivent être des fichiers JSONL au format illustré ci-dessous, où chaque ligne contient un objet JSON représentant une conversation avec les champs et la structure requis. Voici un exemple :

```

{"text": "AWS stands for Amazon Web Services"}
{"text": "Amazon SageMaker is a fully managed machine learning service"}
{"text": "Amazon Bedrock is a fully managed service for foundation models"}

```

Les entrées de texte doivent contenir un contenu naturellement fluide et de haute qualité qui représente le domaine cible.

Vérifiez que les données peuvent être converties [au format Arrow](#). Utilisez le script python ci-dessous pour vous aider. Assurez-vous que la datasets==2.18.0 version minimale est utilisée :

```

from datasets import load_dataset, load_from_disk
from pathlib import Path

input_path = Path("<Your jsonl file>")
output_path = Path("<Your output directory>")

dataset = load_dataset("json", data_files=str(input_path), split="train")
dataset.save_to_disk(str(output_path), max_shard_size="1GB")

```

```
try:
    test_dataset = datasets.load_from_disk(output_dir)
    print(f"Dataset loaded successfully #! Contains {len(test_dataset)} samples")
except Exception as e:
    print(e)
```

Il devrait imprimer le même nombre de lignes que dans le fichier JSONL.

Lorsque vous utilisez le mixage de données, exécutez la première tâche avec `max_steps=2`. Cela permettra de créer des optimisations dans le cluster pour l'accès aux données et de vérifier que tous les mélanges de données sont disponibles.

Comment préparer les données pour le CPT

Les données d'entraînement sont le facteur déterminant le plus crucial pour le succès de la pré-formation continue. Bien que les données du CPT soient souvent décrites comme « non étiquetées », la réalité est bien plus nuancée. La manière dont les données sont structurées, formatées et présentées détermine si le modèle va acquérir les connaissances et les compétences requises pour le cas d'utilisation métier.

Préparation d'ensembles de données métiers structurés pour le CPT

Il s'agit d'un défi courant pour les entreprises et les organisations qui élaborent des modèles de base spécialisés dans leur domaine. La plupart des entreprises possèdent de riches référentiels de données structurées : catalogues de produits, profils utilisateurs, journaux de transactions, soumissions de formulaires, appels d'API et métadonnées opérationnelles. À première vue, cela semble très différent du texte Web non structuré généralement utilisé dans le cadre de la pré-formation standard.

Pour tirer des leçons efficaces des données commerciales structurées, réfléchissez bien aux tâches en aval et concevez la présentation des données de manière à obliger le modèle à apprendre les bonnes relations prédictives.

Pour exploiter tout le potentiel de la pré-formation continue, pensez à :

- Quelles tâches le modèle doit effectuer au moment de l'inférence
- Quelles sont les informations présentes dans les données brutes
- Comment structurer ces données afin que le modèle apprenne à les extraire et à les manipuler correctement

Le simple fait de transférer des données structurées dans la formation n'apprendra pas au modèle à raisonner à ce sujet. Donnez une forme active à la présentation des données afin de guider ce que le modèle apprend.

Dans les sections suivantes, une revue de la littérature démontre l'importance de l'augmentation des données et fournit des exemples de stratégies d'augmentation pour les données commerciales structurées qui donneront des idées utiles sur la façon de traiter et d'organiser un ensemble de données commerciales pour le CPT.

Données structurées pour le CPT dans la littérature

Le CPT peut intégrer les faits du domaine dans le modèle, mais échoue souvent à les rendre récupérables et manipulables lorsque les entrées ou les tâches changent. Des expériences contrôlées montrent que, sans augmentation variable pendant le préentraînement, les modèles mémorisent des faits de manière fragile qui restent difficiles à extraire même après un réglage ultérieur des instructions, et ils recommandent d'injecter des instructions sous forme de signaux au début de l'entraînement. Pour les données semi-structurées, la sérialisation aléatoire et d'autres augmentations réduisent le surajustement du schéma. C'est pourquoi le CPT doit être entrelacé avec des tâches de type instruction plutôt que d'être exécuté d'abord et IFT plus tard. Des travaux axés sur la finance révèlent en outre que le mélange conjoint des données CPT et des données d'instructions par lots améliore la généralisation et réduit l'oubli par rapport à la recette séquentielle. Le rapport technique de Qwen converge sur le même schéma en intégrant des données d'enseignement de haute qualité dans la préformation elle-même, ce qui stimule l'apprentissage contextuel et préserve le suivi des instructions tout en acquérant de nouvelles connaissances dans le domaine.

L'augmentation des données pour les corpus semi-structurés est un levier clé. Le CPT sensible aux graphes synthétiques étend les petits ensembles de domaines en corpus liés à des entités qui enseignent explicitement les relations et les composés en les extrayant au moment de l'inférence. Le mixage conjoint du CPT et des instructions surpasse les pipelines séquentiels dans le domaine de la finance et l'équilibre entre le domaine des données générales réduit la dégradation des compétences générales. Le CPT dans un domaine à très grande échelle peut également conserver une large capacité et même permettre des compromis grâce à la fusion de modèles, tout en indiquant que le réglage des instructions est la prochaine étape essentielle, renforçant ainsi l'intérêt de l'introduction de signaux d'instructions pendant le CPT.

Insuffler de la diversité par le biais de la randomisation et du remaniement

Une stratégie générale qui permet d'enseigner efficacement le modèle à partir des ensembles de données structurés et semi-structurés consiste à modifier l'ordre des champs dans les ensembles de données, voire à supprimer certaines clés de manière aléatoire.

Le remaniement des champs oblige le modèle à lire la signification de chaque valeur plutôt que l'endroit où elle apparaît et à apprendre les relations entre tous les champs. Par exemple, dans le cas d'un jeu vidéo publié sur Amazon Store, lorsque « Titre », « Plateforme », « Prix », « État » et « Édition » arrivent sous différentes formes, le modèle ne peut pas se fier au « troisième emplacement est la plateforme » ; il doit lier les étiquettes aux valeurs et connaître les relations bilatérales entre les attributs : titre, plateforme, prix, condition, prix. Ainsi, il peut, par exemple, déduire une plateforme probable à partir du nom d'un jeu et d'un prix observé, ou estimer une fourchette de prix plausible en fonction d'un titre et d'une plateforme.

La suppression aléatoire de clés lors de la sérialisation agit comme une suppression de fonctionnalités : elle empêche la co-adaptation sur un champ donné et oblige le modèle à récupérer les informations manquantes à partir des preuves restantes. Si le terme « Plateforme » est absent, le modèle doit le trouver dans la chaîne de titre ou dans le texte de compatibilité ; si le terme « Prix » est masqué, il doit être triangulé en fonction de la plate-forme, de l'édition et de l'état. Cela renforce la symétrie ($A \rightarrow B$ et $B \rightarrow A$), la robustesse face aux listes compliquées du monde réel et l'invariance du schéma lorsque des champs sont manquants, renommés ou réorganisés.

Un exemple de style shopping le concrétise. Sérialisez le même article de plusieurs manières : « Titre : 'Elden Ring' | Plateforme : PlayStation 5 | État : usagé, comme neuf | Prix : 34,99\$ » et une permutation telle que « Prix : 34,99\$ | Titre : 'Elden Ring' | État : usagé, comme neuf | Plateforme : 5 ». Parfois, supprimez « Plateforme » tout en laissant « Compatible avec » dans la description. PlayStation PS5 Définissez des objectifs complémentaires tels que la prévision de la plateforme à partir de {title, price} et la prédiction d'une tranche de prix à partir de {title, platform}. Étant donné que l'ordre et même la présence des clés varient, la seule stratégie stable consiste à apprendre les véritables relations entre les attributs plutôt que de mémoriser un modèle.

La façon dont les données sont présentées est importante

LLMs apprenez en prédisant le jeton suivant à partir de ce qu'ils ont déjà vu. Ainsi, l'ordre des champs et des événements affichés pendant l'entraînement détermine ce que le modèle peut apprendre. Si le format d'entraînement correspond à la tâche réelle, la perte se répercute sur les jetons de décision exacts. Si les champs sont mélangés sans structure, le modèle apprend les raccourcis ou mémorise la popularité, puis échoue lorsqu'on lui demande de choisir parmi les options.

Montrez d'abord la situation, puis les options, puis la décision. Si le modèle doit également en apprendre davantage sur les résultats ou les explications, placez-les après la décision.

Échantillons d'emballage pour CPT

Qu'est-ce que l'emballage ?

Cela signifie simplement de remplir chaque fenêtre de séquence des données d'entraînement avec plusieurs exemples complets afin que la fenêtre soit dense avec de vrais jetons, et non avec du remplissage.

Pourquoi est-ce important ?

Pendant l'entraînement, une longueur de contexte maximale est définie, par exemple 8 192 jetons. Les lots sont mis en forme selon [taille du lot × longueur du contexte]. Si un exemple d'entraînement est plus court que la durée du contexte, les positions restantes sont rembourrées. Le rembourrage passe toujours par l'attention et les noyaux MLP, même si la perte est masquée. Le calcul est donc payé pour des jetons qui ne transmettent aucun signal d'apprentissage.

Comment emballer ?

Pour regrouper plusieurs échantillons, concaténez plusieurs échantillons d'apprentissage avec un [DOC] séparateur entre les deux (notez l'espace avant et après le [DOC]) de telle sorte que la longueur totale des échantillons soit inférieure à la longueur de contexte souhaitée.

Voici un exemple de document compressé :

```
{"text": "training sample 1 [DOC] training sample 2 [DOC] training sample 3"}
```

Paramètres de réglage CPT

Les paramètres disponibles pour un réglage précis avec le CPT sont les suivants :

Configuration d'exécution

- **nom** : nom descriptif de votre poste de formation. Cela permet d'identifier votre tâche dans la console AWS de gestion.
- **model_type** : variante du modèle Amazon Nova à utiliser. Les options disponibles sont les `amazon.nova-2-lite-v1:0:256k` suivantes :

- `model_name_or_path` : chemin vers le modèle de base à utiliser pour votre entraînement. Les options disponibles sont `nova-lite-2/prod`, ou le chemin S3 pour le point de contrôle post-entraînement (`s3://customer-escrow-bucket-unique_id/training_run_name`).
- `répliques` : nombre d'instances de calcul à utiliser pour la formation distribuée. Les valeurs disponibles varient en fonction du modèle que vous choisissez. Amazon Nova Lite 2.0 prend en charge 4, 8, 16 ou 32 répliques.
- `data_s3_path` : emplacement S3 du jeu de données d'entraînement, qui est un fichier JSONL. Ce fichier doit résider dans le même AWS compte et dans la même région que le cluster. Tous les emplacements S3 fournis doivent se trouver dans le même compte et la même région.
- `validation_data_s3_path` : (Facultatif) L'emplacement S3 du jeu de données de validation, qui est un fichier JSONL. Ce fichier doit résider dans le même compte et la même région que le cluster. Tous les emplacements S3 fournis doivent se trouver dans le même compte et la même région.
- `output_s3_path` : emplacement S3 où le manifeste et les journaux sont stockés. TensorBoard Tous les emplacements S3 fournis doivent se trouver dans le même AWS compte et dans la même AWS région.
- `mlflow_tracking_uri` : l'ARN de l'application à utiliser pour la journalisation MLFlow MLFlow
- `mlflow_experiment_name` : nom de l'expérience MLFlow
- `mlflow_run_name` : nom de l'exécution MLFlow

Configuration de la formation

- `max_length` : longueur de séquence maximale en jetons. Détermine la taille de la fenêtre contextuelle pour l'entraînement. La valeur maximale prise en charge est de 8 192 jetons pour le CPT.

Des séquences plus longues amélioreront l'efficacité de l'entraînement au prix d'une augmentation des exigences de mémoire. Nous vous recommandons de faire correspondre le paramètre `max_length` à la distribution de vos données.

- `global_batch_size` : nombre total d'échantillons d'entraînement traités ensemble en un seul passage en avant ou en arrière sur tous les appareils et tous les travailleurs.

Cette valeur multiplie la taille du lot par appareil et le nombre d'appareils. Cela affecte la stabilité de l'entraînement et le débit. Nous vous recommandons de commencer par une taille de lot adaptée à la mémoire, puis d'augmenter verticalement à partir de là. Pour les données spécifiques à un domaine, des lots plus importants peuvent trop lisser les gradients.

Paramètres de l'entraîneur

- `max_steps` : le nombre d'étapes d'entraînement à exécuter. Chaque étape entraînera le modèle avec le `global_batch_size` nombre d'éléments

Réglages du modèle

- `hidden_dropout` : probabilité de supprimer les sorties d'état masquées. Augmentez cette valeur d'environ 0,0-0,2 pour réduire les surajustements sur de plus petits jeux de données. Les valeurs valides sont comprises entre 0 et 1 inclus.
- `attention_dropout` : probabilité de perdre du poids d'attention. Ce paramètre peut faciliter la généralisation. Les valeurs valides sont comprises entre 0 et 1 inclus.

Configuration de l'optimiseur

- `lr` : le taux d'apprentissage, qui contrôle la taille des étapes lors de l'optimisation. Nous recommandons des valeurs comprises entre 1e-6 et 1e-4 pour de bonnes performances. Les valeurs valides sont comprises entre 0 et 1 inclus.
- `nom` : L'algorithme de l'optimiseur. Actuellement, seul `distributed_fused_adam` est pris en charge.
- `weight_decay` : La force de régularisation L2. Des valeurs plus élevées (entre 0,01 et 0,1) augmentent la régularisation.
- `warmup_steps` : nombre d'étapes pour augmenter progressivement le taux d'apprentissage. Cela améliore la stabilité de l'entraînement. Les valeurs valides sont comprises entre 1 et 20 inclus.
- `min_lr` : Le taux d'apprentissage minimum à la fin de la décadence. Les valeurs valides sont comprises entre 0 et 1 inclus, mais doivent être inférieures au taux d'apprentissage.

Peaufinage supervisé (SFT)

Le processus de formation SFT comprend deux étapes principales :

- Préparation des données : suivez les directives établies pour créer, nettoyer ou reformater les ensembles de données selon la structure requise. Assurez-vous que les entrées, les sorties et

les informations auxiliaires (telles que les traces de raisonnement ou les métadonnées) sont correctement alignées et formatées.

- Configuration de l'entraînement : définissez la manière dont le modèle sera entraîné. Lors de l'utilisation, cette configuration est écrite dans un fichier de recette YAML qui inclut :
 - Chemins des sources de données (ensembles de données de formation et de validation)
 - Hyperparamètres clés (époches, taux d'apprentissage, taille du lot)
 - Composants optionnels (paramètres d'entraînement distribués, etc.)

Comparaison et sélection des modèles Nova

Amazon Nova 2.0 est un modèle formé sur un ensemble de données plus vaste et plus diversifié qu'Amazon Nova 1.0. Les améliorations clés comprennent les suivantes :

- Capacités de raisonnement améliorées avec prise en charge du mode de raisonnement explicite
- Performances multilingues étendues dans d'autres langues
- Performances améliorées pour les tâches complexes, notamment le codage et l'utilisation des outils
- Gestion étendue du contexte avec une précision et une stabilité accrues à des longueurs de contexte plus longues

Quand utiliser Nova 1.0 ou Nova 2.0

Choisissez Amazon Nova 2.0 lorsque :

- Des performances supérieures associées à des capacités de raisonnement avancées sont nécessaires
- Un support multilingue ou une gestion de tâches complexes sont nécessaires
- De meilleurs résultats en matière de codage, d'appel d'outils ou de tâches analytiques sont nécessaires

SFT sur Nova 2.0

Amazon Nova Lite 2.0 apporte des fonctionnalités améliorées pour un réglage précis supervisé, notamment un mode de raisonnement avancé, une meilleure compréhension multimodale et une gestion étendue du contexte. SFT on Nova 2.0 vous permet d'adapter ces puissantes fonctionnalités

à vos cas d'utilisation spécifiques tout en maintenant les performances supérieures du modèle pour les tâches complexes.

Les principales fonctionnalités de SFT on Nova 2.0 incluent :

- Support du mode raisonnement : entraînez des modèles pour générer des traces de raisonnement explicites avant les réponses finales pour des capacités analytiques améliorées.
- Formation multimodale avancée : affinez la compréhension des documents (PDF), la compréhension vidéo et les tâches basées sur des images avec une précision accrue.
- Capacités d'appel d'outils : entraînez les modèles à utiliser efficacement des outils externes et des fonctions nécessitant des flux de travail complexes.
- Support contextuel étendu : tirez parti de fenêtres contextuelles plus longues avec une stabilité et une précision accrues pour les applications gourmandes en documents.

 Note

Pour plus d'informations sur les images de conteneurs ou les exemples de recettes à utiliser, consultez les [recettes Amazon Nova](#).

Rubriques

- [Sélection du mode de raisonnement \(Nova 2.0 uniquement\)](#)
- [Format des données d'appel de l'outil](#)
- [Document comprenant le format des données](#)
- [Compréhension vidéo pour SFT](#)
- [Instructions de téléchargement des données](#)
- [Création d'un Job de peaufinage](#)
- [Paramètres de réglage SFT](#)
- [Guidage des hyperparamètres](#)

Exemple de recette SFT

Vous trouverez ci-dessous un exemple de recette pour SFT. Vous pouvez trouver cette recette et d'autres dans le référentiel de [recettes](#).

```

run:
  name: my-full-rank-sft-run
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: nova-lite-2/prod
  data_s3_path: s3://my-bucket-name/train.jsonl # only and not compatible with
SageMaker Training Jobs
  replicas: 4 # Number of compute instances for
training, allowed values are 4, 8, 16, 32
  output_s3_path: s3://my-bucket-name/outputs/ # Output artifact path (HyperPod job-
specific; not compatible with standard SageMaker Training Jobs)
  mlflow_tracking_uri: "" # Required for MLFlow
  mlflow_experiment_name: "my-full-rank-sft-experiment" # Optional for MLFlow. Note:
leave this field non-empty
  mlflow_run_name: "my-full-rank-sft-run" # Optional for MLFlow. Note: leave
this field non-empty

training_config:
  max_steps: 100 # Maximum training steps. Minimal is 4.
  save_steps: ${oc.select:training_config.max_steps} # How many training steps the
checkpoint will be saved
  save_top_k: 5 # Keep top K best checkpoints. Note supported only
for jobs. Minimal is 1.
  max_length: 32768 # Sequence length (options: 8192, 16384, 32768
[default], 65536)
  global_batch_size: 32 # Global batch size (options: 32, 64, 128)
  reasoning_enabled: true # If data has reasoningContent, set to true;
otherwise False

  lr_scheduler:
    warmup_steps: 15 # Learning rate warmup steps. Recommend 15% of
max_steps
    min_lr: 1e-6 # Minimum learning rate, must be between 0.0 and
1.0

  optim_config: # Optimizer settings
    lr: 1e-5 # Learning rate, must be between 0.0 and 1.0
    weight_decay: 0.0 # L2 regularization strength, must be between 0.0
and 1.0
    adam_beta1: 0.9 # Exponential decay rate for first-moment
estimates
    adam_beta2: 0.95 # Exponential decay rate for second-moment
estimates

```

```
peft:                                     # Parameter-efficient fine-tuning (LoRA)
  peft_scheme: "null"                     # Disable LoRA for PEFT
```

Sélection du mode de raisonnement (Nova 2.0 uniquement)

Amazon Nova 2.0 prend en charge le mode raisonnement pour des capacités analytiques améliorées :

- Mode de raisonnement (activé) :
 - `reasoning_enabled: true` Paramétré dans la configuration d'entraînement
 - Modélisez des trains pour générer des traces de raisonnement avant les réponses finales
 - Améliore les performances lors de tâches de raisonnement complexes
- Mode sans raisonnement (désactivé) :
 - Définir `reasoning_enabled: false` ou omettre le paramètre (par défaut)
 - SFT standard sans raisonnement explicite
 - Convient aux tâches qui ne bénéficient pas du step-by-step raisonnement

Note

- Lorsque le raisonnement est activé, il fonctionne avec un effort de raisonnement élevé. Il n'existe pas d'option de faible raisonnement pour SFT.
- Le contenu du raisonnement multimodal n'est pas pris en charge pour SFT. Le mode de raisonnement s'applique aux entrées contenant uniquement du texte.

Utilisation du mode raisonnement avec des ensembles de données non raisonnés

L'entraînement d'Amazon Nova sur un ensemble de données non raisonné avec `reasoning_enabled: true` est autorisé. Cela risque toutefois de faire perdre au modèle ses capacités de raisonnement, car Amazon Nova apprend principalement à générer les réponses présentées dans les données sans appliquer de raisonnement.

Si vous entraînez Amazon Nova sur un ensemble de données non raisonné mais que vous souhaitez tout de même utiliser le raisonnement lors de l'inférence :

1. Désactiver le raisonnement pendant l'entraînement (`reasoning_enabled: false`)

2. Permettre le raisonnement ultérieurement lors de l'inférence

Bien que cette approche permette de raisonner au moment de l'inférence, elle ne garantit pas une amélioration des performances par rapport à l'inférence sans raisonnement.

Bonne pratique : activez le raisonnement à la fois pour l'entraînement et pour l'inférence lorsque vous utilisez des ensembles de données de raisonnement, et désactivez-le pour les deux lorsque vous utilisez des ensembles de données sans raisonnement.

Note

Pour plus d'informations sur les images de conteneurs ou les exemples de recettes à utiliser, consultez les [recettes Amazon Nova](#).

Format des données d'appel de l'outil

SFT prend en charge les modèles de formation à l'utilisation des outils (appel de fonctions). Vous trouverez ci-dessous un exemple de format de saisie pour l'appel d'outils :

Exemple de saisie :

```
{
  "schemaVersion": "bedrock-conversation-2024",
  "system": [
    {
      "text": "You are an expert in composing function calls."
    }
  ],
  "toolConfig": {
    "tools": [
      {
        "toolSpec": {
          "name": "getItemCost",
          "description": "Retrieve the cost of an item from the catalog",
          "inputSchema": {
            "json": {
              "type": "object",
              "properties": {
                "item_name": {
                  "type": "string",
                  "description": "The name of the item to retrieve cost for"
                }
              }
            }
          }
        }
      }
    ]
  }
}
```

```

        },
        "item_id": {
            "type": "string",
            "description": "The ASIN of item to retrieve cost for"
        }
    },
    "required": [
        "item_id"
    ]
}
}
},
{
    "toolSpec": {
        "name": "getItemAvailability",
        "description": "Retrieve whether an item is available in a given location",
        "inputSchema": {
            "json": {
                "type": "object",
                "properties": {
                    "zipcode": {
                        "type": "string",
                        "description": "The zipcode of the location to check in"
                    },
                    "quantity": {
                        "type": "integer",
                        "description": "The number of items to check availability for"
                    },
                    "item_id": {
                        "type": "string",
                        "description": "The ASIN of item to check availability for"
                    }
                }
            },
            "required": [
                "item_id", "zipcode"
            ]
        }
    }
}
}
],
"messages": [

```

```

{
  "role": "user",
  "content": [
    {
      "text": "I need to check whether there are twenty pieces of the following
item available. Here is the item ASIN on Amazon: id-123. Please check for the zipcode
94086"
    }
  ]
},
{
  "role": "assistant",
  "content": [
    {
      "reasoningContent": {
        "reasoningText": {
          "text": "The user wants to check how many pieces of the item with ASIN
id-123 are available in the zipcode 94086"
        }
      }
    },
    {
      "toolUse": {
        "toolUseId": "getItemAvailability_0",
        "name": "getItemAvailability",
        "input": {
          "zipcode": "94086",
          "quantity": 20,
          "item_id": "id-123"
        }
      }
    }
  ]
},
{
  "role": "user",
  "content": [
    {
      "toolResult": {
        "toolUseId": "getItemAvailability_0",
        "content": [
          {
            "text": "[{"name": "getItemAvailability", "results":
{"availability": true}]]"}
          ]
        }
      }
    ]
  ]
}

```

```

    }
  ]
}
},
{
  "role": "assistant",
  "content": [
    {
      "text": "Yes, there are twenty pieces of item id-123 available at 94086.
Would you like to place an order or know the total cost?"
    }
  ]
}
]
}
}

```

Considérations importantes concernant les données relatives aux appels d'outils :

- ToolUse doit apparaître uniquement dans les virages d'assistant
- ToolResult doit apparaître uniquement au tour de l'utilisateur
- ToolResult doit être uniquement en texte ou en JSON ; les autres modalités ne sont actuellement pas prises en charge pour les modèles Amazon Nova
- L'InputSchema du ToolSpec doit être un objet de schéma JSON valide
- Chacun ToolResult doit faire référence toolUsed à un assistant précédent valide ToolUse, toolUsed utilisé exactement une fois par conversation

Note

Pour plus d'informations sur les images de conteneurs ou les exemples de recettes à utiliser, consultez les [recettes Amazon Nova](#).

Document comprenant le format des données

SFT prend en charge les modèles de formation sur les tâches de compréhension des documents. Vous trouverez ci-dessous un exemple de format de saisie :

Exemple d'entrée

```
{
  "schemaVersion": "bedrock-conversation-2024",
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "text": "What are the ways in which a customer can experience issues during checkout on Amazon?"
        },
        {
          "document": {
            "format": "pdf",
            "source": {
              "s3Location": {
                "uri": "s3://my-bucket-name/path/to/documents/customer_service_debugging.pdf",
                "bucketOwner": "123456789012"
              }
            }
          }
        }
      ]
    },
    {
      "role": "assistant",
      "content": [
        {
          "text": "Customers can experience issues with 1. Data entry, 2. Payment methods, 3. Connectivity while placing the order. Which one would you like to dive into?"
        }
      ],
      "reasoning_content": [
        {
          "text": "I need to find the relevant section in the document to answer the question.",
          "type": "text"
        }
      ]
    }
  ]
}
```


Considérations importantes pour la compréhension des documents :

- Seuls les fichiers PDF sont pris en charge
- La taille maximale du document est de 10 Mo
- Un échantillon peut contenir des documents et du texte, mais ne peut pas mélanger des documents avec d'autres modalités (telles que des images ou des vidéos)

Note

Pour plus d'informations sur les images de conteneurs ou les exemples de recettes à utiliser, consultez les [recettes Amazon Nova](#).

Compréhension vidéo pour SFT

SFT prend en charge le réglage précis des modèles pour les tâches de compréhension vidéo. Vous trouverez ci-dessous un exemple de format de saisie :

Exemple d'entrée

```
{
  "schemaVersion": "bedrock-conversation-2024",
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "text": "What are the ways in which a customer can experience issues during checkout on Amazon?"
        },
        {
          "video": {
            "format": "mp4",
            "source": {
              "s3Location": {
                "uri": "s3://my-bucket-name/path/to/videos/customer_service_debugging.mp4",
                "bucketOwner": "123456789012"
              }
            }
          }
        }
      ]
    }
  ]
}
```

```
    }
  }
]
},
{
  "role": "assistant",
  "content": [
    {
      "text": "Customers can experience issues with 1. Data entry, 2. Payment methods, 3. Connectivity while placing the order. Which one would you like to dive into?"
    }
  ],
  "reasoning_content": [
    {
      "text": "I need to find the relevant section in the video to answer the question.",
      "type": "text"
    }
  ]
}
]
```

Considérations importantes pour la compréhension de la vidéo :

- Les vidéos peuvent atteindre un maximum de 50 Mo
- Les vidéos peuvent durer jusqu'à 15 minutes
- Une seule vidéo est autorisée par échantillon ; les vidéos multiples dans le même échantillon ne sont pas prises en charge
- Un échantillon peut contenir de la vidéo et du texte, mais ne peut pas mélanger la vidéo avec d'autres modalités (telles que des images ou des documents)

 Note

Pour plus d'informations sur les images de conteneurs ou les exemples de recettes à utiliser, consultez les [recettes Amazon Nova](#).

Instructions de téléchargement des données

Téléchargez des ensembles de données de formation et de validation dans un compartiment S3. Spécifiez les emplacements suivants dans le `run` bloc de la recette :

```
## Run config
run:
  ...
  data_s3_path: "s3://<bucket-name>/<training-directory>/<training-file>.jsonl"
```

Remarque : remplacez `<bucket-name>`, `<training-directory>`, `<validation-directory>` `<training-file>`, et `<validation-file>` par des chemins S3 réels.

Remarque : Les ensembles de données de validation ne sont actuellement pas pris en charge pour SFT avec Amazon Nova 2.0. Si un ensemble de données de validation est fourni, il sera ignoré.

Création d'un Job de peaufinage

Définissez le modèle de base à l'aide `model_name_or_path` des champs `model_type` et du `run` bloc :

```
## Run config
run:
  ...
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: nova-lite-2/prod
  ...
```

Paramètres de réglage SFT

Les paramètres disponibles pour le réglage avec SFT sont les suivants :

Configuration d'exécution

- `nom` : nom descriptif de votre poste de formation. Cela permet d'identifier votre tâche dans la console AWS de gestion.
- `model_type` : variante du modèle Amazon Nova à utiliser. Les options disponibles sont les `amazon.nova-2-lite-v1:0:256k` suivantes :

- `model_name_or_path` : chemin vers le modèle de base à utiliser pour votre entraînement. Les options disponibles sont `nova-lite-2/prod`, ou le chemin S3 pour le point de contrôle post-entraînement (`s3://customer-escrow-bucket-unique_id/training_run_name`).
- `répliques` : nombre d'instances de calcul à utiliser pour la formation distribuée. Les valeurs disponibles varient en fonction du modèle que vous choisissez. Amazon Nova Lite 2.0 prend en charge 4, 8, 16 ou 32 répliques.
- `data_s3_path` : emplacement S3 du jeu de données d'entraînement, qui est un fichier JSONL. Ce fichier doit résider dans le même AWS compte et dans la même région que le cluster. Tous les emplacements S3 fournis doivent se trouver dans le même compte et la même région.
- `validation_data_s3_path` : (Facultatif) L'emplacement S3 du jeu de données de validation, qui est un fichier JSONL. Ce fichier doit résider dans le même compte et la même région que le cluster. Tous les emplacements S3 fournis doivent se trouver dans le même compte et la même région.
- `output_s3_path` : emplacement S3 où le manifeste et les journaux sont stockés. TensorBoard Tous les emplacements S3 fournis doivent se trouver dans le même AWS compte et dans la même AWS région.
- `mlflow_tracking_uri` : l'ARN de l'application à utiliser pour MLFlow la journalisation. MLFlow
- `mlflow_experiment_name` : nom de l'expérience. MLFlow
- `mlflow_run_name` : nom de l'exécution. MLFlow

Configuration de la formation

- `max_steps` : le nombre d'étapes d'entraînement à exécuter. Chaque étape entraînera le modèle avec un `global_batch_size` certain nombre d'éléments.
- `save_steps` : fréquence (en étapes) à laquelle enregistrer les points de contrôle du modèle pendant l'entraînement.
- `save_top_k` : Le nombre maximum de meilleurs points de contrôle à conserver en fonction des métriques de validation.
- `max_length` : longueur de séquence maximale en jetons. Détermine la taille de la fenêtre contextuelle pour l'entraînement. La valeur maximale prise en charge est de 32 768 jetons pour SFT.

Des séquences plus longues amélioreront l'efficacité de l'entraînement au prix d'une augmentation des exigences de mémoire. Nous vous recommandons de faire correspondre le paramètre `max_length` à la distribution de vos données.

- `global_batch_size` : nombre total d'échantillons d'entraînement traités ensemble en un seul passage en avant ou en arrière sur tous les appareils et tous les travailleurs.

Cette valeur multiplie la taille du lot par appareil et le nombre d'appareils. Cela affecte la stabilité de l'entraînement et le débit. Nous vous recommandons de commencer par une taille de lot adaptée à la mémoire, puis d'augmenter verticalement à partir de là. Pour les données spécifiques à un domaine, des lots plus importants peuvent trop lisser les gradients.

- `reasoning_enabled` : indicateur booléen pour activer les capacités de raisonnement pendant l'entraînement.

Planificateur de taux d'apprentissage

- `warmup_steps` : nombre d'étapes pour augmenter progressivement le taux d'apprentissage. Cela améliore la stabilité de l'entraînement.
- `min_lr` : Le taux d'apprentissage minimum à la fin de la décadence. Les valeurs valides sont comprises entre 0 et 1 inclus, mais doivent être inférieures au taux d'apprentissage.

Configuration de l'optimiseur

- `lr` : le taux d'apprentissage, qui contrôle la taille des étapes lors de l'optimisation. Nous recommandons des valeurs comprises entre 1e-6 et 1e-4 pour de bonnes performances. Les valeurs valides sont comprises entre 0 et 1 inclus.
- `weight_decay` : La force de régularisation L2. Des valeurs plus élevées (entre 0,01 et 0,1) augmentent la régularisation.
- `adam_beta1` : Le taux de décroissance exponentiel pour les premières estimations dans l'optimiseur Adam. La valeur par défaut est 0,9.
- `adam_beta2` : Le taux de décroissance exponentiel pour le deuxième instant est estimé dans l'optimiseur Adam. La valeur par défaut est 0,95.

Configuration PEFT

- `peft_scheme` : schéma de réglage fin efficace en termes de paramètres à utiliser. Les options concernent 'null' un réglage précis du rang complet ou un réglage précis basé sur `lora` LoRA.

Réglage LoRa (lorsque `peft_scheme` est « lora »)

- `alpha` : paramètre de mise à l'échelle LoRa. Contrôle l'amplitude de l'adaptation de rang inférieur. Les valeurs typiques sont comprises entre 8 et 128.
- `lora_plus_lr_ratio` : taux d'apprentissage pour l'optimisation LoRa+. Ce multiplicateur ajuste le taux d'apprentissage spécifiquement pour les paramètres LoRa.

Guidage des hyperparamètres

Utilisez les hyperparamètres recommandés suivants en fonction de l'approche d'entraînement :

Entraînement complet

- Époques : 1
- Taux d'apprentissage (`lr`) : 1e-5
- Taux d'apprentissage minimal (`min_lr`) : 1e-6

LoRa (adaptation de bas rang)

- Époques : 2
- Taux d'apprentissage (`lr`) : 5e-5
- Taux d'apprentissage minimal (`min_lr`) : 1e-6

Remarque : ajustez ces valeurs en fonction de la taille du jeu de données et des performances de validation. Surveillez les indicateurs d'entraînement pour éviter le surajustement.

Réglage précis du renforcement (RFT) activé SageMaker HyperPod

Le réglage fin par renforcement (RFT) est une technique d'apprentissage automatique qui améliore les performances des modèles grâce à des signaux de feedback (scores mesurables ou récompenses indiquant la qualité des réponses) plutôt qu'à une supervision directe avec des réponses exactes. Contrairement au réglage fin supervisé traditionnel qui apprend à partir de paires d'entrées-sorties, le RFT utilise des fonctions de récompense pour évaluer les réponses du modèle et optimise le modèle de manière itérative afin de maximiser ces récompenses.

Cette approche est particulièrement efficace pour les tâches où il est difficile de définir le résultat exact correct, mais vous pouvez mesurer de manière fiable la qualité de réponse. Le RFT permet aux

modèles d'apprendre des comportements et des préférences complexes par le biais d'essais et de commentaires, ce qui le rend idéal pour les applications nécessitant une prise de décision nuancée, une résolution créative des problèmes ou le respect de critères de qualité spécifiques pouvant être évalués par programmation.

Quand utiliser RFT

Utilisez la RFT lorsque vous pouvez définir des critères de réussite clairs et mesurables, mais que vous avez du mal à fournir des résultats exacts pour la formation. Il est idéal pour les tâches où la qualité est subjective ou multidimensionnelle, telles que l'écriture créative, l'optimisation du code ou le raisonnement complexe, pour lesquelles plusieurs solutions valides existent, mais certaines sont clairement meilleures que d'autres.

Le RFT fonctionne mieux lorsque vous disposez des éléments suivants :

- Une fonction de récompense fiable qui peut évaluer les résultats du modèle de manière programmatique
- Nécessité d'aligner le comportement du modèle sur des préférences ou des contraintes spécifiques
- Situations dans lesquelles le réglage minutieux supervisé traditionnel ne suffit pas parce que la collecte d'exemples étiquetés de haute qualité est coûteuse ou peu pratique

Envisagez le RFT pour les applications nécessitant une amélioration itérative, une personnalisation ou le respect de règles métier complexes pouvant être codées sous forme de signaux de récompense.

À quoi convient le mieux le RFT

La RFT excelle dans les domaines où la qualité de sortie peut être mesurée objectivement, mais où les réponses optimales sont difficiles à définir dès le départ :

- Résolution de problèmes mathématiques : exactitude vérifiable avec plusieurs voies de résolution
- Génération et optimisation du code : résultats d'exécution et indicateurs de performance testables
- Tâches de raisonnement scientifique : cohérence logique et exactitude factuelle
- Analyse de données structurées : résultats vérifiables par programmation
- Raisonnement en plusieurs étapes : tâches nécessitant une progression step-by-step logique
- Utilisation des outils et appels d'API : succès mesurable par les résultats d'exécution
- Flux de travail complexes : respect de contraintes et de règles commerciales spécifiques

La RFT fonctionne exceptionnellement bien lorsque vous devez équilibrer plusieurs objectifs concurrents tels que la précision, l'efficacité et le style.

Quand utiliser le mode raisonnement pour l'entraînement RFT

Amazon Nova 2.0 prend en charge le mode raisonnement pendant l'entraînement RFT. Les modes suivants sont disponibles :

- none : aucun raisonnement (omettez le champ `reasoning_effort`)
- faible : surcharge de raisonnement minimale
- high : capacité de raisonnement maximale (par défaut lorsque `reasoning_effort` est spécifié)

 Note

Il n'existe pas d'option moyenne pour le RFT. Si le champ `reasoning_effort` est absent de votre configuration, le raisonnement est désactivé.

Utilisez un raisonnement raisonné pour ce qui suit :

- Tâches analytiques complexes
- Résolution de problèmes mathématiques
- Déduction logique en plusieurs étapes
- Tâches où la step-by-step réflexion apporte une valeur ajoutée

Utilisez un raisonnement nul (omettez `reasoning_effort`) ou un raisonnement bas pour ce qui suit :

- Des requêtes factuelles simples
- Classifications directes
- Optimisation de la vitesse et des coûts
- Réponse simple aux questions

Important

Les modes de raisonnement avancés augmentent le temps et les coûts de formation, la latence et le coût des inférences, mais augmentent également la capacité du modèle à effectuer des tâches de raisonnement complexes.

Modèles pris en charge

RFT est onSageMaker HyperPod compatible avec Amazon Nova Lite 2.0 (amazon.nova-2-lite-v1:0:256 k).

Étapes majeures

Le processus RFT comprend quatre phases clés :

- Implémentation d'un évaluateur : créez une fonction de récompense pour évaluer de manière programmatique les réponses du modèle en fonction de vos critères de qualité.
- Instructions de téléchargement : préparez et téléchargez les données d'entraînement dans le format conversationnel spécifié avec des données de référence à des fins d'évaluation.
- Démarrage d'une tâche : lancez le processus de réglage précis du renforcement avec les paramètres que vous avez configurés.
- Surveillance : suivez les progrès de la formation à l'aide de tableaux de bord de mesures pour garantir que le modèle apprend efficacement.

Chaque étape s'appuie sur la précédente, l'évaluateur servant de base qui guide l'ensemble du processus de formation en fournissant des signaux de feedback cohérents.

Rubriques

- [RFT sur Nova 2.0](#)

RFT sur Nova 2.0

Les données d'entraînement RFT suivent le format conversationnel OpenAI. Chaque exemple de formation est un objet JSON contenant des messages, des réponses de référence et des définitions d'outils facultatives. Cette section fournit des conseils sur la préparation de données d'entraînement efficaces pour RFT sur Nova 2.0.

Rubriques

- [Format et structure des données](#)
- [Descriptions des champs](#)
- [Guidage des hyperparamètres](#)
- [Propriétés supplémentaires](#)
- [Recommandations relatives à la taille des jeux](#)
- [Caractéristiques des données d'entraînement efficaces](#)
- [Surveillance de la formation RFT](#)

Format et structure des données

Chaque exemple d'entraînement est un objet JSON contenant les éléments suivants :

- messages : un ensemble de tournants conversationnels utilisant les rôles système, utilisateur et éventuellement assistant
- reference_answer : Résultat attendu ou critères d'évaluation pour le calcul des récompenses
- outils (facultatif) : tableau de définitions de fonctions disponibles pour le modèle
- id (facultatif) : identifiant unique pour le suivi et la déduplication

Chaque exemple doit figurer sur une seule ligne de votre fichier JSONL, avec un objet JSON par ligne.

Exemple 1 : problème de chimie

L'exemple suivant montre un problème de chimie avec une réponse de référence contenant des valeurs de vérité fondamentale :

```
{
  "id": "chem-001",
  "messages": [
    {
      "role": "system",
      "content": "You are a helpful chemistry assistant"
    },
    {
      "role": "user",
```

```

    "content": "Predict hydrogen bond donors and acceptors for this SMILES:
CCN(CC)CCC(=O)c1sc(N)nc1C"
  }
],
"reference_answer": {
  "donor_bond_counts": 2,
  "acceptor_bond_counts": 4,
  "explanation": "Calculated using Lipinski's rule of five: N-H groups (2 donors), N
and 0 atoms with lone pairs (4 acceptors)"
}
}

```

Note

Le `reference_answer` contient des valeurs de vérité de base calculées à l'aide de règles spécifiques au domaine. Votre fonction de récompense compare les valeurs prédites du modèle à ces valeurs de référence pour calculer un score de récompense.

Exemple 2 : problème mathématique

L'exemple suivant montre un problème mathématique avec des étapes de résolution :

```

{
  "id": "math-001",
  "messages": [
    {
      "role": "system",
      "content": "You are a math tutor"
    },
    {
      "role": "user",
      "content": "Solve:  $2x + 5 = 13$ "
    }
  ],
  "reference_answer": {
    "solution": " $x = 4$ ",
    "steps": [" $2x = 13 - 5$ ", " $2x = 8$ ", " $x = 4$ "]
  }
}

```

Exemple 3 : utilisation de l'outil

L'exemple suivant montre l'utilisation de l'outil avec le comportement attendu :

```
{
  "id": "tool-001",
  "messages": [
    {
      "role": "system",
      "content": "You are a helpful game master assistant"
    },
    {
      "role": "user",
      "content": "Generate a strength stat for a warrior character. Apply a +2 racial
bonus modifier."
    }
  ],
  "tools": [
    {
      "type": "function",
      "function": {
        "name": "StatRollAPI",
        "description": "Generates character stats by rolling 4d6, dropping the lowest
die result, and applying a modifier.",
        "parameters": {
          "type": "object",
          "properties": {
            "modifier": {
              "description": "An integer representing the modifier to apply to the
total of the stat roll.",
              "type": "integer"
            }
          }
        },
        "required": ["modifier"]
      }
    }
  ],
  "reference_answer": {
    "tool_called": "StatRollAPI",
    "tool_parameters": {
      "modifier": 2
    },
  },
}
```

```

    "expected_behavior": "Call StatRollAPI with modifier=2 and return the calculated
    stat value"
  }
}

```

Descriptions des champs

Champ	Description	Informations complémentaires	Obligatoire
id	Identifiant unique pour cet exemple de RFT	Chaîne (par exemple, « sample-001 »). Utile pour le suivi et la déduplication.	Non
messages	Liste ordonnée des messages de chat qui définissent l'invite et le contexte	Tableau d'objets . Le mannequin les voit dans l'ordre. Commence généralement par un message système, puis par un message utilisateur.	Oui
messages [].role	Qui parle dans le message	Valeurs communes : « système », « utilisateur » (parfois « assistant » dans d'autres contextes)	Non
messages [].content	Le contenu textuel du message	Corde simple. Pour le système, il s'agit d'instructions, pour l'utilisateur, de la tâche ou de l'entrée.	Non
outils	Spécifications de l'outil disponibles pour	Matrice. Chaque élément définit l'interface et les métadonné	Non

Champ	Description	Informations complémentaires	Obligatoire
	le modèle dans cet exemple	es d'un outil. Les types peuvent inclure « fonction » ou « interne ».	
réponse_référence	La sortie du modèle attendue pour cet exemple	Chaîne ou objet en fonction de la tâche. Utilisé comme cible pour l'évaluation ou la formation.	Non

Note

Tous les champs personnalisés supplémentaires (par exemple, `task_id`, `difficulty_level`, `context_data`) ne sont pas validés et seront transmis à votre fonction de récompense sous forme de métadonnées.

Guidage des hyperparamètres

Utilisez les hyperparamètres recommandés suivants en fonction de votre approche d'entraînement :

Généralités :

- Époques : 1
- Taux d'apprentissage (lr) : 1e-7
- Nombre de générations : 8
- Nombre maximum de nouveaux jetons : 8192
- Taille du lot : 256

LoRa (adaptation de bas rang) :

- Rang LoRa : 32

 Note

Ajustez ces valeurs en fonction de la taille de votre jeu de données et des performances de validation. Surveillez les indicateurs d'entraînement pour éviter le surajustement.

Propriétés supplémentaires

Le paramètre « `AdditionalProperties` » : `true` vous permet d'inclure des champs personnalisés au-delà des exigences de base du schéma, ce qui vous permet d'ajouter toutes les données dont votre fonction de récompense a besoin pour une évaluation correcte.

Champs supplémentaires courants

Vous pouvez inclure les types de champs supplémentaires suivants :

Métadonnées :

- `task_id` : identifiant unique pour le suivi
- `difficulty_level` : indicateur de complexité du problème
- `domaine` : Domaine ou catégorie
- `expected_reasoning_steps` : nombre d'étapes de la solution

Critères d'évaluation :

- `evaluation_criteria` : Rubriques de notation spécifiques
- `custom_scoring_weights` : importance relative des différents aspects
- `context_data` : informations générales sur le problème
- `external_references` : liens vers la documentation ou les ressources pertinentes

Exemple avec des propriétés supplémentaires

L'exemple suivant inclut des champs de métadonnées personnalisés :

```
{  
  "id": "algebra_001",  
  "messages": [  

```

```
{
  "role": "system",
  "content": "You are a math tutor"
},
{
  "role": "user",
  "content": "Solve:  $2x + 5 = 13$ "
}
],
"reference_answer": {
  "solution": " $x = 4$ ",
  "steps": [" $2x = 13 - 5$ ", " $2x = 8$ ", " $x = 4$ "]
},
"task_id": "algebra_001",
"difficulty_level": "easy",
"domain": "algebra",
"expected_reasoning_steps": 3
}
```

Recommandations relatives à la taille des jeux

Point de départ

Commencez par les tailles de jeu de données minimales suivantes :

- Minimum de 100 exemples de formation
- Minimum de 100 exemples d'évaluation

Priorisez les données d'entrée de haute qualité et une fonction de récompense fiable qui s'exécute de manière cohérente sur les réponses du modèle.

Approche axée sur l'évaluation

Avant d'investir dans une formation RFT à grande échelle, évaluez les performances de base de votre modèle :

- Haute performance (récompense supérieure à 95 %) : le RFT n'est peut-être pas nécessaire : votre modèle fonctionne déjà bien
- Très mauvaise performance (0 % de récompense) : passez d'abord à SFT pour établir les fonctionnalités de base
- Performances modérées : le RFT est probablement approprié

Cette approche axée sur l'évaluation garantit que votre fonction de récompense est exempte de bogues et détermine si la RFT est la bonne méthode pour votre cas d'utilisation. Commencer modestement vous permet de vous familiariser avec le flux de travail RFT, d'identifier et de résoudre les problèmes rapidement, de valider votre approche avant de passer à l'échelle supérieure et de tester la fiabilité de la fonction de récompense. Une fois validé, vous pouvez l'étendre à de plus grands ensembles de données pour améliorer encore les performances.

Caractéristiques des données d'entraînement efficaces

Clarté et cohérence

Les bons exemples de RFT nécessitent des données d'entrée claires et sans ambiguïté qui permettent un calcul précis des récompenses sur les différents modèles de sortie. Évitez le bruit dans vos données, notamment :

- Formatage incohérent
- Étiquettes ou instructions contradictoires
- Des instructions ambiguës
- Réponses de référence contradictoires

Toute ambiguïté induira le processus de formation en erreur et incitera le modèle à apprendre des comportements involontaires.

Diversité

Votre ensemble de données doit capturer toute la diversité des cas d'utilisation en production afin de garantir de solides performances réelles. Inclure :

- Différents types de problèmes et niveaux de difficulté
- Différents formats d'entrée et boîtiers
- Échantillons représentatifs de tous les scénarios attendus

Cette diversité permet d'éviter le surajustement et garantit que le modèle gère les entrées inconnues avec élégance.

Considérations relatives aux fonctions de récompense

Concevez votre fonction de récompense pour un entraînement efficace :

- Exécuter en quelques secondes (et non en quelques minutes)
- Parallélisez efficacement avec Lambda
- Obtenez des scores cohérents et fiables
- Gérez les différents types de sorties de modèles avec élégance

Des fonctions de récompense rapides et évolutives permettent une itération rapide et une expérimentation rentable à grande échelle.

Surveillance de la formation RFT

Surveillez les indicateurs clés pendant la formation pour garantir un apprentissage efficace et identifier rapidement les problèmes potentiels.

Rubriques

- [Indicateurs clés à suivre](#)
- [Évaluation après RFT](#)
- [Utilisation de modèles affinés](#)
- [Limites et meilleures pratiques](#)
- [Résolution des problèmes](#)

Indicateurs clés à suivre

Surveillez les indicateurs suivants à l'aide de ces indicateurs MIFlow pendant l'entraînement :

Indicateurs de récompenses :

- Note de récompense moyenne : qualité globale des réponses du modèle (devrait augmenter au fil du temps)
- Distribution des récompenses : pourcentage de réponses recevant des récompenses élevées, moyennes et faibles
- Récompenses liées à la formation ou à la validation : comparez pour détecter le surajustement

Indicateurs de formation :

- Mises à jour des politiques : nombre de mises à jour du poids réussies
- Taux d'achèvement du déploiement : pourcentage d'échantillons évalués avec succès

En ce qui concerne les motifs :

- Plafond des récompenses (indique un mauvais apprentissage)
- Les récompenses de validation diminuent tandis que les récompenses d'entraînement augmentent (surajustement)
- La variance des récompenses augmente de manière significative au fil du temps (instabilité)
- Pourcentage élevé d'erreurs liées à la fonction de récompense (problèmes de mise en œuvre)

Quand arrêter l'entraînement :

- Les indicateurs de performance cibles sont atteints
- Les récompenses plafonnent et ne s'améliorent plus
- Les performances de validation se dégradent (surajustement détecté)
- Le budget de formation maximal est atteint

Évaluation après RFT

Une fois la formation terminée, évaluez votre modèle affiné pour évaluer les améliorations de performance :

- Exécuter une tâche d'évaluation RFT : utilisez le point de contrôle de votre formation RFT comme modèle
- Comparaison à la base de référence : évaluez à la fois le modèle de base et le modèle affiné sur le même ensemble de test
- Analyser les indicateurs : passez en revue les indicateurs spécifiques aux tâches (précision, scores de récompense, etc.)
- Réaliser un examen qualitatif : inspecter manuellement la qualité des échantillons de sortie

Pour les procédures d'évaluation détaillées, consultez la section Évaluation.

Utilisation de modèles affinés

Accès aux points de contrôle :

Une fois la formation terminée, localisez votre point de contrôle :

1. Naviguez vers votre `output_path` dans S3

2. Téléchargez et extrayez `output.tar.gz`
3. Ouvrez `manifest.json`.
4. Copiez la `checkpoint_s3_bucket` valeur

Déploiement à des fins d'inférence :

Utilisez le chemin du point de contrôle S3 à des fins d'inférence ou de formation continue :

```
run:
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: "s3://customer-escrow-<account-number>-smtj-<unique-
  identifier>/<job-name>"
```

Pour les instructions de déploiement et d'inférence, reportez-vous à la section Inférence.

Limites et meilleures pratiques

Limitations actuelles :

Restrictions relatives aux versions bêta

- Besoin de créer un nouveau groupe RIG pour RFT. Cette limitation sera résolue par GA.
- Exigences relatives au type d'instance : seules les instances P5 sont prises en charge (minimum 8 instances P5.48xLarge). Prochainement : Support pour les types d'instances plus petits (ETA : mi-janvier 2025).

Limites fonctionnelles :

- Délai Lambda de 15 minutes : les fonctions de récompense doivent être terminées dans les 15 minutes
- Un tour uniquement : les conversations à plusieurs tours ne sont pas prises en charge
- Ensembles de données de validation : non pris en charge pendant l'entraînement. Utilisez des tâches d'évaluation distinctes pour évaluer les progrès de la formation.

Considérations relatives à la formation :

- Scénarios à faibles récompenses : cela peut être difficile lorsque moins de 5 % des exemples reçoivent des récompenses positives. Pensez d'abord à SFT

- Exigences en matière de données : nécessite une diversité suffisante pour apprendre efficacement
- Coût de calcul : plus coûteux que le réglage fin supervisé

Nova Forge supprime certaines de ces limitations :

- Prend en charge les conversations à plusieurs tours
- Permet aux fonctions de récompense de plus de 15 minutes
- Fournit des algorithmes avancés et des options de réglage
- Conçu pour les cas d'utilisation complexes en entreprise, spécialement conçu pour créer des modèles avant-gardistes

Bonnes pratiques :

Commencez petit et agrandissez :

- Commencez avec un minimum d'ensembles de données (100 à 200 exemples) et peu d'époques d'entraînement
- Validez votre approche avant de la mettre à l'échelle
- Augmentez progressivement la taille de l'ensemble de données et les étapes d'entraînement en fonction des résultats

Base de référence avec SFT d'abord :

- Si les scores de récompense sont constamment faibles (par exemple, toujours 0), effectuez SFT avant RFT
- La RFT nécessite des performances de base raisonnables pour s'améliorer efficacement

Concevez des fonctions de récompense efficaces :

- Exécuter en quelques secondes, et non en quelques minutes
- Minimiser les appels d'API externes
- Utiliser des algorithmes et des structures de données efficaces
- Mettre en œuvre une gestion appropriée des erreurs
- Faites des tests approfondis avant l'entraînement
- Tirez parti des fonctionnalités de mise à l'échelle parallèle de Lambda

Surveillez activement la formation :

- Suivez les scores de récompense moyens au fil du temps
- Regardez la distribution des récompenses entre les échantillons
- Comparez les récompenses de formation aux récompenses de validation
- Recherchez les tendances inquiétantes (plateaux, surajustement, instabilité)

Itérer en fonction des résultats :

- Si les récompenses ne s'améliorent pas après plusieurs itérations, ajustez le design de la fonction de récompense
- Augmenter la diversité des ensembles de données pour fournir des signaux d'apprentissage plus clairs
- Envisagez de passer à SFT si les récompenses restent proches de zéro
- Expérimentez avec différents hyperparamètres (taux d'apprentissage, taille du lot)

Optimisez la qualité des données :

- Garantir la diversité et la représentativité des exemples
- Incluez des étuis de pointe et des échantillons difficiles
- Vérifier que la fonction de récompense note correctement tous les types d'exemples
- Supprimez ou corrigez les échantillons qui perturbent la fonction de récompense

Résolution des problèmes

Erreurs liées à la fonction de récompense :

Symptômes : taux d'erreur élevé lors des appels à la fonction de récompense pendant l'entraînement

Problème	Symptômes	Résolution
délai d'expiration Lambda	Délais d'attente fréquents après 15 minutes	Optimisez les performances des fonctions ; considérez Nova Forge pour les évaluations complexes

Problème	Symptômes	Résolution
Simultanéité insuffisante	Erreurs de régulation Lambda	Augmenter <code>lambda_concurrency_limit</code> ou demander une augmentation du quota
Format de retour non valide	L'entraînement échoue en raison d'erreurs de format	Vérifiez que la structure de retour correspond au format d'interface requis
Exceptions non gérées	Erreurs intermittentes	Ajoutez une gestion et une journalisation complètes des erreurs
Défaillances d'API externes	Notation incohérente	Mettre en œuvre une logique de nouvelle tentative et des stratégies de repli

Mauvaises performances d'entraînement :

Symptômes : les récompenses ne s'améliorent pas ou plafonnent à de faibles valeurs

Résolutions :

- Vérifiez l'exactitude de la fonction de récompense : testez avec des exemples connus good/bad
- Vérifiez les performances de référence : évaluez le modèle de base ; si la précision est proche de zéro, effectuez d'abord le SFT
- Augmenter la diversité des données : ajouter des exemples plus variés couvrant différents scénarios
- Ajustez les hyperparamètres : essayez différents taux d'apprentissage ou différentes tailles de lots
- Vérifiez la qualité du signal de récompense : assurez-vous que les récompenses font la différence entre les bonnes et les mauvaises réponses

Surajustement :

Symptômes : les récompenses d'entraînement augmentent tandis que les récompenses de validation diminuent

Résolutions :

- Réduisez le nombre d'étapes d'entraînement : arrêtez l'entraînement plus tôt
- Augmenter la taille du jeu de données : ajouter d'autres exemples de formation
- Ajouter une régularisation : Ajuster ou `weight_decay` `entropy_coeff`
- Améliorez la diversité des données : assurez-vous que le kit de formation représente une distribution complète

Évaluation de votre modèle entraîné

Une formule d'évaluation est un fichier de configuration YAML qui définit la manière dont votre tâche d'évaluation des modèles Amazon Nova est exécutée. Avec cette formule, vous pouvez évaluer les performances d'un modèle de base ou entraîné par rapport à des points de référence courants ou à vos propres jeux de données personnalisés. Les métriques peuvent être stockées dans Amazon S3 ou TensorBoard. L'évaluation fournit des métriques quantitatives qui vous aident à évaluer les performances du modèle dans le cadre de différentes tâches afin de déterminer si une personnalisation supplémentaire est nécessaire.

L'évaluation des modèles est un processus hors ligne, dans le cadre duquel les modèles sont testés par rapport à des points de référence fixes avec des réponses prédéfinies. Ils ne sont pas évalués en temps réel ni par rapport aux interactions réelles des utilisateurs. Pour les évaluations en temps réel, vous pouvez évaluer le modèle après son déploiement sur Amazon Bedrock en appelant le runtime Amazon Bedrock. APIs

Rubriques

- [Tâches de point de référence disponibles](#)
- [Compréhension des paramètres de la formule](#)
- [Exemples de formules d'évaluation](#)
- [Démarrage d'une tâche d'évaluation](#)
- [Accès aux résultats d'évaluation et analyse](#)
- [Évaluation RFT](#)

Tâches de point de référence disponibles

Un exemple de package de code est disponible qui montre comment calculer les mesures de référence à l'aide de la fonctionnalité d'évaluation du modèle d' SageMaker intelligence artificielle

pour Amazon Nova. Pour accéder aux packages de code, consultez [Sample-nova-lighteval-custom-task](#).

Voici une liste des points de référence standard pris en charge. Vous pouvez spécifier les points de référence suivants dans le paramètre `eval_task` :

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
mmlu	Texte	Compréhension linguistique multitâche : teste les connaissances dans 57 domaines.	précision	zs_cot	Oui
mmlu_pro	Texte	MMLU – Sous-ensemble professionnel : se concentre sur les domaines professionnels tels que le droit, la médecine, la comptabilité et l'ingénierie.	précision	zs_cot	Non
bbh	Texte	Tâches de raisonnement avancées : ensemble de problèmes difficiles qui mettent à	précision	zs_cot	Oui

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
		l'épreuve des compétences cognitives et de résolution de problèmes de haut niveau.			
gpqa	Texte	Réponses aux questions de physique générale : évalue la compréhension des concepts de physique et les capacités de résolution de problèmes connexes.	précision	zs_cot	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
math	Texte	Résolution de problèmes mathématiques : mesure le raisonnement mathématique dans des domaines tels que l'algèbre, le calcul et les problèmes de mots.	exact_match	zs_cot	Oui
strong_reject	Texte	Tâche de contrôle qualité : teste la capacité du modèle à détecter et à rejeter le contenu inapproprié, préjudiciable ou incorrect.	deflection	zs	Oui

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
IFEval	Texte	Évaluation suivant les instructions : évalue la précision avec laquelle un modèle suit les instructions données et exécute les tâches conformément aux spécifications.	précision	zs	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
gen_qa	Texte	Évaluation personnelle du jeu de données : vous permet de fournir votre propre jeu de données à des fins d'analyse comparative, et de comparer les sorties du modèle aux réponses de référence avec des métriques telles que ROUGE et BLEU.	tout	gen_qa	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
llm_judge	Texte	LLM-as-a-Judge Comparaison des préférences : utilise un modèle Amazon Nova Judge pour déterminer la préférence entre les réponses jumelées (B par rapport à A) pour vos invites, en calculant la probabilité que B soit préféré à A.	tout	judge	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
humaneval	Texte	HumanEval - Un ensemble de données de référence conçu pour évaluer les capacités de génération de code des grands modèles de langage	pass@1	zs	Non
mm_llm_judge	Multimodal (image)	Ce nouveau benchmark se comporte de la même manière que le modèle basé sur le texte ci-dessus llm_judge . La seule différence est qu'il prend en charge l'inférence d'image.	tout	judge	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
rubrique_ilm_judge	Texte	Rubric Judge est un modèle LLM-as-a-judge d'évaluation amélioré basé sur Amazon Nova 2.0 Lite. Contrairement au modèle de juge original qui ne fournit que des verdicts de préférence, Rubric Judge génère de manière dynamique des critères d'évaluation personnalisés adaptés à chaque invite et attribue des scores granulaires sur plusieurs dimensions.	tout	judge	Non

Comparaison	Modalité	Description	Métriques	Stratégie	Sous-tâche disponible
aime_2024	Texte	AIME 2024 - Problèmes de l'examen de mathématiques sur invitation américain testant le raisonnement mathématique avancé et la résolution de problèmes	exact_match	zs_cot	Non
calendar_planning	Texte	Natural Plan - Planification du calendrier, test des tâches, capacités de planification pour planifier des réunions sur plusieurs jours et plusieurs personnes	exact_match	fs	Non

Les sous-tâches mmlu suivantes sont disponibles :

```
MMLU_SUBTASKS = [
    "abstract_algebra",
    "anatomy",
```

```
"astronomy",
"business_ethics",
"clinical_knowledge",
"college_biology",
"college_chemistry",
"college_computer_science",
"college_mathematics",
"college_medicine",
"college_physics",
"computer_security",
"conceptual_physics",
"econometrics",
"electrical_engineering",
"elementary_mathematics",
"formal_logic",
"global_facts",
"high_school_biology",
"high_school_chemistry",
"high_school_computer_science",
"high_school_european_history",
"high_school_geography",
"high_school_government_and_politics",
"high_school_macroeconomics",
"high_school_mathematics",
"high_school_microeconomics",
"high_school_physics",
"high_school_psychology",
"high_school_statistics",
"high_school_us_history",
"high_school_world_history",
"human_aging",
"human_sexuality",
"international_law",
"jurisprudence",
"logical_fallacies",
"machine_learning",
"management",
"marketing",
"medical_genetics",
"miscellaneous",
"moral_disputes",
"moral_scenarios",
"nutrition",
"philosophy",
```

```
"prehistory",  
"professional_accounting",  
"professional_law",  
"professional_medicine",  
"professional_psychology",  
"public_relations",  
"security_studies",  
"sociology",  
"us_foreign_policy",  
"virology",  
"world_religions"  
]
```

Les sous-tâches bbh suivantes sont disponibles :

```
BBH_SUBTASKS = [  
    "boolean_expressions",  
    "causal_judgement",  
    "date_understanding",  
    "disambiguation_qa",  
    "dyck_languages",  
    "formal_fallacies",  
    "geometric_shapes",  
    "hyperbaton",  
    "logical_deduction_five_objects",  
    "logical_deduction_seven_objects",  
    "logical_deduction_three_objects",  
    "movie_recommendation",  
    "multistep_arithmetic_two",  
    "navigate",  
    "object_counting",  
    "penguins_in_a_table",  
    "reasoning_about_colored_objects",  
    "ruin_names",  
    "salient_translation_error_detection",  
    "snarks",  
    "sports_understanding",  
    "temporal_sequences",  
    "tracking_shuffled_objects_five_objects",  
    "tracking_shuffled_objects_seven_objects",  
    "tracking_shuffled_objects_three_objects",  
    "web_of_lies",  
    "word_sorting"
```

```
]
```

Les sous-tâches math suivantes sont disponibles :

```
MATH_SUBTASKS = [  
    "algebra",  
    "counting_and_probability",  
    "geometry",  
    "intermediate_algebra",  
    "number_theory",  
    "prealgebra",  
    "precalculus",  
]
```

Compréhension des paramètres de la formule

Configuration d'exécution

Vous trouverez ci-dessous une configuration d'exécution générale et une explication des paramètres impliqués.

```
run:  
  name: eval_job_name  
  model_type: amazon.nova-micro-v1:0:128k  
  model_name_or_path: nova-micro/prod  
  replicas: 1  
  data_s3_path: ""  
  output_s3_path: s3://output_path  
  mlflow_tracking_uri: ""  
  mlflow_experiment_name : ""  
  mlflow_run_name : ""
```

- **name** : nom descriptif de votre tâche d'évaluation (obligatoire). Cela permet d'identifier votre tâche dans la AWS console.
- **model_type** : spécifie la variante du modèle Amazon Nova à utiliser (obligatoire). Ne modifiez pas ce champ manuellement. Voici les options :
 - `amazon.nova-micro-v1:0:128k`
 - `amazon.nova-lite-v1:0:300k`
 - `amazon.nova-pro-v1:0:300k`
 - `amazon.nova-2-lite-v1:0:256k`

- `model_name_or_path` : chemin vers le modèle de base ou le chemin S3 pour le point de contrôle post-entraîné (obligatoire). Voici les options :
 - `nova-micro/prod`
 - `nova-lite/prod`
 - `nova-pro/prod`
 - `nova-lite-2/prod`
 - (chemin S3 pour le chemin de point de contrôle post-entraîné) `s3://<escrow bucket>/<job id>/outputs/checkpoints`
- `replicas` : nombre d'instances de calcul à utiliser pour l'entraînement distribué (obligatoire). Vous devez définir cette valeur sur 1, car les nœuds multiples ne sont pas pris en charge.
- `data_s3_path` : chemin S3 vers le jeu de données d'entrée (obligatoire). Laissez ce paramètre vide, sauf si vous utilisez la formule « apportez votre propre jeu de données » ou LLM-juge.
- `output_s3_path` : chemin S3 pour stocker les artefacts d'évaluation de sortie (obligatoire). Notez que le compartiment S3 de sortie doit être créé par le même compte que celui qui crée la tâche.
- `mlflow_tracking_uri` : ARN du serveur de MLflow suivi (facultatif) pour le suivi des MLFlow exécutions/expériences. Veuillez vous assurer que vous êtes autorisé à accéder au serveur de suivi à partir du rôle d'exécution SageMaker AI

Configuration d'évaluation

Vous trouverez ci-dessous une configuration d'évaluation des modèles et une explication des paramètres impliqués.

```
evaluation:
  task: mmlu
  strategy: zs_cot
  subtask: mathematics
  metric: accuracy
```

- `task` : spécifie le point de référence ou la tâche d'évaluation à utiliser (obligatoire).

Liste de tâches prises en charge :

- `mmlu`
- `mmlu_pro`
- `bbh`

- `gpqa`
- `math`
- `strong_reject`
- `gen_qa`
- `ifeval`
- `llm_judge`
- `humaneval`
- `mm_llm_judge`
- `rubrique_llm_judge`
- `aime_2024`
- `calendar_planning`
- `humaneval`
- `strategy` : définit l'approche d'évaluation (obligatoire) :
 - `zs_cot` : Zero-shot Chain-of-Thought - Une approche visant à générer de grands modèles de langage qui encouragent le step-by-step raisonnement sans nécessiter d'exemples explicites.
 - `zs` : zéro coup, approche pour résoudre un problème sans aucun exemple d'entraînement préalable.
 - `gen_qa` : stratégie spécifique pour apporter vos propres formules de jeux de données.
 - `judge` : stratégie spécifique à Amazon Nova LLM as Judge et `mm_llm_judge`.
- `subtask` : spécifie une sous-tâche spécifique pour certaines tâches d'évaluation (facultatif et amovible). Supprimez ceci de votre formule si votre tâche ne comporte aucune sous-tâche.
- `metric` : métrique d'évaluation à utiliser (obligatoire).
 - `accuracy` : pourcentage de bonnes réponses
 - `exact_match` : renvoie le taux auquel les chaînes prédites en entrée correspondent exactement à leurs références (pour un point de référence `math`).
 - `deflection` : renvoie la déflexion relative par rapport au modèle de base et la différence entre les métriques de signification (pour un point de référence `strong_reject`).
 - `pass@1` : métrique utilisée pour mesurer le pourcentage de cas où la prédiction la plus fiable du modèle correspond à la bonne réponse (pour un point de référence `humaneval`).
 - `all` : renvoie les métriques suivantes :

- Pour `gen_qa` et « apportez votre propre point de référence de jeu de données », renvoie les métriques suivantes :
 - `rouge1` : mesure le chevauchement des unigrammes (mots uniques) entre le texte généré et le texte de référence.
 - `rouge2` : mesure le chevauchement des bigrammes (deux mots consécutifs) entre le texte généré et le texte de référence.
 - `rougeL` : mesure la plus longue sous-séquence commune entre les textes, en tenant compte des lacunes dans la correspondance.
 - `exact_match` : score binaire (0 ou 1) indiquant si le texte généré correspond exactement au texte de référence, caractère par caractère.
 - `quasi_exact_match` : similaire à la correspondance exacte, mais plus indulgent, ignorant généralement les différences entre majuscules et minuscules, la ponctuation et les espaces.
 - `f1_score` : moyenne harmonique de précision et de rappel, mesurant le chevauchement des mots entre les réponses prédites et les réponses de référence.
 - `f1_score_quasi` : similaire à `f1_score` mais avec une correspondance plus souple, en utilisant une comparaison de texte normalisée qui ignore les différences mineures.
 - `bleu` : mesure la précision des correspondances n-grammes entre le texte généré et le texte de référence, couramment utilisé dans l'évaluation des traductions.
- Pour `llm_judge` et `etmm_llm_judge`, apportez votre propre référence de jeu de données, renvoyez les métriques suivantes :
 - `a_scores` : nombre de victoires pour `response_A` sur les passes d'évaluation avant et arrière.
 - `a_scores_stderr` : erreur standard de `response_A` scores pour les jugements par paires.
 - `b_scores` : nombre de victoires pour `response_B` sur les passes d'évaluation avant et arrière.
 - `b_scores_stderr` : erreur standard de `response_B` scores pour les jugements par paires.
 - `ties` : nombre de jugements où `response_A` et `response_B` sont évaluées comme égales.
 - `ties_stderr` : erreur standard des égalités pour les jugements par paires.
 - `inference_error` : nombre de jugements qui n'ont pas pu être correctement évalués.

- `score` : score agrégé basé sur les victoires obtenues lors des passes avant et arrière pour `response_B`.
- `score_stderr` : erreur type du score agrégé pour les jugements par paires.
- `winrate` : probabilité que la réponse `response_B` soit préférée à `response_A`, calculée à l'aide de la probabilité de Bradley-Terry.
- `lower_rate` : limite inférieure (2,5 percentile) du taux de victoire estimé à partir d'un échantillonnage d'amorçage.

Configuration d'inférence

Vous trouverez ci-dessous une configuration d'inférence et une explication des paramètres impliqués. Tous les paramètres sont facultatifs.

```
inference:
  max_new_tokens: 200
  top_k: -1
  top_p: 1.0
  temperature: 0
  top_logprobs: 10
  reasoning_effort: null # options: low/high to enable reasoning or null to disable reasoning
```

- `max_new_tokens` : nombre maximum de jetons à générer. Il doit s'agir d'un nombre entier.
- `top_k` : nombre de jetons présentant la probabilité la plus élevée à prendre en compte. Il doit s'agir d'un nombre entier.
- `top_p` : seuil de probabilité cumulé pour l'échantillonnage de jetons. Il doit s'agir d'une valeur flottante comprise entre 0 et 1 inclus.
- `temperature` : imprévisibilité dans la sélection des jetons. Plus les valeurs sont élevées, plus le caractère aléatoire est élevé. Utilisez 0 pour rendre les résultats déterministes. Il doit s'agir d'une valeur flottante d'au minimum 0.
- `top_logprobs`: Le nombre de meilleures probs logprobs à renvoyer dans la réponse d'inférence. Cette valeur doit être un entier compris entre 0 et 20. Les logprobs contiennent les jetons de sortie pris en compte et les probabilités de journalisation de chaque jeton de sortie renvoyé dans le contenu du message.
- `reasoning_effort`: contrôle le comportement de raisonnement pour les modèles capables de raisonner. Défini `reasoning_effort` uniquement lorsque `model_type` spécifie un modèle

capable de raisonner (actuellement `amazon.nova-2-lite-v1:0:256k`). Les options disponibles sont `null` (valeur par défaut si elle n'est pas définie ; désactive le raisonnement) `low`, ou `high`.

Notez que pour `humaneval`, nous recommandons la configuration d'inférence suivante :

```
inference:
  top_k: 1
  max_new_tokens: 1600
  temperature: 0.0
```

MLFlow configuration

Vous trouverez ci-dessous une MLFlow configuration et une explication des paramètres concernés. Tous les paramètres sont facultatifs.

```
run:
  mlflow_tracking_uri: ""
  mlflow_experiment_name: ""
  mlflow_run_name: ""
```

- `mlflow_tracking_uri`: Facultatif) L'emplacement du serveur de MLflow suivi (uniquement nécessaire sur SMHP)
- `mlflow_experiment_name`: (Facultatif) Nom de l'expérience pour regrouper les essais de machine learning associés
- `mlflow_run_name`: (Facultatif) Nom personnalisé pour un entraînement spécifique effectué dans le cadre d'une expérience

Exemples de formules d'évaluation

Amazon Nova propose quatre types de recettes d'évaluation, disponibles dans le GitHub référentiel de SageMaker HyperPod recettes.

Formules de point de référence de texte général

Ces formules vous permettent d'évaluer les capacités fondamentales des modèles Amazon Nova grâce à une suite complète de points de référence basés uniquement sur le texte. Elles sont fournies au format `xxx_general_text_benchmark_eval.yaml`.

Apportez vos propres formules de point de référence de jeu de données

Ces formules vous permettent d'utiliser votre propre jeu de données à des fins d'analyse comparative et de comparer les résultats du modèle aux réponses de référence à l'aide de différents types de métriques. Elles sont fournies au format `xxx_bring_your_own_dataset_eval.yaml`.

Voici les exigences relatives à l'apport de votre propre jeu de données :

- Exigences en matière de format de fichier d'entrée
 - Vous devez inclure un seul fichier `gen_qa.jsonl` contenant des exemples d'évaluation.
 - Votre ensemble de données doit être chargé sur un emplacement S3 où SageMaker Training Job peut y accéder.
 - Le fichier doit suivre le format de schéma requis pour un jeu de données de questions/réponses général.
- Exigences de format de schéma : chaque ligne du fichier JSONL doit être un objet JSON avec les champs suivants :
 - `query` : chaîne contenant la question ou l'instruction nécessitant une réponse (obligatoire)
 - `response` : chaîne contenant le résultat attendu du modèle (obligatoire)
 - `system` : chaîne contenant l'invite système qui définit le comportement, le rôle ou la personnalité du modèle d'IA avant qu'il ne traite la requête (facultatif)
 - `metadata`: (Facultatif) Chaîne contenant les métadonnées associées à l'entrée à des fins de balisage.

Voici un exemple d'entrée « apportez votre propre jeu de données » :

```
{
  "system":"You are a english major with top marks in class who likes to give minimal
word responses: ",
  "query":"What is the symbol that ends the sentence as a question",
  "response":"?"
}
{
  "system":"You are a pattern analysis specialist that provides succinct answers: ",
  "query":"What is the next number in this series? 1, 2, 4, 8, 16, ?",
  "response":"32"
}
{
```

```
"system":"You have great attention to detail that follows instructions accurately:",
"query":"Repeat only the last two words of the following: I ate a hamburger today and it was kind of dry",
"response":"of dry"
}
```

Pour utiliser votre jeu de données personnalisé, modifiez votre formule d'évaluation avec les champs obligatoires suivants, sans modifier le contenu :

```
evaluation:
  task: gen_qa
  strategy: gen_qa
  metric: all
```

Les limites suivantes s'appliquent :

- Un seul fichier JSONL est autorisé par évaluation.
- Le fichier doit suivre strictement le schéma défini.
- Limite de longueur du contexte : pour chaque échantillon de l'ensemble de données, la longueur du contexte (y compris le système et les invites de requête) doit être inférieure à 3,5 ko.

Formules de point de référence Nova LLM-juge

Amazon Nova LLM-juge est une caractéristique d'évaluation des modèles qui vous permet de comparer la qualité des réponses d'un modèle aux réponses d'un modèle de référence à l'aide d'un jeu de données personnalisé. Il prend en compte un ensemble de données contenant des instructions, des réponses de base et des réponses aux challengers, et utilise un modèle Amazon Nova Judge pour fournir une métrique du taux de victoire basée sur la probabilité de [Bradley-Terry](#) avec des comparaisons par paires.

Les formules sont fournies au format `xxx_llm_judge_eval.yaml`.

Les exigences liées à LLM-juge sont les suivantes :

- Exigences en matière de format de fichier d'entrée
 - Incluez un seul fichier `llm_judge.jsonl` contenant des exemples d'évaluation. Le fichier doit être nommé `llm_judge.jsonl`.

- Votre ensemble de données doit être téléchargé vers un emplacement S3 auquel [SageMaker AI SageMaker HyperPod RIG](#) peut accéder.
- Le fichier doit respecter le format de schéma requis pour le jeu de données `llm_judge.jsonl`.
- Le jeu de données d'entrée doit garantir que tous les enregistrements ont une longueur de contexte inférieure à 12 000.
- Exigences de format de schéma : chaque ligne du fichier JSONL doit être un objet JSON avec les champs suivants :
 - `prompt` : chaîne contenant l'invite de la réponse générée (obligatoire).
 - `response_A` : chaîne contenant la réponse de base.
 - `response_B` : chaîne contenant la réponse alternative comparée à la réponse de base.

Voici un exemple d'entrée LLM-juge :

```
{
  "prompt": "What is the most effective way to combat climate change?",
  "response_A": "The most effective way to combat climate change is through a combination of transitioning to renewable energy sources and implementing strict carbon pricing policies. This creates economic incentives for businesses to reduce emissions while promoting clean energy adoption.",
  "response_B": "We should focus on renewable energy. Solar and wind power are good. People should drive electric cars. Companies need to pollute less."
}
{
  "prompt": "Explain how a computer's CPU works",
  "response_A": "CPU is like brain of computer. It does math and makes computer work fast. Has lots of tiny parts inside.",
  "response_B": "A CPU (Central Processing Unit) functions through a fetch-execute cycle, where instructions are retrieved from memory, decoded, and executed through its arithmetic logic unit (ALU). It coordinates with cache memory and registers to process data efficiently using binary operations."
}
{
  "prompt": "How does photosynthesis work?",
  "response_A": "Plants do photosynthesis to make food. They use sunlight and water. It happens in leaves.",
  "response_B": "Photosynthesis is a complex biochemical process where plants convert light energy into chemical energy. They utilize chlorophyll to absorb sunlight, combining CO2 and water to produce glucose and oxygen through a series of chemical reactions in chloroplasts."
```

```
}
```

Pour utiliser votre jeu de données personnalisé, modifiez votre formule d'évaluation avec les champs obligatoires suivants, sans modifier le contenu :

```
evaluation:
  task: llm_judge
  strategy: judge
  metric: all
```

Les limites suivantes s'appliquent :

- Un seul fichier JSONL est autorisé par évaluation.
- Le fichier doit suivre strictement le schéma défini.
- Les modèles Amazon Nova Judge sont identiques pour toutes les caractéristiques de la famille de modèles (à savoir, Lite, Micro et Pro).
- Les modèles de juges personnalisés ne sont actuellement pas pris en charge.
- Limite de longueur du contexte : pour chaque échantillon de l'ensemble de données, la longueur du contexte (y compris le système et les invites de requête) doit être inférieure à 7 ko.

Nova LLM en tant que juge pour les recettes de référence multimodales (image)

Nova LLM Judge pour multimodal (image), abréviation d'Amazon Nova MM_LLM Judge, est une fonctionnalité d'évaluation de modèle qui vous permet de comparer la qualité des réponses d'un modèle à celles d'un modèle de référence à l'aide d'un ensemble de données personnalisé. Il accepte un ensemble de données contenant des invites, des réponses de base et des réponses aux challengers, ainsi que des images sous forme de chaîne codée en Base64, puis utilise un modèle Amazon Nova Judge pour fournir une métrique du taux de victoire basée sur la probabilité de [Bradley-Terry](#) par le biais de comparaisons par paires. Format de formule : `xxx_mm_llm_judge_eval.yaml`.

Exigences relatives à le jeu de données Nova LLM

Format de fichier :

- Fichier `mm_llm_judge.jsonl` unique contenant des exemples d'évaluation. Le nom du fichier doit être exact `mm_llm_judge.jsonl`.

- Vous devez télécharger votre ensemble de données sur un emplacement S3 où SageMaker Training Jobs peut y accéder.
- Le fichier doit respecter le format de schéma requis pour le jeu de données `mm_11m_judge`.
- L'ensemble de données en entrée doit garantir que tous les enregistrements ont une longueur de contexte inférieure à 12 k, à l'exception de l'attribut de l'image.

Format de schéma : chaque ligne du fichier `.jsonl` doit être un objet JSON avec les champs suivants.

- Champs obligatoires.

`prompt` : chaîne contenant l'invite de réponse générée.

`images` : Tableau contenant une liste d'objets avec des attributs de données (les valeurs sont des chaînes d'image codées en Base64).

`response_A` : chaîne contenant la réponse de base.

`response_B` : chaîne contenant la réponse alternative comparée à la réponse de base.

Exemple d'entrée

Pour des raisons de lisibilité, l'exemple suivant inclut de nouvelles lignes et des indentations, mais dans le jeu de données actuel, chaque enregistrement doit figurer sur une seule ligne.

```
{
  "prompt": "what is in the image?",
  "images": [
    {
      "data": "data:image/jpeg;base64,/9j/2wBDAAQDAwQDAwQEAWQFBAQFBgo..."
    }
  ],
  "response_A": "a dog.",
  "response_B": "a cat.",
}
{
  "prompt": "how many animals in echo of the images?",
  "images": [
    {
      "data": "data:image/jpeg;base64,/9j/2wBDAAQDAwQDAwQEAWQFBAQFBgo..."
    }
  ]
}
```

```
{
  },
  {
    "data": "data:image/jpeg;base64,/DKEafe3gihn..."
  }
],
"response_A": "The first image contains one cat and the second image contains one dog",
"response_B": "The first image has one animal and the second has one animal",
}
```

Pour utiliser votre jeu de données personnalisé, modifiez votre formule d'évaluation avec les champs obligatoires suivants, sans modifier le contenu :

```
evaluation:
  task: mm_llm_judge
  strategy: judge
  metric: all
```

Limites

- Un seul fichier `.jsonl` est autorisé par évaluation.
- Le fichier doit suivre strictement le schéma défini.
- Les modèles Nova MM Judge ne prennent en charge que les références d'image.
- Les modèles Nova MM Judge sont les mêmes pour toutes les spécifications d'Amazon Nova Lite.
- Les modèles de juges personnalisés ne sont actuellement pas pris en charge.
- L'URI de l'image Amazon S3 n'est pas prise en charge.
- L'ensemble de données en entrée doit garantir que tous les enregistrements ont une longueur de contexte inférieure à 12 k, à l'exception de l'attribut images.

Juge basé sur des rubriques

Rubric Judge est un modèle LLM-as-a-judge d'évaluation amélioré basé sur Amazon Nova 2.0 Lite. Contrairement au [modèle de juge original](#) qui ne fournit que des verdicts préférentiels (A>B, B>A ou égalité), Rubric Judge génère dynamiquement des critères d'évaluation personnalisés adaptés à chaque invite et attribue des scores granulaires sur plusieurs dimensions.

Fonctionnalités clés :

- Génération de critères dynamiques : crée automatiquement des dimensions d'évaluation pertinentes en fonction de l'invite de saisie
- Notation pondérée : attribue des poids d'importance à chaque critère pour refléter leur importance relative
- Évaluation granulaire : fournit des scores détaillés sur une base binaire (vrai/faux) ou sur une échelle (1 à 5) pour chaque critère
- Indicateurs de qualité : calcule des scores de qualité continus (échelle de 0 à 1) qui quantifient l'ampleur des différences entre les réponses

Exemple de critère généré par le modèle :

```
price_validation:
  description: "The response includes validation to ensure price is a positive value."
  type: "scale"
  weight: 0.3
```

Le modèle évalue les deux réponses par rapport à tous les critères générés, puis utilise ces scores au niveau des critères pour éclairer sa décision de préférence finale.

Rubriques

- [Configuration de la recette](#)
- [Format du jeu de données d'entrée](#)
- [Résultat de l'évaluation](#)
- [Support du modèle de raisonnement](#)

Configuration de la recette

Recette de Rubric Judge

Activez Rubric Judge task : `rubric_llm_judge` en définissant votre recette :

```
run:
  name: nova-eval-job-name                # [MODIFIABLE] Unique
  identifier for your evaluation job
  model_type: amazon.nova-2-lite-v1:0:256k # [FIXED] Rubric Judge model
  type
```



```

    model_name_or_path: "nova-lite-2/prod"                # [FIXED] Path to model
checkpoint or identifier
    replicas: 1                                           # [MODIFIABLE] Number of
replicas for SageMaker Training job
    data_s3_path: ""                                     # [FIXED] Leave empty for
SageMaker Training job
    output_s3_path: ""                                   # [FIXED] Leave empty for
SageMaker Training job

evaluation:
    task: rubric_llm_judge                               # [FIXED] Evaluation task -
enables Rubric Judge
    strategy: judge                                       # [FIXED] Evaluation strategy
    metric: all                                           # [FIXED] Metric calculation
method

inference:
    max_new_tokens: 12000                                # [MODIFIABLE] Maximum tokens
to generate
    top_k: -1                                             # [MODIFIABLE] Top-k sampling
parameter
    top_p: 1.0                                           # [MODIFIABLE] Nucleus sampling
parameter
    temperature: 0                                       # [MODIFIABLE] Sampling
temperature (0 = deterministic)

```

Recette originale de LLM as a Judge (à titre de comparaison)

Le modèle de juge original utilise task: llm_judge :

```

run:
    name: eval-job-name                                  # [MODIFIABLE] Unique
identifier for your evaluation job
    model_type: amazon.nova-micro-v1:0:128k             # [FIXED] Model type
    model_name_or_path: "nova-micro/prod"               # [FIXED] Path to model
checkpoint or identifier
    replicas: 1                                           # [MODIFIABLE] Number of
replicas for SageMaker Training job
    data_s3_path: ""                                     # [FIXED] Leave empty for
SageMaker Training job
    output_s3_path: ""                                   # [FIXED] Leave empty for
SageMaker Training job

evaluation:

```

```

task: llm_judge                                # [FIXED] Original judge task
strategy: judge                                # [FIXED] Evaluation strategy
metric: all                                    # [FIXED] Metric calculation
method

inference:
  max_new_tokens: 12000                        # [MODIFIABLE] Maximum tokens
  to generate
  top_k: -1                                    # [MODIFIABLE] Top-k sampling
  parameter
  top_p: 1.0                                  # [MODIFIABLE] Nucleus sampling
  parameter
  temperature: 0                              # [MODIFIABLE] Sampling
  temperature (0 = deterministic)

```

Format du jeu de données d'entrée

Le format du jeu de données en entrée est identique à celui du [modèle de juge d'origine](#) :

Champs obligatoires :

- **prompt**: chaîne contenant l'invite de saisie et les instructions
- **response_A**: chaîne contenant la sortie du modèle de référence
- **response_B**: chaîne contenant la sortie du modèle personnalisé

Exemple de jeu de données (format JSONL) :

```

{"prompt": "What is the most effective way to combat climate change?", "response_A":
"The most effective way to combat climate change is through a combination of
transitioning to renewable energy sources and implementing strict carbon pricing
policies. This creates economic incentives for businesses to reduce emissions while
promoting clean energy adoption.", "response_B": "We should focus on renewable energy.
Solar and wind power are good. People should drive electric cars. Companies need to
pollute less."}
{"prompt": "Explain how a computer's CPU works", "response_A": "CPU is like brain of
computer. It does math and makes computer work fast. Has lots of tiny parts inside.",
"response_B": "A CPU (Central Processing Unit) functions through a fetch-execute
cycle, where instructions are retrieved from memory, decoded, and executed through its
arithmetic logic unit (ALU). It coordinates with cache memory and registers to process
data efficiently using binary operations."}
{"prompt": "How does photosynthesis work?", "response_A": "Plants do photosynthesis
to make food. They use sunlight and water. It happens in leaves.", "response_B":

```

```
"Photosynthesis is a complex biochemical process where plants convert light energy into chemical energy. They utilize chlorophyll to absorb sunlight, combining CO2 and water to produce glucose and oxygen through a series of chemical reactions in chloroplasts."}
```

Exigences relatives au format :

- Chaque entrée doit être un objet JSON d'une seule ligne
- Séparez les entrées par de nouvelles lignes
- Suivez le nom exact des champs comme indiqué dans les exemples

Résultat de l'évaluation

Structure de sortie

Rubric Judge produit des métriques d'évaluation améliorées par rapport au modèle de juge original :

```
{
  "config_general": {
    "lighteval_sha": "string",
    "num_fewshot_seeds": "int",
    "max_samples": "int | null",
    "job_id": "int",
    "start_time": "float",
    "end_time": "float",
    "total_evaluation_time_secondes": "string",
    "model_name": "string",
    "model_sha": "string",
    "model_dtype": "string | null",
    "model_size": "string"
  },
  "results": {
    "custom|rubric_llm_judge_judge|0": {
      "a_scores": "float",
      "a_scores_stderr": "float",
      "b_scores": "float",
      "b_scores_stderr": "float",
      "ties": "float",
      "ties_stderr": "float",
      "inference_error": "float",
      "inference_error_stderr": "float",
      "score": "float",

```

```

    "score_stderr": "float",
    "weighted_score_A": "float",
    "weighted_score_A_stderr": "float",
    "weighted_score_B": "float",
    "weighted_score_B_stderr": "float",
    "score_margin": "float",
    "score_margin_stderr": "float",
    "winrate": "float",
    "lower_rate": "float",
    "upper_rate": "float"
  }
},
"versions": {
  "custom|rubric_llm_judge_judge|0": "int"
}
}

```

Nouveaux indicateurs dans Rubric Judge

Les six indicateurs suivants sont propres à Rubric Judge et fournissent une évaluation précise de la qualité :

Métrique	Description
Score pondéré A	Note de qualité normalisée moyenne pour Response_A pour tous les critères d'évaluation générés par le modèle. Les scores sont pondérés en fonction de l'importance du critère et normalisés sur une échelle de 0 à 1 (plus haut = meilleure qualité)
Score_pondéré_A_Stderr	Erreur type de la moyenne pour Weighted_Score_A, indiquant une incertitude statistique
Score pondéré B	Score de qualité normalisé moyen pour Response_B pour tous les critères d'évaluation générés par le modèle. Les scores sont pondérés en fonction de l'importance du critère et normalisés sur une échelle de 0 à 1 (plus haut = meilleure qualité)

Métrique	Description
Score_pondéré_B_Stderr	Erreur type de la moyenne pour Weighted_Score_B, indiquant une incertitude statistique
score_margin	Différence entre les scores pondérés (calculée comme $\text{Weighted\_Score\_A} - \text{Weighted\_Score\_B}$). Plage : -1,0 à 1,0. Positif = Response_A est meilleur ; négatif = Response_B est meilleur ; proche de zéro = qualité similaire
score_margin_stderr	Erreur type de la moyenne pour score_margin, indiquant une incertitude dans la mesure de la différence de qualité

Comprendre les indicateurs de score pondérés

Objectif : Les scores pondérés fournissent des mesures de qualité continues qui complètent les verdicts de préférence binaires, permettant ainsi de mieux comprendre les performances du modèle.

Principales différences par rapport au juge initial :

- Juge d'origine : affiche uniquement des préférences discrètes ($A > B$, $B > A$, $A = B$)
- Rubric Judge : affiche à la fois les préférences ET les scores de qualité continus (échelle de 0 à 1) sur la base de critères personnalisés

Interprétation de score_margin :

- $\text{score\_margin} = -0.128$: Response_B a obtenu 12,8 points de pourcentage de plus que Response_A
- $|\text{score\_margin}| < 0.1$: faible différence de qualité (décision serrée)
- $|\text{score\_margin}| > 0.2$: différence de qualité claire (décision prise en toute confiance)

Cas d'utilisation :

- Amélioration du modèle : identifiez les domaines spécifiques dans lesquels votre modèle est sous-performant
- Quantification de la qualité : mesurez l'ampleur des écarts de performance, pas seulement win/loss les ratios
- Évaluation de la confiance : faites la distinction entre des décisions serrées et des différences de qualité claires

Important

Les verdicts finaux sont toujours basés sur les étiquettes de préférence explicites du modèle du juge afin de préserver le raisonnement holistique et de garantir une atténuation appropriée des biais de position grâce forward/backward à l'évaluation. Les scores pondérés servent d'outils d'observabilité et ne remplacent pas le verdict principal.

Méthodologie de calcul

Les scores pondérés sont calculés selon le processus suivant :

- Extraire les données des critères : analyse la sortie YAML du juge pour extraire les scores et les pondérations des critères
- Normaliser les scores :
 - Critères de type échelle (1-5) : normaliser à 0-1 en calculant $(\text{score} - 1) / 4$
 - Critères binaires (vrai/faux) : Convertir en 1,0/0,0
- Appliquer des pondérations : multipliez chaque score normalisé par son critère de pondération
- Agrégat : additionnez tous les scores pondérés pour chaque réponse
- Calculer la marge : Calculer $\text{score\_margin} = \text{weighted\_score\_A} - \text{weighted\_score\_B}$

Exemple : si Response_A a une somme pondérée de 0,65 et Response_B a une somme pondérée de 0,78, ce `score_margin` serait -0,13, ce qui indique que Response_B est de 13 points de pourcentage plus élevée en termes de qualité pour tous les critères pondérés.

Support du modèle de raisonnement

La prise en charge des modèles de raisonnement permet une évaluation à l'aide de modèles Amazon Nova capables de raisonner qui exécutent un raisonnement interne explicite avant de générer des

réponses finales. Cette fonctionnalité utilise le contrôle au niveau de l'API via le `reasoning_effort` paramètre pour activer ou désactiver dynamiquement la fonctionnalité de raisonnement, ce qui peut améliorer la qualité des réponses pour les tâches analytiques complexes.

Modèles pris en charge :

- `amazon.nova-2-lite-v 1:0:256 k`

Configuration de la recette

Activez le raisonnement en ajoutant le `reasoning_effort` paramètre à la `inference` section de votre recette :

```
run:
  name: eval-job-name                                # [MODIFIABLE] Unique
  identifier for your evaluation job
  model_type: amazon.nova-2-lite-v1:0:256k           # [FIXED] Must be a reasoning-
supported model
  model_name_or_path: nova-lite-2/prod                # [FIXED] Path to model
  checkpoint or identifier
  replicas: 1                                          # [MODIFIABLE] Number of
replicas for SageMaker Training job
  data_s3_path: ""                                   # [MODIFIABLE] Leave empty for
SageMaker Training job; optional for SageMaker SageMaker HyperPod job
  output_s3_path: ""                                 # [MODIFIABLE] Output path for
SageMaker SageMaker HyperPod job (not compatible with SageMaker Training jobs)

evaluation:
  task: mmlu                                          # [MODIFIABLE] Evaluation task
  strategy: generate                                  # [MODIFIABLE] Evaluation
strategy
  metric: all                                         # [MODIFIABLE] Metric
calculation method

inference:
  reasoning_effort: high                             # [MODIFIABLE] Enables
reasoning mode; options: low/medium/high or null to disable
  max_new_tokens: 200                                # [MODIFIABLE] Maximum tokens
to generate
  top_k: 50                                           # [MODIFIABLE] Top-k sampling
parameter
```

```

top_p: 1.0                                # [MODIFIABLE] Nucleus
sampling parameter
temperature: 0                            # [MODIFIABLE] Sampling
temperature (0 = deterministic)

```

Utilisation du paramètre reasoning_effort

Le reasoning_effort paramètre contrôle le comportement de raisonnement des modèles capables de raisonner.

Prérequis :

- **Compatibilité des modèles :** définie reasoning_effort uniquement lorsqu'un modèle capable de raisonner est spécifié (actuellement) amazon.nova-2-lite-v1:0:256k
- **Gestion des erreurs :** l'utilisation reasoning_effort avec des modèles non pris en charge échouera avec ConfigValidationError: "Reasoning mode is enabled but model '{model_type}' does not support reasoning. Please use a reasoning-capable model or disable reasoning mode."

Options disponibles :

Option	Comportement	Limite de jetons	Cas d'utilisation
null (par défaut)	Désactive le mode raisonnement	N/A	Évaluation standard sans frais de raisonnement
bas	Permet de raisonner avec des contraintes	4 000 jetons pour le raisonnement interne	Scénarios nécessitent un raisonnement concis ; optimisations en termes de rapidité et de coût
haut	Permet de raisonner sans contraintes	Aucune limite symbolique au raisonnement interne	Problèmes complexes nécessitant une analyse et un step-by-step raisonnement approfondis

Quand activer le raisonnement

Utilisez le mode raisonnement (**lowmedium**, ou **high**) pour :

- Tâches complexes de résolution de problèmes (mathématiques, énigmes logiques, codage)
- Questions analytiques en plusieurs étapes nécessitant un raisonnement intermédiaire
- Tâches pour lesquelles des explications ou des step-by-step réflexions détaillées améliorent la précision
- Scénarios dans lesquels la qualité de réponse est privilégiée par rapport à la vitesse

Utilisez le mode non raisonné (omettre le paramètre) pour :

- Questions et réponses simples ou questions factuelles
- Tâches d'écriture créative
- Lorsque des temps de réponse plus rapides sont essentiels
- Analyse comparative des performances où les frais de raisonnement doivent être exclus
- Optimisation des coûts lorsque le raisonnement n'améliore pas les performances des tâches

Résolution des problèmes

Erreur : « Le mode raisonnement est activé mais le modèle ne prend pas en charge le raisonnement »

Cause : le `reasoning_effort` paramètre est défini sur une valeur non nulle, mais la valeur spécifiée `model_type` ne permet pas de raisonner.

Résolution :

- Vérifiez que le type de votre modèle est `amazon.nova-2-lite-v1:0:256k`
- Si vous utilisez un modèle différent, passez à un modèle capable de raisonner ou supprimez le `reasoning_effort` paramètre de votre recette

Démarrage d'une tâche d'évaluation

Vous trouverez ci-dessous une suggestion de configuration du type d'instance d'évaluation et du type de modèle :

```
# Install Dependencies (Helm - https://helm.sh/docs/intro/install/)
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh
rm -f ./get_helm.sh

# Install the SageMaker HyperPod CLI
git clone --recurse-submodules https://github.com/aws/sagemaker-hyperpod-cli.git
git checkout -b release_v2
cd sagemaker-hyperpod-cli
pip install .

# Verify the installation
hyperpod --help

# Connect to a SageMaker HyperPod Cluster
hyperpod connect-cluster --cluster-name cluster-name

# Submit the Job using the recipe for eval
# Namespace by default should be kubeflow
hyperpod start-job [--namespace namespace] --recipe evaluation/nova/
nova_micro_p5_48xl_general_text_benchmark_eval --override-parameters \
'{
    "instance_type": "p5d.48xlarge",
    "container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-
HP-Eval-V2-latest",
    "recipes.run.name": custom-run-name,
    "recipes.run.model_type": model_type,
    "recipes.run.model_name_or_path": "model name or finetune checkpoint s3uri",
    "recipes.run.data_s3_path": "s3 for input data only for genqa and llm_judge, must be
full S3 path that include filename",
}'

# List jobs
hyperpod list-jobs [--namespace namespace] [--all-namespaces]

# Getting Job details
hyperpod get-job --job-name job-name [--namespace namespace] [--verbose]

# Listing Pods
hyperpod list-pods --job-name job-name --namespace namespace
```

```
# Cancel Job
hyperpod cancel-job --job-name job-name [--namespace namespace]
```

Vous devriez également être en mesure de consulter le statut de la tâche via la console du cluster Amazon EKS.

Accès aux résultats d'évaluation et analyse

Une fois votre tâche d'évaluation terminée avec succès, vous pouvez accéder aux résultats et les analyser à l'aide des informations de cette section. Sur la base de ce qui est défini dans la formule `output_s3_path` (par exemple, `s3://output_path/`), la structure de sortie est la suivante :

```
job_name/
### eval-result/
#   ### results_[timestamp].json
#   ### inference_output.jsonl (only present for gen_qa)
#   ### details/
#       ### model/
#           ### execution-date-time/
#               ###details_task_name_#_datetime.parquet
### tensorboard-results/
    ### eval/
        ### events.out.tfevents.[timestamp]
```

Les résultats des métriques sont stockés dans l'emplacement de sortie S3 `s3://output_path/job_name/eval-result/result-timestamp.json` spécifié.

Les résultats Tensorboard sont stockés dans le chemin S3 `s3://output_path/job_name/eval-tensorboard-result/eval/event.out.tfevents.epoch+ip`.

Toutes les sorties d'inférence, à l'exception de `llm_judge` et de `strong_reject`, sont stockées dans le chemin S3 `s3://output_path/job_name/eval-result/details/model/taskname.parquet`.

Pour `gen_qa`, le fichier `inference_output.jsonl` contient les champs suivants pour chaque objet JSON :

- `invite` : le message final soumis au modèle
- `inférence` : la sortie d'inférence brute du modèle
- `gold` - La réponse cible de l'ensemble de données en entrée

- **metadata** - La chaîne de métadonnées de l'ensemble de données en entrée, si elle est fournie

Pour visualiser vos métriques d'évaluation dans Tensorboard, suivez les étapes ci-après :

1. Accédez à SageMaker AI Tensorboard.
2. Sélectionnez les dossiers S3.
3. Ajoutez le chemin de votre dossier S3, par exemple `s3://output_path/job-name/eval-tensorboard-result/eval`.
4. Patientez jusqu'à ce que la synchronisation soit terminée.

Les séries temporelles, les scalaires et les visualisations de texte sont disponibles.

Nous recommandons les bonnes pratiques suivantes :

- Organisez vos chemins de sortie par modèle et par type de référence.
- Maintenez des conventions de dénomination cohérentes pour faciliter le suivi.
- Enregistrez les résultats extraits dans un emplacement sécurisé.
- Surveillez l'état de TensorBoard synchronisation pour un chargement de données réussi.

Vous pouvez trouver les journaux des erreurs de SageMaker HyperPod travail dans le groupe de CloudWatch journaux/`aws/sagemaker/Clusters/cluster-id`.

Format de sortie de probabilité logarithmique

Lorsque `top_logprobs` est configuré dans vos paramètres d'inférence, le résultat de l'évaluation inclut les probabilités logarithmiques au niveau des jetons dans les fichiers parquet. Chaque position de jeton contient un dictionnaire des meilleurs jetons candidats avec leurs probabilités logarithmiques dans la structure suivante :

```
{
  "Ġint": {"logprob_value": -17.8125, "decoded_value": " int"},
  "Ġthe": {"logprob_value": -2.345, "decoded_value": " the"}
}
```

Chaque entrée de jeton contient :

- **logprob_value**: valeur de probabilité logarithmique pour le jeton

- `decoded_value`: la représentation sous forme de chaîne décodée lisible par l'homme du jeton

Le jeton tokenizer brut est utilisé comme clé de dictionnaire pour garantir l'unicité, tout en `decoded_value` fournissant une interprétation lisible.

Évaluation RFT

Note

L'évaluation via des fonctions de récompense à distance dans votre propre AWS environnement n'est disponible que si vous êtes client d'Amazon Nova Forge.

Important

Le champ `rl_env` de configuration est utilisé exclusivement pour l'évaluation, pas pour la formation. Pendant l'entraînement, vous configurez les fonctions de récompense en utilisant `reward_lambda_arn` (un tour) ou une infrastructure BYOO avec `rollout.delegate: true` (plusieurs tours).

Qu'est-ce que l'évaluation RFT ?

L'évaluation RFT vous permet d'évaluer les performances de votre modèle à l'aide de fonctions de récompense personnalisées avant, pendant ou après l'entraînement par renforcement. Contrairement aux évaluations standard qui utilisent des métriques prédéfinies, RFT Evaluation vous permet de définir vos propres critères de réussite grâce à une fonction Lambda qui note les résultats du modèle en fonction de vos besoins spécifiques.

Pourquoi évaluer avec RFT ?

L'évaluation est cruciale pour déterminer si le processus de réglage précis des RL a :

- Alignement amélioré du modèle avec votre cas d'utilisation spécifique et vos valeurs humaines
- Fonctionnalités du modèle maintenues ou améliorées pour les tâches clés
- Effets secondaires imprévus évités tels qu'une baisse de la factualité, une verbosité accrue ou une dégradation des performances sur d'autres tâches
- Répond à vos critères de réussite personnalisés tels que définis par votre fonction de récompense

Quand utiliser l'évaluation RFT

Utilisez l'évaluation RFT dans les scénarios suivants :

- Avant la formation RFT : établissez des mesures de référence sur votre jeu de données d'évaluation
- Pendant la formation RFT : surveillez la progression de la formation avec des points de contrôle intermédiaires
- Après la formation RFT : vérifiez que le modèle final répond à vos exigences
- Comparaison de modèles : évaluez plusieurs versions de modèles à l'aide de critères de récompense cohérents

Note

Utilisez l'évaluation RFT lorsque vous avez besoin de mesures personnalisées et spécifiques à un domaine. Pour une évaluation à usage général (précision, perplexité, BLEU), utilisez des méthodes d'évaluation standard.

Rubriques

- [Exigences de format de données](#)
- [Préparation de votre recette d'évaluation](#)
- [Fonctions de récompense prédéfinies](#)
- [Création de votre fonction de récompense](#)
- [Autorisations IAM](#)
- [Exécution de la tâche d'évaluation](#)
- [Comprendre les résultats de l'évaluation](#)

Exigences de format de données

Structure des données d'entrée

Les données d'entrée d'évaluation RFT doivent suivre le format OpenAI Reinforcement Fine-Tuning. Chaque exemple est un objet JSON contenant :

- messages: éventail de tournures de conversation system et de rôles user

- Autres métadonnées facultatives, par exemple `reference_answer`

Exemple de format de données

L'exemple suivant montre le format requis :

```
{
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": "Solve for x. Return only JSON like {\"x\": <number>}. Equation: 2x + 5 = 13"
        }
      ]
    }
  ],
  "reference_answer": {
    "x": 4
  }
}
```

Limitations actuelles

Les limites suivantes s'appliquent à l'évaluation RFT :

- Texte uniquement : aucune entrée multimodale (images, audio, vidéo) n'est prise en charge
- Conversations à tour unique : ne prend en charge que les messages d'un seul utilisateur (pas de dialogues à plusieurs tours)
- Format JSON : les données d'entrée doivent être au format JSONL (un objet JSON par ligne)
- Sorties du modèle : L'évaluation est effectuée sur les complétions générées à partir du modèle spécifié

Préparation de votre recette d'évaluation

Exemple de configuration de recette

L'exemple suivant montre une recette d'évaluation RFT complète :

```

run:
  name: nova-lite-rft-eval-job
  model_type: amazon.nova-lite-v1:0:300k
  model_name_or_path: s3://escrow_bucket/model_location      # [MODIFIABLE] S3 path to
your model or model identifier
  replicas: 1                                                # [MODIFIABLE] For SageMaker
Training jobs only; fixed for SageMaker HyperPod jobs
  data_s3_path: ""                                          # [REQUIRED FOR HYPERPOD]
Leave empty for SageMaker Training jobs
  output_s3_path: ""                                       # [REQUIRED] Output artifact
S3 path for evaluation results

evaluation:
  task: rft_eval                                           # [FIXED] Do not modify
  strategy: rft_eval                                       # [FIXED] Do not modify
  metric: all                                              # [FIXED] Do not modify

# Inference Configuration
inference:
  max_new_tokens: 8196                                     # [MODIFIABLE] Maximum tokens
to generate
  top_k: -1                                                # [MODIFIABLE] Top-k sampling
parameter
  top_p: 1.0                                              # [MODIFIABLE] Nucleus
sampling parameter
  temperature: 0                                           # [MODIFIABLE] Sampling
temperature (0 = deterministic)
  top_logprobs: 0

# Evaluation Environment Configuration (NOT used in training)
rl_env:
  reward_lambda_arn: arn:aws:lambda:<region>:<account_id>:function:<reward-function-
name>

```

Fonctions de récompense prédéfinies

Deux fonctions de récompense prédéfinies (`prime_codeetprime_math`) sont disponibles sous forme de couche Lambda pour une intégration facile à vos fonctions Lambda RFT.

Présentation de

Ces fonctions prédéfinies fournissent des capacités out-of-the-box d'évaluation pour :

- `prime_code` : génération de code et évaluation de l'exactitude
- `prime_math` : évaluation du raisonnement mathématique et de la résolution de problèmes

Configuration rapide

Pour utiliser les fonctions de récompense prédéfinies :

1. [Téléchargez la couche Lambda depuis les versions nova-custom-eval-sdk](#)
2. Publiez une couche Lambda à l'aide de la CLI AWS :

```
aws lambda publish-layer-version \
  --layer-name preset-function-layer \
  --description "Preset reward function layer with dependencies" \
  --zip-file fileb://universal_reward_layer.zip \
  --compatible-runtimes python3.9 python3.10 python3.11 python3.12 \
  --compatible-architectures x86_64 arm64
```

3. Ajoutez la couche à votre fonction Lambda dans la AWS console (preset-function-layersélectionnez la couche personnalisée et ajoutez également AWSSDKPandas-Python 312 pour les dépendances numpy)
4. Importez et utilisez dans votre code Lambda :

```
from prime_code import compute_score # For code evaluation
from prime_math import compute_score # For math evaluation
```

fonction `prime_code`

Objectif : Évalue les tâches de génération de code Python en exécutant le code par rapport à des cas de test et en mesurant l'exactitude.

Exemple de format de jeu de données d'entrée issu de l'évaluation :

```
{"messages":[{"role":"user","content":"Write a function that returns the sum of two numbers."}], "reference_answer":{"inputs":["3\n5", "10\n-2", "0\n0"], "outputs":["8", "8", "0"]}}
```

```
{"messages":[{"role":"user","content":"Write a function to check if a number is even."}], "reference_answer":{"inputs":["4", "7", "0", "-2"], "outputs":["True", "False", "True", "True"]}}
```

Caractéristiques principales :

- Extraction automatique du code à partir de blocs de code Markdown
- Détection des fonctions et tests basés sur les appels
- Exécution des scénarios de test avec protection contre le délai d'expiration
- Validation de syntaxe et contrôles de compilation
- Signalement détaillé des erreurs avec retraçage

fonction prime_math

Objectif : Évalue les capacités de raisonnement mathématique et de résolution de problèmes à l'aide d'un support mathématique symbolique.

Format d'entrée :

```
{"messages":[{"role":"user","content":"What is the derivative of x^2 + 3x?."}], "reference_answer":"2*x + 3"}
```

Caractéristiques principales :

- Évaluation mathématique symbolique à l'aide de SymPy
- Formats de réponse multiples (LaTeX, texte brut, symbolique)
- Vérification de l'équivalence mathématique
- Normalisation et simplification des expressions

Bonnes pratiques

Suivez ces bonnes pratiques lorsque vous utilisez des fonctions de récompense prédéfinies :

- Utilisez les types de données appropriés dans les cas de test (entiers contre chaînes, booléens contre « Vrai »)
- Fournir des signatures de fonction claires en cas de problèmes de code
- Inclure les cas extrêmes dans les entrées de test (zéro, nombres négatifs, entrées vides)
- Formater les expressions mathématiques de manière cohérente dans les réponses de référence
- Testez votre fonction de récompense avec des exemples de données avant le déploiement

Création de votre fonction de récompense

ARN Lambda

Vous devez vous référer au format suivant pour l'ARN Lambda :

```
"arn:aws:lambda:*:*:function:*SageMaker*"
```

Si le Lambda ne possède pas ce schéma de dénomination, la tâche échouera avec cette erreur :

```
[ERROR] Unexpected error: lambda_arn must contain one of: ['SageMaker', 'sagemaker', 'Sagemaker'] when running on SMHP platform (Key: lambda_arn)
```

Structure de la fonction Lambda

Votre fonction Lambda reçoit des lots de sorties de modèles et renvoie des scores de récompense.

Vous trouverez ci-dessous un exemple de mise en œuvre :

```
from typing import List, Any
import json
import re
from dataclasses import asdict, dataclass

@dataclass
class MetricResult:
    """Individual metric result."""
    name: str
    value: float
    type: str

@dataclass
class RewardOutput:
    """Reward service output."""
    id: str
    aggregate_reward_score: float
    metrics_list: List[MetricResult]

def lambda_handler(event, context):
    """ Main lambda handler """
    return lambda_grader(event)
```

```
def lambda_grader(samples: list[dict]) -> list[dict]:
    """ Core grader function """
    scores: List[RewardOutput] = []

    for sample in samples:
        print("Sample: ", json.dumps(sample, indent=2))

        # Extract components
        idx = sample.get("id", "no id")
        if not idx or idx == "no id":
            print(f"ID is None/empty for sample: {sample}")

        ground_truth = sample.get("reference_answer")

        if "messages" not in sample:
            print(f"Messages is None/empty for id: {idx}")
            continue

        if ground_truth is None:
            print(f"No answer found in ground truth for id: {idx}")
            continue

        # Get model's response (last turn is assistant turn)
        last_message = sample["messages"][-1]

        if last_message["role"] != "nova_assistant":
            print(f>Last message is not from assistant for id: {idx}")
            continue

        if "content" not in last_message:
            print(f"Completion text is empty for id: {idx}")
            continue

        model_text = last_message["content"]

        # --- Actual scoring logic (lexical overlap) ---
        ground_truth_text = _extract_ground_truth_text(ground_truth)

        # Calculate main score and individual metrics
        overlap_score = _lexical_overlap_score(model_text, ground_truth_text)

        # Create two separate metrics as in the first implementation
```

```

    accuracy_score = overlap_score # Use overlap as accuracy
    fluency_score = _calculate_fluency(model_text) # New function for fluency

    # Create individual metrics
    metrics_list = [
        MetricResult(name="accuracy", value=accuracy_score, type="Metric"),
        MetricResult(name="fluency", value=fluency_score, type="Reward")
    ]

    ro = RewardOutput(
        id=idx,
        aggregate_reward_score=overlap_score,
        metrics_list=metrics_list
    )

    print(f"Response for id: {idx} is {ro}")
    scores.append(ro)

# Convert to dict format
result = []
for score in scores:
    result.append({
        "id": score.id,
        "aggregate_reward_score": score.aggregate_reward_score,
        "metrics_list": [asdict(metric) for metric in score.metrics_list]
    })

return result

def _extract_ground_truth_text(ground_truth: Any) -> str:
    """
    Turn the `ground_truth` field into a plain string.
    """
    if isinstance(ground_truth, str):
        return ground_truth

    if isinstance(ground_truth, dict):
        # Common patterns: { "explanation": "...", "answer": "..." }
        if "explanation" in ground_truth and isinstance(ground_truth["explanation"],
str):
            return ground_truth["explanation"]
        if "answer" in ground_truth and isinstance(ground_truth["answer"], str):
            return ground_truth["answer"]

```

```
# Fallback: stringify the whole dict
return json.dumps(ground_truth, ensure_ascii=False)

# Fallback: stringify anything else
return str(ground_truth)

def _tokenize(text: str) -> List[str]:
    # Very simple tokenizer: lowercase + alphanumeric word chunks
    return re.findall(r"\w+", text.lower())

def _lexical_overlap_score(model_text: str, ground_truth_text: str) -> float:
    """
    Simple lexical overlap score in [0, 1]:
    score = |tokens(model) ∩ tokens(gt)| / |tokens(gt)|
    """
    gt_tokens = _tokenize(ground_truth_text)
    model_tokens = _tokenize(model_text)

    if not gt_tokens:
        return 0.0

    gt_set = set(gt_tokens)
    model_set = set(model_tokens)
    common = gt_set & model_set

    return len(common) / len(gt_set)

def _calculate_fluency(text: str) -> float:
    """
    Calculate a simple fluency score based on:
    - Average word length
    - Text length
    - Sentence structure

    Returns a score between 0 and 1.
    """
    # Simple implementation - could be enhanced with more sophisticated NLP
    words = _tokenize(text)

    if not words:
        return 0.0
```

```

# Average word length normalized to [0,1] range
# Assumption: average English word is ~5 chars, so normalize around that
avg_word_len = sum(len(word) for word in words) / len(words)
word_len_score = min(avg_word_len / 10, 1.0)

# Text length score - favor reasonable length responses
ideal_length = 100 # words
length_score = min(len(words) / ideal_length, 1.0)

# Simple sentence structure check (periods, question marks, etc.)
sentence_count = len(re.findall(r'[.!?]+', text)) + 1
sentence_ratio = min(sentence_count / (len(words) / 15), 1.0)

# Combine scores
fluency_score = (word_len_score + length_score + sentence_ratio) / 3

return fluency_score

```

Format de demande Lambda

Votre fonction Lambda reçoit des données au format suivant :

```

[
  {
    "id": "sample-001",
    "messages": [
      {
        "role": "user",
        "content": [
          {
            "type": "text",
            "text": "Do you have a dedicated security team?"
          }
        ]
      },
      {
        "role": "nova_assistant",
        "content": [
          {
            "type": "text",
            "text": "As an AI developed by Company, I don't have a dedicated security team in the traditional sense. However, the development and deployment of AI systems

```

```

like me involve extensive security measures, including data encryption, user privacy
protection, and other safeguards to ensure safe and responsible use."
    }
  ]
}
],
"reference_answer": {
  "compliant": "No",
  "explanation": "As an AI developed by Company, I do not have a traditional
security team. However, the deployment involves stringent safety measures, such as
encryption and privacy safeguards."
}
}
]

```

Note

La structure du message inclut le contenu tableau imbriqué, correspondant au format des données d'entrée. Le dernier message avec un rôle nova_assistant contient la réponse générée par le modèle.

Format de réponse Lambda

Votre fonction Lambda doit renvoyer des données au format suivant :

```

[
  {
    "id": "sample-001",
    "aggregate_reward_score": 0.75,
    "metrics_list": [
      {
        "name": "accuracy",
        "value": 0.85,
        "type": "Metric"
      },
      {
        "name": "fluency",
        "value": 0.90,
        "type": "Reward"
      }
    ]
  }
]

```



```
}  
]
```

Champs de réponse :

- **id**: Doit correspondre à l'identifiant de l'échantillon d'entrée
- **aggregate_reward_score**: score global (généralement de 0,0 à 1,0)
- **metrics_list**: Tableau de mesures individuelles avec :
 - **name**: Identifiant métrique (par exemple, « précision », « fluidité »)
 - **value**: score métrique (généralement de 0,0 à 1,0)
 - **type**: soit « Métrique » (pour les rapports), soit « Récompense » (utilisée dans le cadre de la formation)

Autorisations IAM

Autorisations requises

Votre rôle d'exécution SageMaker AI doit être autorisé à invoquer votre fonction Lambda. Ajoutez cette politique à votre rôle d'exécution de l' SageMaker IA :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "lambda:InvokeFunction"  
      ],  
      "Resource": "arn:aws:lambda:region:account-id:function:function-name"  
    }  
  ]  
}
```

Rôle d'exécution Lambda

Le rôle d'exécution de votre fonction Lambda nécessite des autorisations d'exécution Lambda de base :

```
{  
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "logs:CreateLogGroup",  
      "logs:CreateLogStream",  
      "logs:PutLogEvents"  
    ],  
    "Resource": "arn:aws:logs:*:*:*"  
  }  
]  
}
```

Autorisations supplémentaires : si votre fonction Lambda accède à d'autres AWS services (par exemple, Amazon S3 pour les données de référence, DynamoDB pour la journalisation), ajoutez ces autorisations au rôle d'exécution Lambda.

Exécution de la tâche d'évaluation

1. Préparez vos données

- Formatez vos données d'évaluation conformément aux exigences en matière de format de données
- Téléchargez votre fichier JSONL sur Amazon S3 : `s3://your-bucket/eval-data/eval_data.jsonl`

2. Configurez votre recette

Mettez à jour l'exemple de recette avec votre configuration :

- Réglez `model_name_or_path` selon l'emplacement de votre modèle
- Réglez `lambda_arn` sur l'ARN de votre fonction de récompense
- Réglez `output_s3_path` sur l'emplacement de sortie souhaité
- Ajustez `inference` les paramètres selon les besoins

Enregistrez la recette sous `rft_eval_recipe.yaml`

3. Exécuter l'évaluation

Exécutez le travail d'évaluation à l'aide du bloc-notes fourni : [carnet d'évaluation du modèle Nova](#)

4. Surveiller les progrès

Surveillez votre travail d'évaluation par le biais de :

- SageMaker Console AI : vérifiez l'état des tâches et les journaux
- CloudWatch Journaux : Afficher les journaux d'exécution détaillés
- Lambda Logs : résolution des problèmes liés à la fonction de récompense

Comprendre les résultats de l'évaluation

Format de sortie

La tâche d'évaluation produit les résultats vers l'emplacement Amazon S3 que vous avez spécifié au format JSONL. Chaque ligne contient les résultats de l'évaluation pour un échantillon :

```
{
  "id": "sample-001",
  "aggregate_reward_score": 0.75,
  "metrics_list": [
    {
      "name": "accuracy",
      "value": 0.85,
      "type": "Metric"
    },
    {
      "name": "fluency",
      "value": 0.90,
      "type": "Reward"
    }
  ]
}
```

Note

La sortie du Job d'évaluation RFT est identique au format Lambda Response. Le service d'évaluation passe en revue la réponse de votre fonction Lambda sans modification, garantissant ainsi la cohérence entre vos calculs de récompenses et les résultats finaux.

Interprétation des résultats

Score de récompense global :

- Plage : généralement de 0,0 (pire) à 1,0 (meilleur), mais cela dépend de votre implémentation
- Objectif : numéro unique résumant les performances globales
- Utilisation : comparer les modèles, suivre les améliorations au fil de la formation

Métriques individuelles :

- Type de métrique : métriques informatives pour l'analyse
- Type de récompense : mesures utilisées lors de l'entraînement RFT
- Interprétation : des valeurs plus élevées indiquent généralement de meilleures performances (sauf si vous concevez des métriques inverses)

Critères de performance

Ce qui constitue une « bonne » performance dépend de votre cas d'utilisation :

Fourchette de scores	Interprétation	Action
0,8 - 1,0	Excellente	Modèle prêt pour le déploiement
0,6 - 0,8	Bon	Des améliorations mineures peuvent être bénéfiques
0,4 - 0,6	Fair	Une amélioration significative est nécessaire
0,0 - 0,4	Pauvre	Passez en revue les données d'entraînement et la fonction de récompense

Important

Il s'agit de directives générales. Définissez vos propres seuils en fonction des exigences commerciales, des performances du modèle de référence, des contraintes spécifiques au domaine et de l'analyse coûts-avantages de la formation continue.

Suivi des HyperPod tâches avec MLflow

Vous pouvez l'utiliser MLflow pour suivre et surveiller vos tâches de formation sur SageMaker HyperPod. Suivez ces étapes pour le configurer MLflow et le connecter à vos recettes d'entraînement.

Créez l' MLflow application

Exemple de AWS CLI commande

```
aws sagemaker-mlflow create-mlflow-app \  
  --name <app-name> \  
  --artifact-store-uri <s3-bucket-name> \  
  --role-arn <role-arn> \  
  --region <region-name>
```

Exemple de sortie

```
{  
  "Arn": "arn:aws:sagemaker:us-east-1:111122223333:mlflow-app/app-LGZE0Z2UY4NZ"  
}
```

Générer une URL pré-signée

Exemple de AWS CLI commande

```
aws sagemaker-mlflow create-presigned-mlflow-app-url \  
  --arn <app-arn> \  
  --region <region-name> \  
  --output text
```

Exemple de sortie

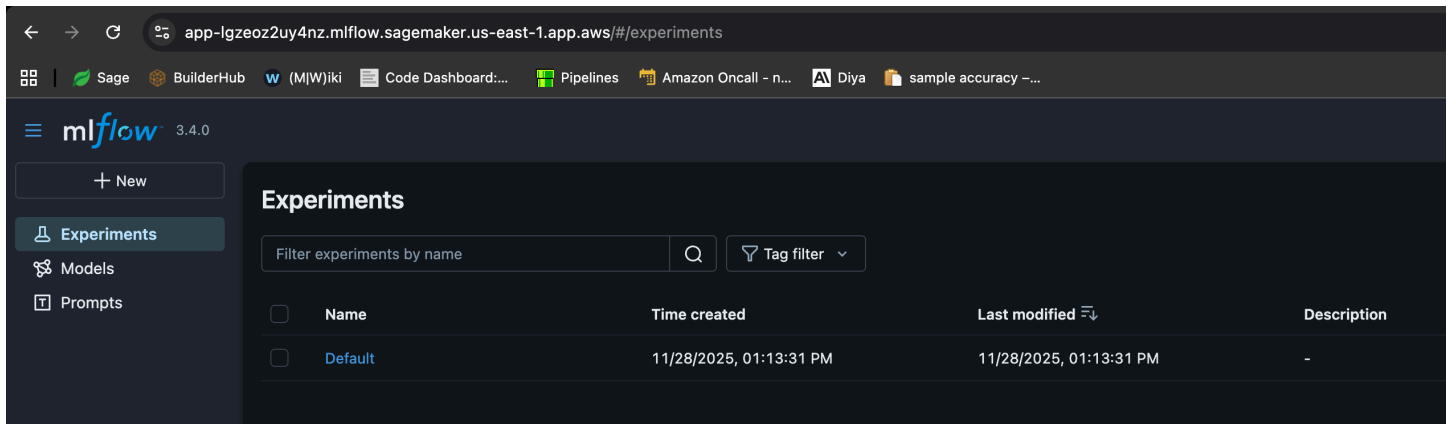
```
https://app-LGZE0Z2UY4NZ.mlflow.sagemaker.us-east-1.app.aws/auth?  
authToken=eyJhbGciOiJIUzI1NiJ9.eyJhdXRoVG9rZW5JZCI6IkxETVBPUyIsImZhc0NyZWRLbnRpYWxzIjoiQWdWNGhD
```

Ouvrez l'URL présignée et consultez l'application

Cliquez sur

```
https://app-LGZE0Z2UY4NZ.mlflow.sagemaker.us-east-1.app.aws/auth?
authToken=eyJhbGciOiJIUzI1NiJ9.eyJhdXRoVG9rZW5JZCI6IkxETVBPUyIsImZhc0NyZWRLbnRpYWxzIjoiQWdWNGhD
```

Afficher

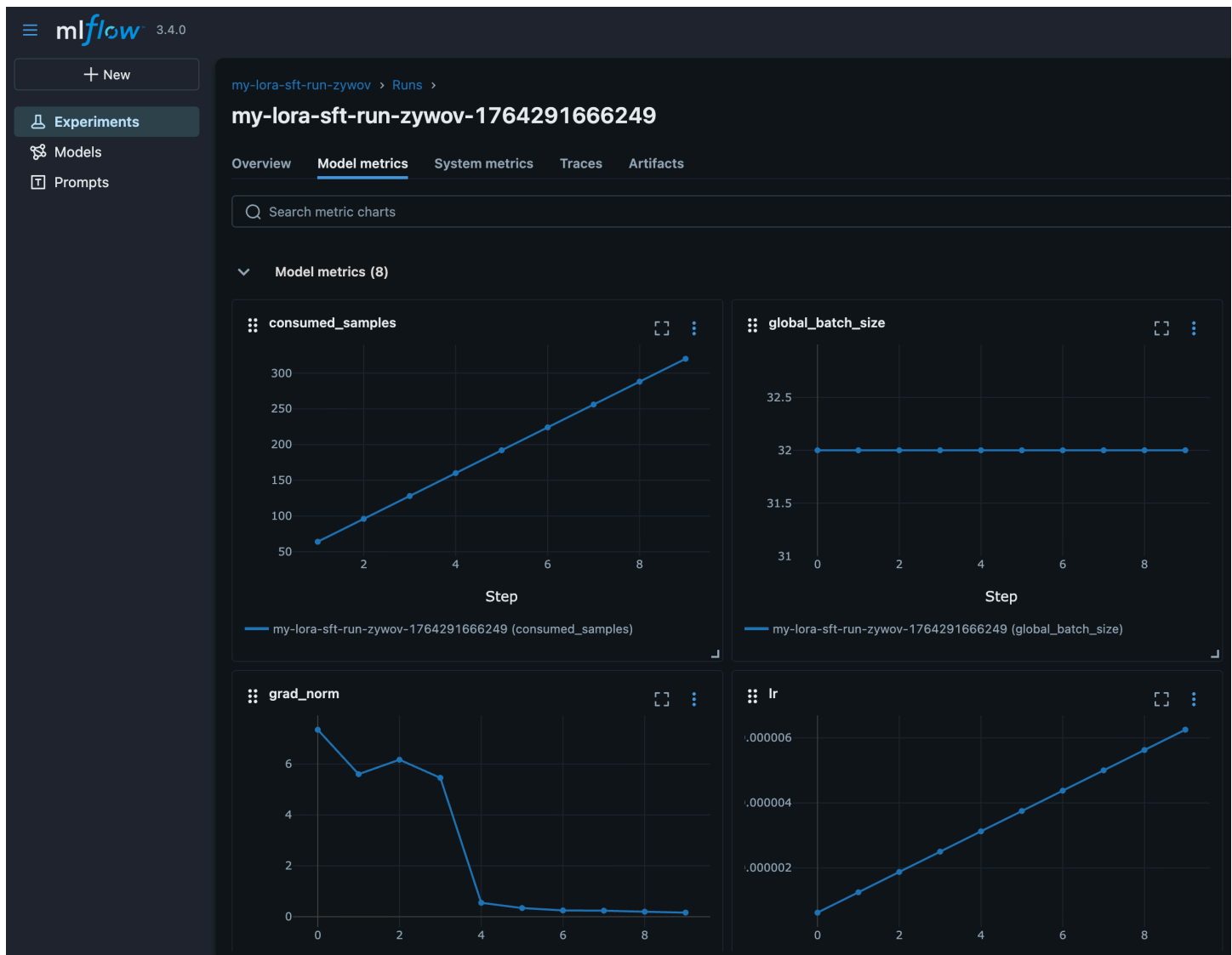


Passez à la recette sous le bloc d'exécution de votre SageMaker HyperPod recette

Formule

```
run
    mlflow_tracking_uri: arn:aws:sagemaker:us-east-1:111122223333:mlflow-app/app-
    LGZE0Z2UY4NZ
```

Afficher



Entraînement itératif

Présentation de

L'entraînement itératif est le processus qui consiste à peaufiner un modèle à plusieurs reprises au cours de plusieurs cycles d'entraînement utilisant différentes méthodes d'entraînement (formation, évaluation, analyse des erreurs, ajustement data/objectives/hyperparameters), chaque cycle commençant par le point de contrôle précédent. Cette approche vous permet de cibler systématiquement les modes de défaillance des modèles, d'intégrer des exemples sélectionnés pour remédier à des faiblesses spécifiques et de vous adapter à l'évolution des exigences au fil du temps.

Avantages par rapport à une formation en un seul passage :

- Amélioration ciblée : Corriger les modèles de défaillance spécifiques découverts lors de l'évaluation
- Raffinement adaptatif : répondre aux changements de distribution ou à l'évolution des exigences des produits
- Atténuation des risques : validez les améliorations progressivement plutôt que de vous engager dans une seule formation de longue durée
- Efficacité des données : concentrer les efforts de collecte de données sur les domaines dans lesquels le modèle est sous-performant
- Curriculum training : plusieurs cycles de formation avec des données de meilleure qualité

Comment ça marche

Emplacement et accès aux points de contrôle

Une fois chaque tâche de formation terminée, un fichier manifeste est généré à l'emplacement de sortie spécifié par le `output_path` paramètre dans votre configuration d'entraînement.

Pour accéder à votre point de contrôle

- Naviguez jusqu'à ce que vous avez spécifié `output_path` dans S3
- Téléchargez et extrayez le `output.tar.gz` fichier
- Ouvrez le `manifest.json` fichier à l'intérieur
- Localisez le `checkpoint_s3_bucket` paramètre, qui contient l'URI S3 de votre modèle entraîné

Exemple de structure manifest.json

```
{
  "checkpoint_s3_bucket": "s3://customer-escrow-<account-number>-smtj-<unique-
  identifier>/<job-name>/stepID",
  ...
}
```

Comprendre les compartiments d'entiercement

Les poids Amazon Nova étant exclusifs, les points de contrôle des modèles entraînés sont stockés dans des compartiments S3 séquestres au sein de comptes AWS gérés au lieu d'être copiés sur votre compte. Ces compartiments d'entiercement :

- Conservez en toute sécurité les poids de votre modèle personnalisé
- Peut être référencé par d'autres AWS services (inférence, évaluation et tâches de formation ultérieures)
- Ne sont accessibles qu'à votre AWS compte via les autorisations IAM
- Encourez des frais de stockage S3 standard sur votre compte (voir Considérations financières)

Vous pouvez utiliser le chemin du compartiment d'entiercement `model_name_or_path` lors de votre prochain cycle d'entraînement pour poursuivre l'entraînement itératif.

Utilisation de points de contrôle pour la formation itérative

Configurez votre prochaine tâche de formation pour utiliser le point de contrôle précédent comme modèle de base :

```
run:
  name: "my-iterative-training-job"
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: "s3://customer-escrow-<account-number>-smtj-<unique-identifiant>/<previous-job-name>"
  data_s3_path: s3://<bucket>/<data-file>.jsonl
  replicas: 4
```

Quand utiliser la formation itérative

Cas d'utilisation idéaux

Utilisez l'entraînement itératif lorsque vous avez :

- Boucles de rétroaction — Capacité à collecter des cas de défaillance réels et à les traiter systématiquement
- Environnements dynamiques : documentation évolutive ou rubriques APIs de support nécessitant des mises à jour périodiques du modèle
- Évaluation robuste — Des repères et des cadres d'évaluation solides (voir les exemples ci-dessous) pour mesurer les améliorations en toute confiance
- Fonctionnalité des opérations ML — Ressources pour gérer plusieurs cycles de formation et le contrôle des versions

Exemples de cadres d'évaluation robustes

- Suites de référence automatisées avec pass/fail seuils
- Protocoles d'évaluation humaine avec mesures de fiabilité inter-évaluateurs
- Scénarios de test en équipe rouge couvrant les cas critiques et les contributions contradictoires
- Infrastructure de tests A/B pour mesurer l'impact sur la production

Schémas courants

Pipeline SFT → RFT : Un modèle itératif fréquemment utilisé implique :

- SFT first — Apprenez au modèle à résoudre des problèmes grâce à des exemples de démonstration
- Deuxième RFT — Optimisez les performances dans l'ensemble des problèmes à l'aide de signaux de récompense

Cette séquence est essentielle lorsque les modèles fonctionnent mal au départ : la RFT sur des modèles de précision proche de zéro n'améliorera pas les performances sans établir au préalable des capacités de résolution de problèmes de base via SFT.

Quand ne pas utiliser la formation itérative

Évitez les formations itératives pour :

- Tâches stables et bien définies — Des données fixes répondant à des exigences constantes atteignent déjà des performances proches du plafond
- Problèmes de classification simples — Tâches restreintes pour lesquelles une formation en un seul passage suffit
- Contraintes en matière de ressources — Absence de capacités dédiées aux opérations de machine learning pour gérer plusieurs cycles de formation
- Gains marginaux — Lorsque les frais généraux ne justifient pas une amélioration minimale des performances

Exemple de flux de travail : SFT → RFT

Cet exemple illustre un modèle d'entraînement itératif courant pour les modèles de raisonnement.

Étape 1 : Formation initiale au SFT

Configurez et lancez votre tâche de formation SFT avec votre ensemble de données :

```
run:
  name: "initial-sft-training"
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: "nova-lite-2/prod"
  data_s3_path: s3://<bucket>/sft-training-data.jsonl
  validation_data_s3_path: s3://<bucket>/sft-validation-data.jsonl
```

Justification : SFT propose des démonstrations supplémentaires qui façonnent les sorties du modèle dans le format et la voix souhaités, établissant ainsi les capacités de base.

Une fois la formation terminée

- Notez ce qui `output_path` est configuré dans votre tâche de formation
- Téléchargez `output.tar.gz` depuis cet emplacement
- Extraire et localiser `manifest.json`
- Copiez la `checkpoint_s3_bucket` valeur

Étape 2 : formation RFT sur le point de contrôle SFT

Créez une nouvelle tâche de formation RFT à l'aide du point de contrôle SFT :

```
run:
  name: "rft-on-sft-checkpoint"
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: "s3://customer-escrow-<account-number>-smtj-<unique-identifiant>/<initial-sft-training>"
  data_s3_path: s3://<bucket>/rft-training-data.jsonl
  reward_lambda_arn: <your-reward-function-arn>
```

Justification : La formation RFT s'appuie sur les bases du SFT, permettant au modèle de développer des modèles de raisonnement plus complexes optimisés par votre fonction de récompense.

Étape 3 : Evaluer et itérer

Exécutez une évaluation sur le point de contrôle RFT pour évaluer les performances :

```
run:
  name: "evaluate-rft-checkpoint"
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: "s3://customer-escrow-<account-number>-smtj-<unique-identifier>/
<rft-on-sft-checkpoint>"
  data_s3_path: s3://<bucket>/evaluation-data.jsonl
```

Si les métriques cibles ne sont pas satisfaites, poursuivez l'itération avec des données ajustées ou des hyperparamètres.

Important

La technique d'entraînement (LoRa contre Full Rank) doit rester cohérente à toutes les itérations :

- Si vous utilisez SFT avec LoRa, vous devez utiliser RFT avec LoRa
- Si vous utilisez SFT avec rang complet, vous devez utiliser RFT avec rang complet
- Vous ne pouvez pas basculer entre LoRa et Full Rank en milieu de pipeline

Important

Si une clé KMS est utilisée pour le chiffrement dans le compartiment S3 de sortie appartenant à Amazon, cette même clé KMS doit être utilisée pour toutes les itérations futures.

Suivi des progrès au fil des itérations

Vous pouvez suivre les indicateurs MLflow en [MLflowconfigurant](#) vos tâches.

Création d'une MLflow application

Utilisation de l'interface utilisateur de Studio : si vous créez une tâche de formation via l'interface utilisateur de Studio, une MLflow application par défaut est créée automatiquement et sélectionnée par défaut dans les options avancées.

Utilisation de la CLI : si vous utilisez la CLI, vous devez créer une MLflow application et la transmettre comme entrée à la demande d'API de la tâche de formation.

```
mlflow_app_name="<enter your MLflow app name>"
role_arn="<enter your role ARN>"
bucket_name="<enter your bucket name>"
region="<enter your region>"

mlflow_app_arn=$(aws sagemaker create-mlflow-app \
  --name $mlflow_app_name \
  --artifact-store-uri "s3://$bucket_name" \
  --role-arn $role_arn \
  --region $region)
```

Accédez à l' MLflow application

À l'aide de la CLI : créez une URL présignée pour accéder à l'interface utilisateur de l' MLflow application :

```
aws sagemaker create-presigned-mlflow-app-url \
  --arn $mlflow_app_arn \
  --region $region \
  --output text
```

Utilisation de l'interface utilisateur de Studio : l'interface utilisateur de Studio affiche les indicateurs clés enregistrés MLflow et fournit un lien vers l'interface utilisateur de MLflow l'application.

Indicateurs clés à suivre

Surveillez ces indicateurs au fil des itérations pour évaluer les améliorations et suivre l'avancement des travaux :

Pour SFT

- Courbes de pertes d'entraînement
- Nombre d'échantillons consommés et temps de traitement des échantillons
- Précision des performances sur des ensembles de test conservés
- Conformité au format (par exemple, débit de sortie JSON valide)
- Perplexité quant aux données d'évaluation spécifiques à un domaine

Pour RFT

- Scores de récompense moyens au cours de l'entraînement

- Répartition des récompenses (pourcentage de réponses offrant des récompenses élevées)
- Tendances relatives aux récompenses de validation (attention au surajustement)
- Taux de réussite spécifiques aux tâches (par exemple, taux de réussite de l'exécution du code, précision des problèmes mathématiques)

Général

- Comparez les deltas de performance entre les itérations
- Scores d'évaluation humaine sur des échantillons représentatifs
- Métriques de production (en cas de déploiement itératif)

Déterminer quand s'arrêter

Arrêtez d'itérer lorsque :

- Plateaux de performance — L'entraînement supplémentaire n'améliore plus de manière significative les indicateurs cibles
- Le changement de technique est utile — Si une technique plafonne, essayez de la changer (par exemple, SFT → RFT → SFT) pour dépasser les plafonds de performance
- Indicateurs cibles atteints — Vos critères de réussite sont atteints
- Régression détectée — Les nouvelles itérations dégradent les performances (voir les procédures de restauration ci-dessous)

Pour les procédures d'évaluation détaillées, reportez-vous à la section Évaluation.

Bonnes pratiques

Commencez petit et agrandissez progressivement

Commencez avec un minimum d'ensembles de données et des périodes d'entraînement uniques pour valider votre approche avant de la mettre à l'échelle. Cela renforce la confiance et aide à identifier les problèmes à un stade précoce.

Établissez des indicateurs de réussite clairs

Définissez des indicateurs quantitatifs et qualitatifs avant de commencer :

Exemples de mesures de réussite par cas d'utilisation

- Réponse aux questions — Précision exacte des correspondances, score F1, évaluations des préférences humaines
- Génération de code — Taux de réussite des tests unitaires, réussite de la compilation, temps d'exécution
- Tâches de raisonnement — Précision des pas, exactitude de la réponse finale, scores de récompense
- Génération de contenu : scores de cohérence, précision factuelle, respect du style

Mettre en œuvre une évaluation automatisée

Configurez des pipelines d'évaluation automatisés pour suivre les performances après chaque cycle, ce qui permet une itération rapide et une comparaison objective.

Maintenez un contrôle rigoureux des versions

Document pour chaque itération :

- Versions et modifications des ensembles de données
- Modéliser les emplacements des points de contrôle
- Modifications des hyperparamètres
- Indicateurs de performance et deltas
- Observations qualitatives

Cela renforce les connaissances institutionnelles et permet le débogage.

Concentrez-vous sur la qualité des données plutôt que sur la quantité

Analysez les cas d'échec des cycles précédents et ajoutez des exemples ciblés et de haute qualité au lieu de simplement augmenter la taille du jeu de données.

Planifier le budget d'itération

Prévoyez 3 à 5 itérations comme intervalle typique :

- 1 à 2 itérations : souvent suffisantes pour de simples améliorations ou un polissage final

- 3 à 5 itérations : convient aux tâches complexes nécessitant plusieurs cycles de raffinement
- Plus de 5 itérations — Cela peut indiquer une baisse des rendements ou le besoin d'approches différentes

Ajustez en fonction du budget informatique et des taux d'amélioration des performances.

Mettre en œuvre des fonctionnalités de restauration

Si une itération introduit des régressions :

- Identifier la régression — Comparez les mesures d'évaluation entre les points de contrôle
- Retourner au point de contrôle précédent — Utilisez le chemin S3 du point de contrôle précédent comme votre `model_name_or_path`
- Ajuster l'approche d'entraînement : modifiez les données, les hyperparamètres ou la technique avant de réessayer
- Documenter la défaillance — Enregistrer la cause de la régression pour éviter de la répéter

Exemple de rétrogradation

```
run:
  name: "rollback-to-iteration-2"
  model_type: amazon.nova-2-lite-v1:0:256k
  # Use iteration 2 checkpoint instead of failed iteration 3
  model_name_or_path: "s3://customer-escrow-<account-number>-smtj-<unique-identifiser>/
<iteration-2-job-name>"
```

Considérations de coût

Stockage aux points de contrôle

- Emplacement — Les points de contrôle stockés dans des compartiments d'entiercement entraînent des frais de stockage S3 standard facturés sur votre compte AWS
- Conservation — Les points de contrôle sont conservés indéfiniment sauf s'ils sont explicitement supprimés
- Gestion — Mettez en œuvre des politiques de cycle de vie pour archiver ou supprimer les anciens points de contrôle dont vous n'avez plus besoin

Conseils d'optimisation des coûts

- Supprimer les points de contrôle intermédiaires après avoir validé les nouvelles itérations
- Archivez les points de contrôle dans S3 Glacier pour une conservation à long terme à moindre coût
- Définissez des politiques de rétention en fonction de vos besoins en matière de conformité et d'expérimentation

Limitations

Cohérence des familles de modèles

Lors d'un entraînement itératif, vous devez utiliser le même type de modèle pour toutes les itérations.

Formation initiale

```
run:
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: "nova-lite-2/prod"
```

Les itérations suivantes doivent utiliser le même `model_type`

```
run:
  model_type: amazon.nova-2-lite-v1:0:256k # Must match original
  model_name_or_path: "s3://customer-escrow-<account-number>-smtj-<unique-identifier>/<job-name>"
```

Cohérence des techniques d'entraînement

La technique d'entraînement doit rester cohérente d'une itération à l'autre :

- Les modèles entraînés par LoRa ne peuvent être entraînés que de manière itérative avec LoRa
- Full-Rank-trained les modèles ne peuvent être entraînés de manière itérative qu'avec Full-Rank

Comment fonctionnent les adaptateurs LoRa dans l'entraînement itératif

- Chaque itération d'entraînement LoRa produit de nouveaux poids adaptateurs
- Les nouveaux adaptateurs remplacent (et non empilés avec) les adaptateurs précédents

- Le modèle de base reste figé ; seuls les adaptateurs sont mis à jour

Matrice de compatibilité technique

Formation initiale	Peut itérer avec
SFT (classement complet)	SFT (rang complet), RFT (rang complet)
ST (LoRa)	ST (LoRa), RFT (LoRa)
RFT (classement complet)	RFT (classement complet)
RF (LoRa)	RF (LoRa)

Vérifier la compatibilité avant de commencer une tâche

- Vérifiez votre recette d'entraînement précédente pour identifier le type de modèle et la technique d'entraînement (LoRa ou Full-Rank)
- Assurez-vous que votre nouvelle recette correspond à la fois au type de modèle et à la technique
- Consultez le fichier manifest.json pour confirmer que le chemin du point de contrôle est correct

Résolution des problèmes

Erreur : « Des techniques d'entraînement de modèles incompatibles ont été détectées »

Cause : La technique d'entraînement (LoRa contre Full-Rank) ne correspond pas à la technique du point de contrôle.

Résolution : Assurez-vous que votre recette utilise la même technique d'entraînement que le modèle original :

- Si le point de contrôle a été entraîné avec LoRa, utilisez LoRa dans votre nouvelle recette
- Si le point de contrôle a été entraîné avec Full-Rank, utilisez Full-Rank dans votre nouvelle recette

Erreur : « Le modèle de base de la tâche extraite de `model_name_or_path` ne correspond pas à `model_type` »

Cause : Le type de modèle spécifié dans `model_type` ne correspond pas au modèle réel du point de contrôle.

Résolution : Vérifiez que :

- Le modèle `model_type` figurant dans votre recette correspond au type de modèle d'origine
- Le chemin d'entrée du point de contrôle S3 `model_name_or_path` est correct
- Vous utilisez le chemin du fichier `manifest.json` correct

Exemple de configuration correcte

```
run:
  model_type: amazon.nova-2-lite-v1:0:256k # Must match checkpoint's model
  model_name_or_path: "s3://customer-escrow-<account-number>-smtj-<unique-identifier>/<job-name>"
```

Erreur : « Configuration du modèle introuvable »

Cause : Le chemin d'entrée S3 `model_name_or_path` est pas valide ou est inaccessible.

Résolution :

- Vérifiez que le chemin S3 est correctement copié depuis le fichier `manifest.json`
- Assurez-vous que votre rôle IAM est autorisé à accéder au compartiment d'entiercement
- Confirmez que le travail de formation précédent a été terminé avec succès
- Vérifiez les fautes de frappe dans le chemin

Régression des performances après itération

Symptômes : les indicateurs d'évaluation diminuent après une nouvelle itération d'entraînement.

Résolution :

- Annulation — Utilisez le point de contrôle précédent comme modèle de base

- Analyser — Vérifiez les journaux d'entraînement et la qualité des données en cas d'échec de l'itération
- Ajuster — Modifiez les hyperparamètres (réduisez le taux d'apprentissage), améliorez la qualité des données ou réduisez les périodes d'entraînement
- Réessayer — Lance une nouvelle itération avec des ajustements

SageMaker Inférence

Les modèles Amazon Nova personnalisés sont désormais disponibles par SageMaker inférence. Avec Amazon Nova activé SageMaker, vous pouvez commencer à obtenir des prédictions, ou des inférences, à partir de vos modèles Amazon Nova personnalisés et entraînés. SageMaker propose une large sélection d'options de déploiement d'infrastructures et de modèles de machine learning pour répondre à tous vos besoins en matière d'inférence de machine learning. Grâce à SageMaker l'inférence, vous pouvez adapter le déploiement de vos modèles, gérer les modèles plus efficacement en production et réduire la charge opérationnelle.

SageMaker vous propose diverses options d'inférence, telles que des points de terminaison en temps réel pour obtenir une inférence à faible latence et des points de terminaison asynchrones pour des lots de demandes. En tirant parti de l'option d'inférence adaptée à votre cas d'utilisation, vous pouvez garantir des déploiements et des inférences de modèles efficaces. Pour plus d'informations sur l' SageMaker inférence, voir [Déployer des modèles pour l'inférence](#).

Important

Seuls les modèles personnalisés de rang complet et les modèles fusionnés par LoRA sont pris en charge par inférence. SageMaker Pour les modèles LoRa et les modèles de base non fusionnés, utilisez Amazon Bedrock.

Caractéristiques

Les fonctionnalités suivantes sont disponibles pour les modèles Amazon Nova par SageMaker inférence :

Capacités du modèle

- Génération de texte

Déploiement et évolutivité

- Points de terminaison en temps réel avec sélection d'instance personnalisée
- Auto Scaling — Ajustez automatiquement la capacité en fonction des modèles de trafic afin d'optimiser les coûts et l'utilisation du GPU. Pour plus d'informations, consultez [Automatically Scale Amazon SageMaker Models](#).
- Support de l'API de streaming pour la génération de jetons en temps réel

Surveillance et optimisation

- CloudWatch Intégration avec Amazon pour la surveillance et les alertes
- Optimisation de la latence en fonction de la zone de disponibilité grâce à la configuration VPC

Outils de développement

- AWS CLI support — Pour plus d'informations, consultez le manuel de [référence des commandes AWS CLI pour SageMaker](#).
- Intégration à un ordinateur portable via le support du SDK

Modèles et instances pris en charge

Lorsque vous créez vos points de terminaison d' SageMaker inférence, vous pouvez définir deux variables d'environnement pour configurer votre déploiement : `CONTEXT_LENGTH` et `MAX_CONCURRENCY`

- `CONTEXT_LENGTH`— Longueur totale maximale du jeton (entrée+sortie) par demande
- `MAX_CONCURRENCY`— Nombre maximum de demandes simultanées traitées par le point de terminaison

Le tableau suivant répertorie les modèles, les types d'instances et les configurations Amazon Nova pris en charge. Les valeurs `MAX_CONCURRENCY` représentent la simultanéité maximale prise en charge pour chaque paramètre `CONTEXT_LENGTH` :

Modèle	Type d'instance	Configurations prises en charge
Amazon Nova Micro	ml.g5.12xlarge	LONGUEUR_CONTEXTE : 4000, SIMULTANÉITÉ MAXIMALE : 32 LONGUEUR_CONTEXTE : 8000, SIMULTANÉITÉ MAXIMALE : 16
	ml.g5.24xlarge	LONGUEUR_CONTEXTE : 8000, SIMULTANÉITÉ MAXIMALE : 32
	ml.g6.12xlarge	LONGUEUR_CONTEXTE : 4000, SIMULTANÉITÉ MAXIMALE : 32 LONGUEUR_CONTEXTE : 8000, SIMULTANÉITÉ MAXIMALE : 16
	ml.g6.24xlarge	LONGUEUR_CONTEXTE : 8000, SIMULTANÉITÉ MAXIMALE : 32
	ml.g6.48xlarge	LONGUEUR_CONTEXTE : 8000, SIMULTANÉITÉ MAXIMALE : 32
	ml.p5.48xlarge	LONGUEUR_CONTEXTE : 8000, SIMULTANÉITÉ MAXIMALE : 32 LONGUEUR DU CONTEXTE : 16 000, CONCURRENCE MAXIMALE : 2 LONGUEUR DU CONTEXTE : 24 000, CONCURRENCE MAXIMALE : 1
Amazon Nova Lite	ml.g6.48xlarge	LONGUEUR_CONTEXTE : 4000, SIMULTANÉITÉ MAXIMALE : 32 LONGUEUR_CONTEXTE : 8000, SIMULTANÉITÉ MAXIMALE : 16

Modèle	Type d'instance	Configurations prises en charge
	ml.p5.48xlarge	LONGUEUR_CONTEXTE : 8000, SIMULTANÉITÉ MAXIMALE : 32 LONGUEUR DU CONTEXTE : 16 000, CONCURRENCE MAXIMALE : 2 LONGUEUR DU CONTEXTE : 24 000, CONCURRENCE MAXIMALE : 1
Nova 2 Lite	ml.p5.48xlarge	LONGUEUR_CONTEXTE : 8000, SIMULTANÉITÉ MAXIMALE : 32 LONGUEUR DU CONTEXTE : 16 000, CONCURRENCE MAXIMALE : 2 LONGUEUR DU CONTEXTE : 24 000, CONCURRENCE MAXIMALE : 1

Note

Les valeurs MAX_CONCURRENCY affichées sont les limites supérieures de chaque paramètre CONTEXT_LENGTH. Vous pouvez utiliser des longueurs de contexte plus faibles avec la même simultanéité, mais le dépassement de ces valeurs entraînera l'échec de la création du point de SageMaker terminaison.

Par exemple, sur Amazon Nova Micro avec un ml.g5.12xlarge :

- CONTEXT_LENGTH=2000, MAX_CONCURRENCY=32 → Valide
- CONTEXT_LENGTH=8000, MAX_CONCURRENCY=32 → Rejeté (la limite de simultanéité est de 16 à une longueur de contexte de 8 000)
- CONTEXT_LENGTH=8000, MAX_CONCURRENCY=4 → Valide
- CONTEXT_LENGTH=8000, MAX_CONCURRENCY=16 → Valide

- `CONTEXT_LENGTH=10000` → Rejeté (la longueur maximale du contexte est de 8 000 sur cette instance)

AWS Régions prises en charge

Le tableau suivant répertorie les AWS régions dans lesquelles les modèles Amazon Nova sont disponibles par SageMaker inférence :

Nom de la région	Code région	Disponibilité
USA Est (Virginie du Nord)	us-east-1	Available
USA Ouest (Oregon)	us-west-2	Available

Images de conteneurs prises en charge

Le tableau suivant répertorie l'image du conteneur URIs pour les modèles Amazon Nova par SageMaker inférence par région. Deux balises d'image sont disponibles pour chaque région : une balise versionnée (`v1.0.0`) et une balise la plus récente (`SM-Inference-latest`). Pour les déploiements de production, nous recommandons d'utiliser la balise versionnée.

Région	Image du conteneur URIs
us-east-1	<code>708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-inference-repo:SM-Inference-latest</code>
us-west-2	<code>176779409107.dkr.ecr.us-west-2.amazonaws.com/nova-inference-repo:SM-Inference-latest</code>

Bonnes pratiques

Pour connaître les meilleures pratiques relatives au déploiement et à la gestion des modèles sur SageMaker, consultez la section [Meilleures pratiques pour SageMaker](#).

Support

En cas de problème ou d'assistance concernant les modèles Amazon Nova par SageMaker inférence, contactez le AWS Support via la console ou votre responsable de AWS compte.

Rubriques

- [Démarrage](#)
- [Référence d'API](#)
- [Évaluer les modèles hébergés sur SageMaker Inference](#)
- [Déploiement des modèles Amazon Nova Forge dans le cadre de la détection des abus SageMaker par Amazon Inference](#)

Démarrage

Ce guide explique comment déployer des modèles Amazon Nova personnalisés sur des points de terminaison SageMaker en temps réel, configurer les paramètres d'inférence et invoquer vos modèles à des fins de test.

Conditions préalables

Les conditions requises pour déployer des modèles Amazon Nova par SageMaker inférence sont les suivantes :

- Créer un Compte AWS - Si vous n'en avez pas déjà un, consultez la section [Création d'un AWS compte](#).
- Autorisations IAM requises : assurez-vous que les politiques gérées suivantes sont associées à votre utilisateur ou à votre rôle IAM :
 - AmazonSageMakerFullAccess
 - AmazonS3FullAccess
- SDKs/CLI Versions requises : les versions du SDK suivantes ont été testées et validées par inférence avec les modèles Amazon SageMaker Nova :
 - SageMaker SDK Python v3.0.0+ (`sagemaker>=3.0.0`) pour une approche d'API basée sur les ressources
 - Boto3 version 1.35.0+ (`boto3>=1.35.0`) pour les appels d'API directs. Les exemples présentés dans ce guide utilisent cette approche.

- Augmentation du quota de service : demandez une augmentation du quota de SageMaker service Amazon pour le type d'instance ML que vous prévoyez d'utiliser pour votre point de terminaison d' SageMaker inférence (par exemple, `ml.p5.48xlarge` for endpoint usage). Pour obtenir la liste des types d'instances, consultez [the section called “Modèles et instances pris en charge”](#). Pour demander une augmentation, consultez la section [Demande d'augmentation de quota](#). Pour plus d'informations sur les quotas d' SageMaker instance, consultez la section [SageMaker Points de terminaison et quotas](#).

Étape 1 : Configuration des AWS informations d'identification

Configurez vos AWS informations d'identification à l'aide de l'une des méthodes suivantes :

Option 1 : AWS CLI (recommandée)

```
aws configure
```

Entrez votre clé AWS d'accès, votre clé secrète et votre région par défaut lorsque vous y êtes invité.

Option 2 : fichier AWS d'informations d'identification

Créez ou modifiez `~/.aws/credentials` :

```
[default]
aws_access_key_id = YOUR_ACCESS_KEY
aws_secret_access_key = YOUR_SECRET_KEY
```

Option 3 : variables d'environnement

```
export AWS_ACCESS_KEY_ID=your_access_key
export AWS_SECRET_ACCESS_KEY=your_secret_key
```

Note

Pour plus d'informations sur les informations AWS d'identification, consultez [Configuration et paramètres du fichier d'informations d'identification](#).

Initialiser les clients AWS

Créez un script ou un bloc-notes Python avec le code suivant pour initialiser le AWS SDK et vérifier vos informations d'identification :

```
import boto3

# AWS Configuration - Update these for your environment
REGION = "us-east-1" # Supported regions: us-east-1, us-west-2
AWS_ACCOUNT_ID = "YOUR_ACCOUNT_ID" # Replace with your AWS account ID

# Initialize AWS clients using default credential chain
sagemaker = boto3.client('sagemaker', region_name=REGION)
sts = boto3.client('sts')

# Verify credentials
try:
    identity = sts.get_caller_identity()
    print(f"Successfully authenticated to AWS Account: {identity['Account']}")

    if identity['Account'] != AWS_ACCOUNT_ID:
        print(f"Warning: Connected to account {identity['Account']}, expected {AWS_ACCOUNT_ID}")
except Exception as e:
    print(f"Failed to authenticate: {e}")
    print("Please verify your credentials are configured correctly.")
```

Si l'authentification est réussie, vous devriez voir un message confirmant l'identifiant de votre AWS compte.

Étape 2 : Création d'un rôle SageMaker d'exécution

Un rôle SageMaker d'exécution est un rôle IAM qui accorde des SageMaker autorisations pour accéder aux AWS ressources en votre nom, telles que les compartiments Amazon S3 pour les artefacts du modèle et CloudWatch pour la journalisation.

Création du rôle d'exécution

 Note

La création de rôles IAM nécessite `iam:CreateRole` et `iam:AttachRolePolicy` autorisations. Assurez-vous que votre utilisateur ou rôle IAM dispose de ces autorisations avant de continuer.

Le code suivant crée un rôle IAM doté des autorisations nécessaires pour déployer des modèles personnalisés Amazon Nova :

```
import json

# Create SageMaker Execution Role
role_name = f"SageMakerInference-ExecutionRole-{AWS_ACCOUNT_ID}"

trust_policy = {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {"Service": "sagemaker.amazonaws.com"},
            "Action": "sts:AssumeRole"
        }
    ]
}

iam = boto3.client('iam', region_name=REGION)

# Create the role
role_response = iam.create_role(
    RoleName=role_name,
    AssumeRolePolicyDocument=json.dumps(trust_policy),
    Description='SageMaker execution role with S3 and SageMaker access'
)

# Attach required policies
iam.attach_role_policy(
    RoleName=role_name,
    PolicyArn='arn:aws:iam::aws:policy/AmazonSageMakerFullAccess'
)

iam.attach_role_policy(
```

```
    RoleName=role_name,  
    PolicyArn='arn:aws:iam::aws:policy/AmazonS3FullAccess'  
)  
  
SAGEMAKER_EXECUTION_ROLE_ARN = role_response['Role']['Arn']  
print(f"Created SageMaker execution role: {SAGEMAKER_EXECUTION_ROLE_ARN}")
```

Utilisation d'un rôle d'exécution existant (facultatif)

Si vous avez déjà un rôle SageMaker d'exécution, vous pouvez l'utiliser à la place :

```
# Replace with your existing role ARN  
SAGEMAKER_EXECUTION_ROLE_ARN = "arn:aws:iam::YOUR_ACCOUNT_ID:role/  
YOUR_EXISTING_ROLE_NAME"
```

Pour rechercher des SageMaker rôles existants dans votre compte, procédez comme suit :

```
iam = boto3.client('iam', region_name=REGION)  
response = iam.list_roles()  
sagemaker_roles = [role for role in response['Roles'] if 'SageMaker' in  
    role['RoleName']]  
for role in sagemaker_roles:  
    print(f"{role['RoleName']}: {role['Arn']}")
```

Important

Le rôle d'exécution doit avoir une relation de confiance `sagemaker.amazonaws.com` et être autorisé à accéder à Amazon S3 et à SageMaker ses ressources.

Pour plus d'informations sur les rôles SageMaker d'exécution, consultez la section [SageMaker Rôles](#).

Étape 3 : Configuration des paramètres du modèle

Configurez les paramètres de déploiement pour votre modèle Amazon Nova. Ces paramètres contrôlent le comportement du modèle, l'allocation des ressources et les caractéristiques d'inférence. Pour obtenir une liste des types d'instances pris en charge et des valeurs `CONTEXT_LENGTH` et `MAX_CONCURRENCY` prises en charge pour chacun, consultez [the section called “Modèles et instances pris en charge”](#)

Paramètres requis

- **IMAGE:** URI de l'image du conteneur Docker pour le conteneur d'inférence Amazon Nova. Cela sera fourni par AWS.
- **CONTEXT_LENGTH:** longueur du contexte du modèle.
- **MAX_CONCURRENCY:** nombre maximum de séquences par itération ; définit la limite du nombre de demandes utilisateur individuelles (invites) pouvant être traitées simultanément au sein d'un même lot sur le GPU. Plage : entier supérieur à 0.

Paramètres de génération facultatifs

- **DEFAULT_TEMPERATURE:** Contrôle le caractère aléatoire de la génération. Plage : 0,0 à 2,0 (0,0 = déterministe, plus élevé = plus aléatoire).
- **DEFAULT_TOP_P:** seuil d'échantillonnage du noyau pour la sélection des jetons. Plage : 1e-10 à 1,0.
- **DEFAULT_TOP_K:** Limite la sélection de jetons aux K jetons les plus probables. Plage : entier égal ou supérieur à -1 (-1 = aucune limite).
- **DEFAULT_MAX_NEW_TOKENS:** Nombre maximum de jetons à générer en réponse (c'est-à-dire nombre maximum de jetons de sortie). Plage : entier égal ou supérieur à 1.
- **DEFAULT_LOGPROBS:** nombre de probabilités de log à renvoyer par jeton. Plage : nombre entier de 1 à 20.

Configurez votre déploiement

```
# AWS Configuration
REGION = "us-east-1" # Must match region from Step 1

# ECR Account mapping by region
ECR_ACCOUNT_MAP = {
    "us-east-1": "708977205387",
    "us-west-2": "176779409107"
}

# Container Image
IMAGE = f"{ECR_ACCOUNT_MAP[REGION]}.dkr.ecr.{REGION}.amazonaws.com/nova-inference-
repo:SM-Inference-latest"
print(f"IMAGE = {IMAGE}")

# Model Parameters
```

```

CONTEXT_LENGTH = "16000"          # Maximum total context length
MAX_CONCURRENCY = "2"             # Maximum concurrent sequences

# Optional: Default generation parameters (uncomment to use)
DEFAULT_TEMPERATURE = "0.0"      # Deterministic output
DEFAULT_TOP_P = "1.0"            # Consider all tokens
# DEFAULT_TOP_K = "50"           # Uncomment to limit to top 50 tokens
# DEFAULT_MAX_NEW_TOKENS = "2048" # Uncomment to set max output tokens
# DEFAULT_LOGPROBS = "1"         # Uncomment to enable log probabilities

# Build environment variables for the container
environment = {
    'CONTEXT_LENGTH': CONTEXT_LENGTH,
    'MAX_CONCURRENCY': MAX_CONCURRENCY,
}

# Add optional parameters if defined
if 'DEFAULT_TEMPERATURE' in globals():
    environment['DEFAULT_TEMPERATURE'] = DEFAULT_TEMPERATURE
if 'DEFAULT_TOP_P' in globals():
    environment['DEFAULT_TOP_P'] = DEFAULT_TOP_P
if 'DEFAULT_TOP_K' in globals():
    environment['DEFAULT_TOP_K'] = DEFAULT_TOP_K
if 'DEFAULT_MAX_NEW_TOKENS' in globals():
    environment['DEFAULT_MAX_NEW_TOKENS'] = DEFAULT_MAX_NEW_TOKENS
if 'DEFAULT_LOGPROBS' in globals():
    environment['DEFAULT_LOGPROBS'] = DEFAULT_LOGPROBS

print("Environment configuration:")
for key, value in environment.items():
    print(f" {key}: {value}")

```

Configuration des paramètres spécifiques au déploiement

Configurez maintenant les paramètres spécifiques pour le déploiement de votre modèle Amazon Nova, notamment l'emplacement des artefacts du modèle et la sélection du type d'instance.

Définir l'identifiant de déploiement

```

# Deployment identifier - use a descriptive name for your use case
JOB_NAME = "my-nova-deployment"

```

Spécifier l'emplacement des artefacts du modèle

Indiquez l'URI Amazon S3 où sont stockés vos artefacts de modèle Amazon Nova entraînés. Il doit s'agir de l'emplacement de sortie de votre tâche d'entraînement ou de mise au point de votre modèle.

```
# S3 location of your trained Nova model artifacts
# Replace with your model's S3 URI - must end with /
MODEL_S3_LOCATION = "s3://your-bucket-name/path/to/model/artifacts/"
```

Sélectionnez la variante du modèle et le type d'instance

```
# Configure model variant and instance type
TESTCASE = {
    "model": "lite2",          # Options: micro, lite, lite2
    "instance": "ml.p5.48xlarge" # Refer to "Supported models and instances" section
}

# Generate resource names
INSTANCE_TYPE = TESTCASE["instance"]
MODEL_NAME = JOB_NAME + "-" + TESTCASE["model"] + "-" + INSTANCE_TYPE.replace(".", "-")
ENDPOINT_CONFIG_NAME = MODEL_NAME + "-Config"
ENDPOINT_NAME = MODEL_NAME + "-Endpoint"

print(f"Model Name: {MODEL_NAME}")
print(f"Endpoint Config: {ENDPOINT_CONFIG_NAME}")
print(f"Endpoint Name: {ENDPOINT_NAME}")
```

Conventions de dénomination

Le code génère automatiquement des noms cohérents pour les AWS ressources :

- Nom du modèle : {JOB_NAME}-{model}-{instance-type}
- Config du point de terminaison : {MODEL_NAME}-Config
- Nom du point de terminaison : {MODEL_NAME}-Endpoint

Étape 4 : Création de la configuration SageMaker du modèle et du point de terminaison

Au cours de cette étape, vous allez créer deux ressources essentielles : un objet de SageMaker modèle qui référence les artefacts de votre modèle Amazon Nova et une configuration de point de terminaison qui définit la manière dont le modèle sera déployé.

SageMaker Modèle : objet modèle qui regroupe l'image du conteneur d'inférence, l'emplacement des artefacts du modèle et la configuration de l'environnement. Il s'agit d'une ressource réutilisable qui peut être déployée sur plusieurs terminaux.

Configuration du point de terminaison : définit les paramètres d'infrastructure pour le déploiement, notamment le type d'instance, le nombre d'instances et les variantes de modèle. Cela vous permet de gérer les paramètres de déploiement séparément du modèle lui-même.

Création du SageMaker modèle

Le code suivant crée un SageMaker modèle qui fait référence aux artefacts de votre modèle Amazon Nova :

```
try:
    model_response = sagemaker.create_model(
        ModelName=MODEL_NAME,
        PrimaryContainer={
            'Image': IMAGE,
            'ModelDataSource': {
                'S3DataSource': {
                    'S3Uri': MODEL_S3_LOCATION,
                    'S3DataType': 'S3Prefix',
                    'CompressionType': 'None'
                }
            },
            'Environment': environment
        },
        ExecutionRoleArn=SAGEMAKER_EXECUTION_ROLE_ARN,
        EnableNetworkIsolation=True
    )
    print("Model created successfully!")
    print(f"Model ARN: {model_response['ModelArn']}")

except sagemaker.exceptions.ClientError as e:
    print(f"Error creating model: {e}")
```

Paramètres clés :

- **ModelName**: identifiant unique pour votre modèle
- **Image**: URI de l'image du conteneur Docker pour l'inférence Amazon Nova
- **ModelDataSource**: emplacement des artefacts de votre modèle sur Amazon S3

- **Environment**: variables d'environnement configurées à l'étape 3
- **ExecutionRoleArn**: rôle IAM à partir de l'étape 2
- **EnableNetworkIsolation**: défini sur True pour une sécurité renforcée (empêche le conteneur de passer des appels réseau sortants)

Création de la configuration du point de terminaison

Créez ensuite une configuration de point de terminaison qui définit votre infrastructure de déploiement :

```
# Create Endpoint Configuration
try:
    production_variant = {
        'VariantName': 'primary',
        'ModelName': MODEL_NAME,
        'InitialInstanceCount': 1,
        'InstanceType': INSTANCE_TYPE,
    }

    config_response = sagemaker.create_endpoint_config(
        EndpointConfigName=ENDPOINT_CONFIG_NAME,
        ProductionVariants=[production_variant]
    )
    print("Endpoint configuration created successfully!")
    print(f"Config ARN: {config_response['EndpointConfigArn']}")

except sagemaker.exceptions.ClientError as e:
    print(f"Error creating endpoint configuration: {e}")
```

Paramètres clés :

- **VariantName**: identifiant pour cette variante de modèle (utilisez « principal » pour les déploiements à modèle unique)
- **ModelName**: fait référence au modèle créé ci-dessus
- **InitialInstanceCount**: nombre d'instances à déployer (commencez par 1, puis augmentez ultérieurement si nécessaire)
- **InstanceType**: type d'instance ML sélectionné à l'étape 3

Vérifier la création de ressources

Vous pouvez vérifier que vos ressources ont été créées avec succès :

```
# Describe the model
model_info = sagemaker.describe_model(ModelName=MODEL_NAME)
print(f"Model Status: {model_info['ModelName']} created")

# Describe the endpoint configuration
config_info =
    sagemaker.describe_endpoint_config(EndpointConfigName=ENDPOINT_CONFIG_NAME)
print(f"Endpoint Config Status: {config_info['EndpointConfigName']} created")
```

Étape 5 : Déployer le point de terminaison

L'étape suivante consiste à déployer votre modèle Amazon Nova en créant un point de terminaison SageMaker en temps réel. Ce point de terminaison hébergera votre modèle et fournira un point de terminaison HTTPS sécurisé pour effectuer des demandes d'inférence.

La création d'un point de terminaison prend généralement 15 à 30 minutes pour AWS provisionner l'infrastructure, télécharger les artefacts de votre modèle et initialiser le conteneur d'inférence.

Créer le point de terminaison

```
import time

try:
    endpoint_response = sagemaker.create_endpoint(
        EndpointName=ENDPOINT_NAME,
        EndpointConfigName=ENDPOINT_CONFIG_NAME
    )
    print("Endpoint creation initiated successfully!")
    print(f"Endpoint ARN: {endpoint_response['EndpointArn']}")
except Exception as e:
    print(f"Error creating endpoint: {e}")
```

Surveillez la création de terminaux

Le code suivant interroge l'état du terminal jusqu'à ce que le déploiement soit terminé :

```
# Monitor endpoint creation progress
print("Waiting for endpoint creation to complete...")
print("This typically takes 15-30 minutes...\n")
```

```

while True:
    try:
        response = sagemaker.describe_endpoint(EndpointName=ENDPOINT_NAME)
        status = response['EndpointStatus']

        if status == 'Creating':
            print(f"# Status: {status} - Provisioning infrastructure and loading
model...")
        elif status == 'InService':
            print(f"# Status: {status}")
            print("\nEndpoint creation completed successfully!")
            print(f"Endpoint Name: {ENDPOINT_NAME}")
            print(f"Endpoint ARN: {response['EndpointArn']}")
            break
        elif status == 'Failed':
            print(f"# Status: {status}")
            print(f"Failure Reason: {response.get('FailureReason', 'Unknown')}")
            print("\nFull response:")
            print(response)
            break
        else:
            print(f"Status: {status}")

    except Exception as e:
        print(f"Error checking endpoint status: {e}")
        break

time.sleep(30) # Check every 30 seconds

```

Vérifiez que le terminal est prêt

Une fois que le point de terminaison est en place InService, vous pouvez vérifier sa configuration :

```

# Get detailed endpoint information
endpoint_info = sagemaker.describe_endpoint(EndpointName=ENDPOINT_NAME)

print("\n=== Endpoint Details ===")
print(f"Endpoint Name: {endpoint_info['EndpointName']}")
print(f"Endpoint ARN: {endpoint_info['EndpointArn']}")
print(f"Status: {endpoint_info['EndpointStatus']}")
print(f"Creation Time: {endpoint_info['CreationTime']}")
print(f>Last Modified: {endpoint_info['LastModifiedTime']}")

```

```
# Get endpoint config for instance type details
endpoint_config_name = endpoint_info['EndpointConfigName']
endpoint_config =
    sagemaker.describe_endpoint_config(EndpointConfigName=endpoint_config_name)

# Display production variant details
for variant in endpoint_info['ProductionVariants']:
    print(f"\nProduction Variant: {variant['VariantName']}")
    print(f"  Current Instance Count: {variant['CurrentInstanceCount']}")
    print(f"  Desired Instance Count: {variant['DesiredInstanceCount']}")
    # Get instance type from endpoint config
    for config_variant in endpoint_config['ProductionVariants']:
        if config_variant['VariantName'] == variant['VariantName']:
            print(f"  Instance Type: {config_variant['InstanceType']}")
            break
```

Résolution des problèmes liés à la création de terminaux

Raisons de défaillance courantes :

- Capacité insuffisante : le type d'instance demandé n'est pas disponible dans votre région
 - Solution : essayez un autre type d'instance ou demandez une augmentation de quota
- Autorisations IAM : le rôle d'exécution ne dispose pas des autorisations nécessaires
 - Solution : vérifier que le rôle a accès aux artefacts du modèle Amazon S3 et aux SageMaker autorisations nécessaires
- Artefacts du modèle introuvables : l'URI Amazon S3 est incorrecte ou inaccessible
 - Solution : vérifiez l'URI Amazon S3 et les autorisations du bucket, assurez-vous que vous vous trouvez dans la bonne région
- Limites de ressources : limites de compte dépassées pour les points de terminaison ou les instances
 - Solution : demander une augmentation du quota de service via Service Quotas ou AWS Support

Note

Si vous devez supprimer un point de terminaison défaillant et recommencer à zéro :

```
sagemaker.delete_endpoint(EndpointName=ENDPOINT_NAME)
```

Étape 6 : Invoquer le point de terminaison

Une fois que votre point de terminaison est en place InService, vous pouvez envoyer des demandes d'inférence pour générer des prédictions à partir de votre modèle Amazon Nova. SageMaker prend en charge les points de terminaison synchrones (en temps réel avec modes de streaming/non streaming) et les points de terminaison asynchrones (basés sur Amazon S3 pour le traitement par lots).

Configuration du client d'exécution

Créez un client SageMaker Runtime avec les paramètres de délai d'expiration appropriés :

```
import json
import boto3
import botocore
from botocore.exceptions import ClientError

# Configure client with appropriate timeouts
config = botocore.config.Config(
    read_timeout=120,      # Maximum time to wait for response
    connect_timeout=10,    # Maximum time to establish connection
    retries={'max_attempts': 3} # Number of retry attempts
)

# Create SageMaker Runtime client
runtime_client = boto3.client('sagemaker-runtime', config=config, region_name=REGION)
```

Création d'une fonction d'inférence universelle

La fonction suivante gère à la fois les demandes de diffusion en continu et les demandes non liées au streaming :

```
def invoke_nova_endpoint(request_body):
    """
    Invoke Nova endpoint with automatic streaming detection.

    Args:
        request_body (dict): Request payload containing prompt and parameters

    Returns:
        dict: Response from the model (for non-streaming requests)
        None: For streaming requests (prints output directly)
    """
```

```

body = json.dumps(request_body)
is_streaming = request_body.get("stream", False)

try:
    print(f"Invoking endpoint ({'streaming' if is_streaming else 'non-
streaming'})...")

    if is_streaming:
        response = runtime_client.invoke_endpoint_with_response_stream(
            EndpointName=ENDPOINT_NAME,
            ContentType='application/json',
            Body=body
        )

        event_stream = response['Body']
        for event in event_stream:
            if 'PayloadPart' in event:
                chunk = event['PayloadPart']
                if 'Bytes' in chunk:
                    data = chunk['Bytes'].decode()
                    print("Chunk:", data)
            else:
                # Non-streaming inference
                response = runtime_client.invoke_endpoint(
                    EndpointName=ENDPOINT_NAME,
                    ContentType='application/json',
                    Accept='application/json',
                    Body=body
                )

                response_body = response['Body'].read().decode('utf-8')
                result = json.loads(response_body)
                print("# Response received successfully")
                return result

except ClientError as e:
    error_code = e.response['Error']['Code']
    error_message = e.response['Error']['Message']
    print(f"# AWS Error: {error_code} - {error_message}")
except Exception as e:
    print(f"# Unexpected error: {str(e)}")

```

Exemple 1 : fin d'un chat sans diffusion en continu

Utilisez le format de chat pour les interactions conversationnelles :

```
# Non-streaming chat request
chat_request = {
    "messages": [
        {"role": "user", "content": "Hello! How are you?"}
    ],
    "max_tokens": 100,
    "max_completion_tokens": 100, # Alternative to max_tokens
    "stream": False,
    "temperature": 0.7,
    "top_p": 0.9,
    "top_k": 50,
    "logprobs": True,
    "top_logprobs": 3,
    "reasoning_effort": "low", # Options: "low", "high"
    "allowed_token_ids": None, # List of allowed token IDs
    "truncate_prompt_tokens": None, # Truncate prompt to this many tokens
    "stream_options": None
}

response = invoke_nova_endpoint(chat_request)
```

Exemple de réponse:

```
{
  "id": "chatcmpl-123456",
  "object": "chat.completion",
  "created": 1234567890,
  "model": "default",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Hello! I'm doing well, thank you for asking. I'm here and ready to help you with any questions or tasks you might have. How can I assist you today?"
      },
      "logprobs": {
        "content": [
          {
            "token": "Hello",
```



```

        "logprob": -0.123,
        "top_logprobs": [
            {"token": "Hello", "logprob": -0.123},
            {"token": "Hi", "logprob": -2.456},
            {"token": "Hey", "logprob": -3.789}
        ]
    }
    # Additional tokens...
]
},
"finish_reason": "stop"
}
],
"usage": {
    "prompt_tokens": 12,
    "completion_tokens": 28,
    "total_tokens": 40
}
}

```

Exemple 2 : complétion de texte simple

Utilisez le format de complétion pour une génération de texte simple :

```

# Simple completion request
completion_request = {
    "prompt": "The capital of France is",
    "max_tokens": 50,
    "stream": False,
    "temperature": 0.0,
    "top_p": 1.0,
    "top_k": -1, # -1 means no limit
    "logprobs": 3, # Number of log probabilities to return
    "allowed_token_ids": None, # List of allowed token IDs
    "truncate_prompt_tokens": None, # Truncate prompt to this many tokens
    "stream_options": None
}

response = invoke_nova_endpoint(completion_request)

```

Exemple de réponse:

```
{
```

```

    "id": "cml-789012",
    "object": "text_completion",
    "created": 1234567890,
    "model": "default",
    "choices": [
      {
        "text": " Paris.",
        "index": 0,
        "logprobs": {
          "tokens": [" Paris", "."],
          "token_logprobs": [-0.001, -0.002],
          "top_logprobs": [
            {" Paris": -0.001, " London": -5.234, " Rome": -6.789},
            {".": -0.002, ",": -4.567, " ": -7.890}
          ]
        },
        "finish_reason": "stop"
      }
    ],
    "usage": {
      "prompt_tokens": 6,
      "completion_tokens": 2,
      "total_tokens": 8
    }
  }
}

```

Exemple 3 : fin du chat en streaming

```

# Streaming chat request
streaming_request = {
  "messages": [
    {"role": "user", "content": "Tell me a short story about a robot"}
  ],
  "max_tokens": 200,
  "stream": True,
  "temperature": 0.7,
  "top_p": 0.95,
  "top_k": 40,
  "logprobs": True,
  "top_logprobs": 2,
  "reasoning_effort": "high", # For more detailed reasoning
  "stream_options": {"include_usage": True}
}

```

```
invoke_nova_endpoint(streaming_request)
```

Exemple de sortie de streaming :

```
Chunk: data: {"id":"chatcmpl-029ca032-fa01-4868-80b7-
c4cb1af90fb9","object":"chat.completion.chunk","created":1772060532,"model":"default","choices"
[{"index":0,"delta":
{"role":"assistant","content":"","reasoning_content":null},"logprobs":null,"finish_reason":null}
Chunk: data: {"id":"chatcmpl-029ca032-fa01-4868-80b7-
c4cb1af90fb9","object":"chat.completion.chunk","created":1772060532,"model":"default","choices"
[{"index":0,"delta":{"content":" Once","reasoning_content":null},"logprobs":
{"content":[{"token":"\u2581Once","logprob":-0.6078429222106934,"bytes":
[226,150,129,79,110,99,101],"top_logprobs":
[{"token":"\u2581Once","logprob":-0.6078429222106934,"bytes":
[226,150,129,79,110,99,101]}]},{"token":"\u2581In","logprob":-0.7864127159118652,"bytes":
[226,150,129,73,110]}]}}],"finish_reason":null,"token_ids":null}]]}
Chunk: data: {"id":"chatcmpl-029ca032-fa01-4868-80b7-
c4cb1af90fb9","object":"chat.completion.chunk","created":1772060532,"model":"default","choices"
[{"index":0,"delta":{"content":" upon","reasoning_content":null},"logprobs":
{"content":[{"token":"\u2581upon","logprob":-0.0012345,"bytes":
[226,150,129,117,112,111,110],"top_logprobs":
[{"token":"\u2581upon","logprob":-0.0012345,"bytes":[226,150,129,117,112,111,110]}],
{"token":"\u2581a","logprob":-6.789,"bytes":
[226,150,129,97]}]}}],"finish_reason":null,"token_ids":null}]]}
Chunk: data: {"id":"chatcmpl-029ca032-fa01-4868-80b7-
c4cb1af90fb9","object":"chat.completion.chunk","created":1772060532,"model":"default","choices"
[{"index":0,"delta":{"content":" a","reasoning_content":null},"logprobs":
{"content":[{"token":"\u2581a","logprob":-0.0001234,"bytes":
[226,150,129,97],"top_logprobs":[{"token":"\u2581a","logprob":-0.0001234,"bytes":
[226,150,129,97]}]},{"token":"\u2581time","logprob":-9.123,"bytes":
[226,150,129,116,105,109,101]}]}}],"finish_reason":null,"token_ids":null}]]}
Chunk: data: {"id":"chatcmpl-029ca032-fa01-4868-80b7-
c4cb1af90fb9","object":"chat.completion.chunk","created":1772060532,"model":"default","choices"
[{"index":0,"delta":{"content":" time","reasoning_content":null},"logprobs":
{"content":[{"token":"\u2581time","logprob":-0.0023456,"bytes":
[226,150,129,116,105,109,101],"top_logprobs":
[{"token":"\u2581time","logprob":-0.0023456,"bytes":[226,150,129,116,105,109,101]}],
{"token":"","logprob":-6.012,"bytes":
[44]}]}}],"finish_reason":null,"token_ids":null}]]}

# Additional chunks...
```

```

Chunk: data: {"id":"chatcmpl-029ca032-fa01-4868-80b7-
c4cb1af90fb9","object":"chat.completion.chunk","created":1772060532,"model":"default","choices"
[{"index":0,"delta":{"logprobs":null,"finish_reason":"stop"}],"usage":
{"prompt_tokens":15,"completion_tokens":87,"total_tokens":102}}
Chunk: data: [DONE]

```

Exemple 4 : achèvement d'une conversation multimodale

Utilisez le format multimodal pour les entrées d'image et de texte :

```

# Multimodal chat request (if supported by your model)
multimodal_request = {
    "messages": [
        {
            "role": "user",
            "content": [
                {"type": "text", "text": "What's in this image?"},
                {"type": "image_url", "image_url": {"url": "data:image/
jpeg;base64,..."}}
            ]
        }
    ],
    "max_tokens": 150,
    "temperature": 0.3,
    "top_p": 0.8,
    "stream": False
}

response = invoke_nova_endpoint(multimodal_request)

```

Exemple de réponse:

```

{
    "id": "chatcmpl-345678",
    "object": "chat.completion",
    "created": 1234567890,
    "model": "default",
    "choices": [
        {
            "index": 0,
            "message": {
                "role": "assistant",
                "content": "The image shows..."
            }
        }
    ]
}

```

```

        },
        "finish_reason": "stop"
    }
],
"usage": {
    "prompt_tokens": 1250,
    "completion_tokens": 45,
    "total_tokens": 1295
}
}

```

Étape 7 : Nettoyer les ressources (facultatif)

Pour éviter d'encourir des frais inutiles, supprimez les AWS ressources que vous avez créées au cours de ce didacticiel. SageMaker les terminaux sont soumis à des frais pendant leur fonctionnement, même si vous ne faites pas activement de demandes d'inférence.

Important

La suppression de ressources est permanente et ne peut pas être annulée. Assurez-vous de ne plus avoir besoin de ces ressources avant de continuer.

Supprimer le point de terminaison

```

import boto3

# Initialize SageMaker client
sagemaker = boto3.client('sagemaker', region_name=REGION)

try:
    print("Deleting endpoint...")
    sagemaker.delete_endpoint(EndpointName=ENDPOINT_NAME)
    print(f"# Endpoint '{ENDPOINT_NAME}' deletion initiated")
    print("Charges will stop once deletion completes (typically 2-5 minutes)")
except Exception as e:
    print(f"# Error deleting endpoint: {e}")

```

 Note

La suppression du point de terminaison est asynchrone. Vous pouvez contrôler l'état de suppression :

```
import time

print("Monitoring endpoint deletion...")
while True:
    try:
        response = sagemaker.describe_endpoint(EndpointName=ENDPOINT_NAME)
        status = response['EndpointStatus']
        print(f"Status: {status}")
        time.sleep(10)
    except sagemaker.exceptions.ClientError as e:
        if e.response['Error']['Code'] == 'ValidationException':
            print("# Endpoint successfully deleted")
            break
        else:
            print(f"Error: {e}")
            break
```

Supprimer la configuration du point de terminaison

Une fois le point de terminaison supprimé, supprimez la configuration du point de terminaison :

```
try:
    print("Deleting endpoint configuration...")
    sagemaker.delete_endpoint_config(EndpointConfigName=ENDPOINT_CONFIG_NAME)
    print(f"# Endpoint configuration '{ENDPOINT_CONFIG_NAME}' deleted")
except Exception as e:
    print(f"# Error deleting endpoint configuration: {e}")
```

Supprimer le modèle

Supprimez l'objet SageMaker modèle :

```
try:
    print("Deleting model...")
    sagemaker.delete_model(ModelName=MODEL_NAME)
```

```
print(f"# Model '{MODEL_NAME}' deleted")
except Exception as e:
    print(f"# Error deleting model: {e}")
```

Référence d'API

Les modèles Amazon Nova SageMaker utilisent l'API SageMaker Runtime standard à des fins d'inférence. Pour consulter la documentation complète de l'API, voir [Tester un modèle déployé](#).

Invocation du terminal

Les modèles Amazon Nova SageMaker prennent en charge deux méthodes d'invocation :

- Invocation synchrone : utilisez l'[InvokeEndpoint](#) API pour les demandes d'inférence en temps réel, sans diffusion en continu.
- Invocation en streaming : utilisez l'[InvokeEndpointWithResponseStream](#) API pour les demandes d'inférence en streaming en temps réel.

Format des demandes

Les modèles Amazon Nova prennent en charge deux formats de demande :

Format d'achèvement du chat

Utilisez ce format pour les interactions conversationnelles :

```
{
  "messages": [
    {"role": "user", "content": "string"}
  ],
  "max_tokens": integer,
  "max_completion_tokens": integer,
  "stream": boolean,
  "temperature": float,
  "top_p": float,
  "top_k": integer,
  "logprobs": boolean,
  "top_logprobs": integer,
  "reasoning_effort": "low" | "high",
  "allowed_token_ids": [integer],
```

```
"truncate_prompt_tokens": integer,
"stream_options": {
  "include_usage": boolean
}
}
```

Format de complétion du texte

Utilisez ce format pour générer du texte en toute simplicité :

```
{
  "prompt": "string",
  "max_tokens": integer,
  "stream": boolean,
  "temperature": float,
  "top_p": float,
  "top_k": integer,
  "logprobs": integer,
  "allowed_token_ids": [integer],
  "truncate_prompt_tokens": integer,
  "stream_options": {
    "include_usage": boolean
  }
}
```

Format de complétion de chat multimodal

Utilisez ce format pour les saisies d'images et de texte :

```
{
  "messages": [
    {
      "role": "user",
      "content": [
        {"type": "text", "text": "What's in this image?"},
        {"type": "image_url", "image_url": {"url": "data:image/jpeg;base64,..."}}
      ]
    }
  ],
  "max_tokens": integer,
  "temperature": float,
  "top_p": float,
  "stream": boolean
}
```



```
}
```

Paramètres de requête

- `messages(array)` : Pour le format de fin du chat. Tableau d'objets de message avec `role` et `content` champs. Le contenu peut être une chaîne pour le texte uniquement ou un tableau pour les entrées multimodales.
- `prompt(chaine)` : Pour le format de complétion du texte. Le texte d'entrée à partir duquel générer.
- `max_tokens(entier)` : nombre maximum de jetons à générer dans la réponse. Plage : 1 ou plus.
- `max_completion_tokens(entier)` : alternative à `max_tokens` pour terminer le chat. Nombre maximum de jetons de complétion à générer.
- `temperature(float)` : contrôle le caractère aléatoire de la génération. Plage : 0,0 à 2,0 (0,0 = déterministe, 2,0 = caractère aléatoire maximal).
- `top_p(float)` : seuil d'échantillonnage du noyau. Plage : 1e-10 à 1,0.
- `top_k(entier)` : limite la sélection des jetons aux K jetons les plus probables. Plage : -1 ou plus (-1 = aucune limite).
- `stream(booléen)` : s'il faut diffuser la réponse. Réglez sur `true` pour le streaming, `false` pour le non-streaming.
- `logprobs(booléen/entier)` : pour terminer le chat, utilisez le booléen. Pour les complétions de texte, utilisez un entier pour le nombre de probabilités logarithmiques à renvoyer. Gamme : 1 à 20.
- `top_logprobs(entier)` : nombre de jetons les plus susceptibles de renvoyer des probabilités de journalisation (discussions terminées uniquement).
- `reasoning_effort(string)` : niveau d'effort de raisonnement. Options : « faible », « élevé » (achèvement du chat pour les modèles personnalisés de Nova 2 Lite uniquement).
- `allowed_token_ids(array)` : liste des jetons IDs dont la génération est autorisée. Limite la sortie aux jetons spécifiés.
- `truncate_prompt_tokens(entier)` : tronquez l'invite à ce nombre de jetons s'il dépasse la limite.
- `stream_options(objet)` : options pour le streaming des réponses. Contient un `include_usage` booléen pour inclure l'utilisation des jetons dans les réponses de streaming.

Format de la réponse

Le format de réponse dépend de la méthode d'appel et du type de demande :

Réponse à la fin du chat (hors diffusion)

Pour les demandes de fin de chat synchrone :

```
{
  "id": "chatcmpl-123e4567-e89b-12d3-a456-426614174000",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "nova-micro-custom",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Hello! I'm doing well, thank you for asking. How can I help you
today?",
        "refusal": null,
        "reasoning": null,
        "reasoning_content": null
      },
      "logprobs": {
        "content": [
          {
            "token": "Hello",
            "logprob": -0.31725305,
            "bytes": [72, 101, 108, 108, 111],
            "top_logprobs": [
              {
                "token": "Hello",
                "logprob": -0.31725305,
                "bytes": [72, 101, 108, 108, 111]
              },
              {
                "token": "Hi",
                "logprob": -1.3190403,
                "bytes": [72, 105]
              }
            ]
          }
        ]
      },
      "finish_reason": "stop",
      "stop_reason": null,
      "token_ids": [9906, 0, 358, 2157, 1049, 11, 1309, 345, 369, 6464, 13]
    }
  ],
}
```

```

"usage": {
  "prompt_tokens": 9,
  "completion_tokens": 12,
  "total_tokens": 21,
  "prompt_tokens_details": {
    "cached_tokens": 0
  }
},
"prompt_token_ids": [9906, 0, 358]
}

```

Réponse de fin de texte (hors diffusion en continu)

Pour les demandes de saisie synchrone de texte :

```

{
  "id": "cmpl-123e4567-e89b-12d3-a456-426614174000",
  "object": "text_completion",
  "created": 1677652288,
  "model": "nova-micro-custom",
  "choices": [
    {
      "index": 0,
      "text": "Paris, the capital and most populous city of France.",
      "logprobs": {
        "tokens": ["Paris", ",", " the", " capital"],
        "token_logprobs": [-0.31725305, -0.07918124, -0.12345678, -0.23456789],
        "top_logprobs": [
          {
            "Paris": -0.31725305,
            "London": -1.3190403,
            "Rome": -2.1234567
          },
          {
            ",": -0.07918124,
            " is": -1.2345678
          }
        ]
      },
      "finish_reason": "stop",
      "stop_reason": null,
      "prompt_token_ids": [464, 6864, 315, 4881, 374],
      "token_ids": [3915, 11, 279, 6864, 323, 1455, 95551, 3363, 315, 4881, 13]
    }
  ]
}

```

```
],  
  "usage": {  
    "prompt_tokens": 5,  
    "completion_tokens": 11,  
    "total_tokens": 16,  
    "prompt_tokens_details": {  
      "cached_tokens": 0  
    }  
  }  
}
```

Réponse en streaming à la fin du chat

Pour les demandes de fin de chat en streaming, les réponses sont envoyées sous forme d'événements envoyés par le serveur (SSE) :

```
data: {  
  "id": "chatcmpl-123e4567-e89b-12d3-a456-426614174000",  
  "object": "chat.completion.chunk",  
  "created": 1677652288,  
  "model": "nova-micro-custom",  
  "choices": [  
    {  
      "index": 0,  
      "delta": {  
        "role": "assistant",  
        "content": "Hello",  
        "refusal": null,  
        "reasoning": null,  
        "reasoning_content": null  
      },  
      "logprobs": {  
        "content": [  
          {  
            "token": "Hello",  
            "logprob": -0.31725305,  
            "bytes": [72, 101, 108, 108, 111],  
            "top_logprobs": [  
              {  
                "token": "Hello",  
                "logprob": -0.31725305,  
                "bytes": [72, 101, 108, 108, 111]  
              }  
            ]  
          }  
        ]  
      }  
    }  
  ]  
}
```

```

        ]
      }
    ]
  },
  "finish_reason": null,
  "stop_reason": null
}
],
"usage": null,
"prompt_token_ids": null
}

data: {
  "id": "chatcmpl-123e4567-e89b-12d3-a456-426614174000",
  "object": "chat.completion.chunk",
  "created": 1677652288,
  "model": "nova-micro-custom",
  "choices": [
    {
      "index": 0,
      "delta": {
        "content": "! I'm"
      },
      "logprobs": null,
      "finish_reason": null,
      "stop_reason": null
    }
  ],
  "usage": null
}

data: {
  "id": "chatcmpl-123e4567-e89b-12d3-a456-426614174000",
  "object": "chat.completion.chunk",
  "created": 1677652288,
  "model": "nova-micro-custom",
  "choices": [
    {
      "index": 0,
      "delta": {},
      "finish_reason": "stop",
      "stop_reason": null
    }
  ],

```

```

"usage": {
  "prompt_tokens": 9,
  "completion_tokens": 12,
  "total_tokens": 21,
  "prompt_tokens_details": {
    "cached_tokens": 0
  }
}
}
data: [DONE]

```

Réponse en streaming à la fin du texte

Pour les demandes de complétion de texte en streaming :

```

data: {
  "id": "cmpl-123e4567-e89b-12d3-a456-426614174000",
  "object": "text_completion",
  "created": 1677652288,
  "model": "nova-micro-custom",
  "choices": [
    {
      "index": 0,
      "text": "Paris",
      "logprobs": {
        "tokens": ["Paris"],
        "token_logprobs": [-0.31725305],
        "top_logprobs": [
          {
            "Paris": -0.31725305,
            "London": -1.3190403
          }
        ]
      },
      "finish_reason": null,
      "stop_reason": null
    }
  ],
  "usage": null
}

data: {
  "id": "cmpl-123e4567-e89b-12d3-a456-426614174000",

```

```
"object": "text_completion",
"created": 1677652288,
"model": "nova-micro-custom",
"choices": [
  {
    "index": 0,
    "text": ", the capital",
    "logprobs": null,
    "finish_reason": null,
    "stop_reason": null
  }
],
"usage": null
}

data: {
  "id": "cmpl-123e4567-e89b-12d3-a456-426614174000",
  "object": "text_completion",
  "created": 1677652288,
  "model": "nova-micro-custom",
  "choices": [
    {
      "index": 0,
      "text": "",
      "finish_reason": "stop",
      "stop_reason": null
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 11,
    "total_tokens": 16
  }
}

data: [DONE]
```

Explication des champs de réponse

- **id**: identifiant unique pour la réalisation
- **object**: Type d'objet renvoyé (« chat.completion », « text_completion », « chat.completion.chunk »)

- `created`: horodatage Unix indiquant le moment où la complétion a été créée
- `model`: Modèle utilisé pour la réalisation
- `choices`: éventail de choix de complétion
- `usage`: informations sur l'utilisation des jetons, y compris l'invite, l'achèvement et le nombre total de jetons
- `logprobs`: enregistrez les informations de probabilité pour les jetons (sur demande)
- `finish_reason`: Raison pour laquelle le modèle a cessé de générer (« stop », « length », « content_filter »)
- `delta`: contenu incrémental dans les réponses au streaming
- `reasoning`: contenu de raisonnement lorsque `reasoning_effort` est utilisé
- `token_ids`: tableau de jetons IDs pour le texte généré

Pour une documentation complète sur les API, consultez les [InvokeEndpoint sections Référence d'InvokeEndpointWithResponseStream API et Référence d'API](#).

Évaluer les modèles hébergés sur SageMaker Inference

Ce guide explique comment évaluer vos modèles Amazon Nova personnalisés déployés sur des points de terminaison d' SageMaker inférence à l'aide d'[Inspect AI](#), un framework d'évaluation open source.

Note

Pour une présentation pratique, consultez le bloc-notes de [démarrage rapide SageMaker Inspect AI](#).

Présentation de

Vous pouvez évaluer vos modèles Amazon Nova personnalisés déployés sur des SageMaker terminaux à l'aide de benchmarks standardisés élaborés par la communauté des chercheurs en intelligence artificielle. Cette approche vous permet de :

- Évaluez des modèles Amazon Nova personnalisés (affinés, distillés ou autrement adaptés) à grande échelle
- Exécutez des évaluations par inférence parallèle sur plusieurs instances de point de terminaison

- Comparez les performances des modèles à l'aide de benchmarks tels que MMLU, TruthfulQA et HumanEval
- Intégrez à votre SageMaker infrastructure existante

Modèles pris en charge

Le fournisseur SageMaker d'inférence travaille avec :

- Modèles Amazon Nova (Nova Micro, Nova Lite, Nova Lite 2)
- Modèles déployés via des serveurs d'inférence compatibles vLLM ou OpenAI
- Tout point de terminaison compatible avec le format d'API OpenAI Chat Completions

Conditions préalables

Avant de commencer, assurez-vous de disposer des éléments suivants :

- Et Compte AWS avec les autorisations nécessaires pour créer et invoquer des points de SageMaker terminaison
- AWS informations d'identification AWS CLI configurées via des variables d'environnement ou un rôle IAM
- Python 3.9 ou supérieur

Autorisations IAM requises

Votre utilisateur ou rôle IAM a besoin des autorisations suivantes :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:InvokeEndpoint",
        "sagemaker:DescribeEndpoint"
      ],
      "Resource": "arn:aws:sagemaker:*:*:endpoint/*"
    }
  ]
}
```

```
}
```

Étape 1 : Déployer un SageMaker point de terminaison

Avant d'exécuter des évaluations, vous avez besoin d'un point de terminaison d' SageMaker inférence exécutant votre modèle.

Pour obtenir des instructions sur la création d'un point de terminaison d' SageMaker inférence avec les modèles Amazon Nova, consultez [the section called “Prise en main”](#).

Une fois que votre terminal est en InService état, notez le nom du point de terminaison à utiliser dans les commandes d'évaluation.

Étape 2 : Installation des dépendances d'évaluation

Créez un environnement virtuel Python et installez les packages requis.

```
# Create virtual environment
python3.12 -m venv venv
source venv/bin/activate

# Install uv for faster package installation
pip install uv

# Install Inspect AI and evaluation benchmarks
uv pip install inspect-ai inspect-evals

# Install AWS dependencies
uv pip install aioboto3 boto3 botocore openai
```

Étape 3 : Configuration des AWS informations d'identification

Choisissez l'une des méthodes d'authentification suivantes :

Option 1 : AWS CLI (recommandée)

```
aws configure
```

Entrez votre ID de clé AWS d'accès, votre clé d'accès secrète et votre région par défaut lorsque vous y êtes invité.

Option 2 : variables d'environnement

```
export AWS_ACCESS_KEY_ID=your_access_key
export AWS_SECRET_ACCESS_KEY=your_secret_key
export AWS_DEFAULT_REGION=us-west-2
```

Option 3 : rôle IAM

Si l'instance est exécutée sur Amazon EC2 ou sur SageMaker des ordinateurs portables, le rôle IAM de l'instance est automatiquement utilisé.

Vérifier les informations d'identification

```
import boto3

sts = boto3.client('sts')
identity = sts.get_caller_identity()
print(f"Account: {identity['Account']}")
print(f>User/Role: {identity['Arn']}")
```

Étape 4 : Installation du SageMaker fournisseur

Le SageMaker fournisseur permet à Inspect AI de communiquer avec vos SageMaker terminaux. Le processus d'installation du fournisseur est rationalisé dans le [bloc-notes de démarrage rapide](#).

Étape 5 : Téléchargez les benchmarks d'évaluation

Clonez le référentiel Inspect Evals pour accéder aux benchmarks standard :

```
git clone https://github.com/UKGovernmentBEIS/inspect_evals.git
```

Ce référentiel inclut des benchmarks tels que :

- MMLU et MMLU-Pro (connaissances et raisonnement)
- TruthfulQA (véracité)
- HumanEval (génération de code)
- GSM8K (raisonnement mathématique)

Étape 6 : Exécuter les évaluations

Exécutez une évaluation à l'aide de votre SageMaker point de terminaison :

```
cd inspect_evals/src/inspect_evals/

inspect eval mmlu_pro/mmlu_pro.py \
  --model sagemaker/my-nova-endpoint \
  -M region_name=us-west-2 \
  --max-connections 256 \
  --max-retries 100 \
  --display plain
```

Principaux paramètres

Paramètre	Par défaut	Description
--max-connections	10	Nombre de requêtes parallèles adressées au point de terminaison. Echelle en fonction du nombre d'instances (par exemple, 10 instances × 25 = 250).
--max-retries	3	Réessayez en cas d'échec des demandes. Utilisez 50-100 pour les évaluations de grande envergure.
-M region_name	us-east-1	AWS région dans laquelle votre terminal est déployé.
-M read_timeout	600	Délai d'expiration de la demande en secondes.
-M connect_timeout	60	Délai d'expiration de la connexion en secondes.

Recommandations de réglage

Pour un point de terminaison multi-instance :

```
# 10-instance endpoint example
--max-connections 250 # ~25 connections per instance
--max-retries 100     # Handle transient errors
```

Un réglage --max-connections trop élevé risque de submerger le terminal et de provoquer un ralentissement. Si elle est trop basse, la capacité est sous-utilisée.

Étape 7 : Afficher les résultats

Lancez le visualiseur Inspect AI pour analyser les résultats de l'évaluation :

```
inspect view
```

Le visualiseur affiche :

- Scores et indicateurs globaux
- Résultats par échantillon avec réponses du modèle
- Analyse des erreurs et modèles de défaillance

Gestion des points de terminaison

Mettre à jour un terminal

Pour mettre à jour un point de terminaison existant avec un nouveau modèle ou une nouvelle configuration :

```
import boto3

sagemaker = boto3.client('sagemaker', region_name=REGION)

# Create new model and endpoint configuration
# Then update the endpoint
sagemaker.update_endpoint(
    EndpointName=EXISTING_ENDPOINT_NAME,
    EndpointConfigName=NEW_ENDPOINT_CONFIG_NAME
)
```

Supprimer un point de terminaison

```
sagemaker.delete_endpoint(EndpointName=ENDPOINT_NAME)
```

Intégration de benchmarks personnalisés

Vous pouvez ajouter de nouveaux points de référence à Inspect AI à l'aide du flux de travail suivant :

1. Étudier le format de l'ensemble de données de référence et les mesures d'évaluation
2. Passez en revue les implémentations similaires dans `inspect_evals/`
3. Créez un fichier de tâches qui convertit les enregistrements de jeux de données en échantillons Inspect AI

4. Implémenter des solveurs et des évaluateurs appropriés

5. Validez avec un petit essai

Exemple de structure de tâches :

```
from inspect_ai import Task, task
from inspect_ai.dataset import hf_dataset
from inspect_ai.scorer import choice
from inspect_ai.solver import multiple_choice

@task
def my_benchmark():
    return Task(
        dataset=hf_dataset("dataset_name", split="test"),
        solver=multiple_choice(),
        scorer=choice()
    )
```

Résolution des problèmes

Problèmes courants

Limitation ou temporisation des terminaux

- Réduire `--max-connections`
- Augmenter `--max-retries`
- Vérifiez les CloudWatch indicateurs des terminaux pour détecter les problèmes de capacité

Erreurs d'authentification

- Vérifiez que les AWS informations d'identification sont correctement configurées
- Vérifiez que les autorisations IAM incluent `sagemaker:InvokeEndpoint`

Erreurs de modèle

- Vérifiez que le terminal est en InService état
- Vérifiez que le modèle supporte le format d'API OpenAI Chat Completions

Ressources connexes

- [Inspectez la documentation de l'IA](#)
- [Inspecter le référentiel Evals](#)
- [SageMaker Manuel du développeur](#)
- [Déployer des modèles à des fins d'inférence](#)
- [Configuration du AWS CLI](#)

Déploiement des modèles Amazon Nova Forge dans le cadre de la détection des abus SageMaker par Amazon Inference

AWS s'engage à utiliser l'IA de manière responsable. Pour éviter toute utilisation abusive potentielle, lorsque vous déployez des modèles Amazon Nova Forge dans Amazon SageMaker Inference, SageMaker Inference met en œuvre des mécanismes automatisés de détection des abus afin d'identifier les violations potentielles de la [politique d' AWS utilisation acceptable](#) (AUP) et des conditions de service de l'entreprise, y compris la politique d'intelligence artificielle [responsable](#).

Nos mécanismes de détection des abus sont entièrement automatisés. Personne ne vérifie ni ne consulte les entrées utilisateur ni les sorties des modèles.

La détection automatique des abus inclut ce qui suit :

- Catégoriser le contenu — Nous utilisons des classificateurs pour détecter les contenus préjudiciables (tels que les contenus incitant à la violence) dans les entrées des utilisateurs et dans les sorties des modèles. Un classificateur est un algorithme qui traite les entrées et les sorties des modèles, et attribue le type de préjudice et le niveau de confiance. Nous pouvons exécuter ces classificateurs en fonction de l'utilisation du modèle Amazon Nova Forge. Le processus de classification est automatisé et n'implique aucune vérification humaine des entrées utilisateur ni des sorties des modèles.
- Identifier les modèles — Nous utilisons les métriques du classificateur pour identifier les violations potentielles et les comportements récurrents. Nous pouvons compiler et partager des statistiques de classification anonymisées. Amazon SageMaker Inference ne stocke pas les entrées de l'utilisateur ni les sorties du modèle.
- Détection et blocage du matériel pédopornographique (CSAM) : vous êtes responsable du contenu que vous (et vos utilisateurs finaux) téléchargez sur Amazon SageMaker Inference et devez vous assurer que ce contenu ne contient pas d'images illégales. Pour aider à stopper la diffusion du

CSAM, lors du déploiement d'un modèle Amazon Nova Forge dans Amazon SageMaker Inference, SageMaker Inference peut utiliser des mécanismes automatisés de détection des abus (tels que la technologie de hachage ou les classificateurs) pour détecter les CSAM apparents. Si Amazon SageMaker Inference détecte un CSAM apparent dans vos images saisies, Amazon SageMaker Inference bloquera la demande et vous recevrez un message d'erreur automatique. Amazon SageMaker Inference peut également déposer un rapport auprès du National Center for Missing and Exploited Children (NCMEC) ou d'une autorité compétente. Nous prenons le CSAM très au sérieux et nous continuerons à mettre à jour nos mécanismes de détection, de blocage et de signalement. Les lois applicables peuvent vous obliger à prendre des mesures supplémentaires, et vous êtes responsable de ces actions.

Une fois que nos mécanismes automatisés de détection des abus ont identifié des violations potentielles, nous pouvons vous demander des informations sur votre utilisation d'Amazon SageMaker Inference et sur le respect de nos conditions d'utilisation. Si vous ne répondez pas, si vous ne voulez pas ou si vous êtes incapable de vous conformer à ces conditions ou politiques, vous AWS pouvez suspendre votre accès à Amazon SageMaker Inference. L'échec de la tâche d'inférence peut également vous être facturé si nos tests automatisés détectent que les réponses du modèle ne sont pas conformes à nos conditions générales.

Contactez AWS le Support si vous avez d'autres questions. Pour plus d'informations, consultez le site [Amazon SageMaker FAQs](#).

Inférence Amazon Bedrock

Une fois que vous avez entraîné et testé votre modèle Amazon Nova, vous pouvez le déployer sur Amazon Bedrock à des fins d'inférence à l'échelle de production. Le processus de déploiement consiste à créer un modèle Amazon Bedrock à l'aide de l' `CreateCustomModel` API, à y exporter les artefacts de votre modèle à partir d'un compartiment Amazon S3 géré, puis, une fois le modèle ACTIF, à configurer un point de terminaison avec une inférence à la demande ou à débit provisionné.

Après avoir créé un modèle personnalisé dans SageMaker, vous pouvez utiliser l' `CreateCustomModel` API pour le déployer sur Amazon Bedrock depuis le SageMaker dépôt fiduciaire afin d'exécuter une inférence. Vous pouvez ensuite l'utiliser `CreateCustomModelDeployment` pour créer un point de terminaison d'inférence OD ou configurer une inférence de débit provisionnée pour un modèle PEFT (Parameter Efficient Fine Tuned). Vous pouvez configurer l'inférence du débit provisionné pour un modèle personnalisé Full Rank.

Vous pouvez également utiliser le SDK Amazon Nova Forge pour déployer des modèles Amazon Nova personnalisés. Le SDK Amazon Nova Forge fournit une expérience rationalisée permettant d'extraire les informations pertinentes d'une tâche de formation ou d'un point de contrôle du modèle S3 et de les publier sur Amazon Bedrock. Pour plus d'informations, consultez le [SDK Amazon Nova Forge](#).

Pour connaître les étapes détaillées de configuration de l'inférence Amazon Bedrock pour un modèle personnalisé, consultez [Déploiement de modèles Amazon Nova personnalisés sur Amazon Bedrock](#).

La section suivante fournit plus de détails sur l'inférence à la demande sur des modèles personnalisés.

Déployer un modèle personnalisé pour l'inférence à la demande

Une fois que vous avez créé un modèle personnalisé à l'aide d'une tâche de personnalisation de modèle (peaufinage, distillation ou pré-entraînement continu), vous pouvez configurer l'inférence à la demande pour ce modèle.

Pour configurer l'inférence à la demande pour un modèle personnalisé, vous déployez le modèle avec un déploiement de modèle personnalisé. Après avoir déployé votre modèle personnalisé, vous utilisez le Amazon Resource Name (ARN) du déploiement comme paramètre `modelId` dans vos opérations API `InvokeModel` ou `Converse`. Vous pouvez utiliser le modèle déployé pour l'inférence à la demande avec les fonctionnalités Amazon Bedrock telles que les terrains de jeu, les agents et les bases de connaissances.

Rubriques

- [Modèles pris en charge](#)
- [Déployer un modèle personnalisé](#)
- [Utiliser un déploiement pour l'inférence à la demande](#)
- [Supprimer un déploiement de modèle personnalisé](#)

Modèles pris en charge

Vous pouvez configurer l'inférence à la demande pour les modèles suivants :

- Amazon Nova Canvas
- Amazon Nova Lite
- Amazon Nova Micro

- Amazon Nova Pro

Déployer un modèle personnalisé

Vous pouvez déployer un modèle personnalisé avec la console Amazon Bedrock AWS Command Line Interface, ou AWS SDKs. Pour plus d'informations sur l'utilisation du déploiement pour l'inférence, consultez [Utiliser un déploiement pour l'inférence à la demande](#).

Rubriques

- [Déployer un modèle personnalisé \(console\)](#)
- [Déployer un modèle personnalisé \(AWS Command Line Interface\)](#)
- [Déploiement d'un modèle personnalisé \(AWS SDKs\)](#)

Déployer un modèle personnalisé (console)

Vous déployez un modèle personnalisé à partir de la page Modèles personnalisés comme suit. Vous pouvez également déployer un modèle à partir de la page Modèle personnalisé à la demande avec les mêmes champs. Pour trouver cette page, dans Inférence et évaluation dans le panneau de navigation, sélectionnez Modèle personnalisé à la demande.

Pour déployer un modèle personnalisé

1. Connectez-vous à l' AWS Management Console aide d'un [rôle IAM avec les autorisations Amazon Bedrock](#) et ouvrez la console Amazon Bedrock à l'adresse. <https://console.aws.amazon.com/nova/>
2. Dans le panneau de navigation gauche, sélectionnez Modèles personnalisés sous Modèles de fondation.
3. Dans l'onglet Modèles, sélectionnez le bouton radio correspondant au modèle que vous voulez déployer.
4. Choisissez Configurer l'inférence et choisissez Déployer avec la fonctionnalité à la demande.
5. Sous Détails du déploiement, fournissez les informations suivantes :
 - Nom du déploiement (obligatoire) : saisissez un nom unique pour votre déploiement.
 - Description (facultatif) : entrez une description de votre déploiement.
 - Balises (facultatif) : ajoutez des balises pour la répartition des coûts et la gestion des ressources.

6. Choisissez Créer. Lorsque le statut affiche Completed, votre modèle personnalisé est prêt pour l'inférence à la demande. Pour plus d'informations sur l'utilisation du modèle personnalisé, consultez [Utiliser un déploiement pour l'inférence à la demande](#).

Déployer un modèle personnalisé (AWS Command Line Interface)

Pour déployer un modèle personnalisé à des fins d'inférence à la demande à l'aide de AWS Command Line Interface, utilisez la `create-custom-model-deployment` commande avec le Amazon Resource Name (ARN) de votre modèle personnalisé. Cette commande utilise l'opération [CreateCustomModelDeployment](#) API. Elle renvoie l'ARN du déploiement que vous pouvez utiliser comme `modelId` lorsque vous effectuez des demandes d'inférence. Pour plus d'informations sur l'utilisation du déploiement pour l'inférence, consultez [Utiliser un déploiement pour l'inférence à la demande](#).

```
aws bedrock create-custom-model-deployment \  
--model-deployment-name "Unique name" \  
--model-arn "Custom Model ARN" \  
--description "Deployment description" \  
--tags '[  
  {  
    "key": "Environment",  
    "value": "Production"  
  },  
  {  
    "key": "Team",  
    "value": "ML-Engineering"  
  },  
  {  
    "key": "Project",  
    "value": "CustomerSupport"  
  }  
' \  
--client-request-token "unique-deployment-token" \  
--region region
```

Déploiement d'un modèle personnalisé (AWS SDKs)

Pour déployer un modèle personnalisé à des fins d'inférence à la demande, utilisez l'opération d'[CreateCustomModelDeployment](#) API avec le Amazon Resource Name (ARN) de votre modèle personnalisé. La réponse renvoie l'ARN du déploiement que vous pouvez utiliser comme `modelId`

lors des demandes d'inférence. Pour plus d'informations sur l'utilisation du déploiement pour l'inférence, consultez [Utiliser un déploiement pour l'inférence à la demande](#).

Le code suivant montre comment utiliser le kit SDK pour Python (Boto3) afin de déployer un modèle personnalisé.

```
def create_custom_model_deployment(bedrock_client):
    """Create a custom model deployment
    Args:
        bedrock_client: A boto3 Bedrock client for making API calls

    Returns:
        str: The ARN of the created custom model deployment

    Raises:
        Exception: If there is an error creating the deployment
    """

    try:
        response = bedrock_client.create_custom_model_deployment(
            modelDeploymentName="Unique deployment name",
            modelArn="Custom Model ARN",
            description="Deployment description",
            tags=[
                {'key': 'Environment', 'value': 'Production'},
                {'key': 'Team', 'value': 'ML-Engineering'},
                {'key': 'Project', 'value': 'CustomerSupport'}
            ],
            clientRequestToken=f"deployment-{uuid.uuid4()}"
        )

        deployment_arn = response['customModelDeploymentArn']
        print(f"Deployment created: {deployment_arn}")
        return deployment_arn

    except Exception as e:
        print(f"Error creating deployment: {str(e)}")
        raise
```

Utiliser un déploiement pour l'inférence à la demande

Après avoir déployé votre modèle personnalisé pour l'inférence à la demande, vous pouvez l'utiliser pour générer des réponses en effectuant des demandes d'inférence. Pour les opérations

InvokeModel ou Converse, vous utilisez le nom de ressource Amazon Resource Name (ARN) du déploiement comme `modelId`.

Pour plus d'informations sur la création de demandes d'inférence, consultez les rubriques suivantes :

- [Soumettre des invites et générer des réponses à l'aide de l'inférence de modèle](#)
- [Conditions préalables à l'exécution de l'inférence de modèle](#)
- [Soumettre des invites et générer des réponses à l'aide de l'API](#)

Supprimer un déploiement de modèle personnalisé

Une fois que vous avez fini d'utiliser votre modèle pour l'inférence à la demande, vous pouvez supprimer le déploiement. Après avoir supprimé le déploiement, vous ne pouvez plus l'utiliser pour l'inférence à la demande, mais cela ne supprime pas le modèle personnalisé sous-jacent.

Vous pouvez supprimer un déploiement de modèle personnalisé à l'aide de la console Amazon Bedrock AWS Command Line Interface, ou AWS SDKs.

Important

La suppression d'un déploiement de modèle personnalisé est irréversible. Assurez-vous que vous n'avez plus besoin du déploiement avant de procéder à la suppression. Si vous devez réutiliser le modèle personnalisé pour l'inférence à la demande, vous devez créer un nouveau déploiement.

Rubriques

- [Suppression d'un déploiement de modèle personnalisé \(console\)](#)
- [Supprimer un déploiement de modèle personnalisé \(AWS Command Line Interface\)](#)
- [Suppression d'un déploiement de modèle personnalisé \(AWS SDKs\)](#)

Suppression d'un déploiement de modèle personnalisé (console)

Pour supprimer un déploiement de modèle personnalisé

1. Dans le panneau de navigation, sous Inférence et évaluation, sélectionnez Modèle personnalisé à la demande.

2. Sélectionnez le déploiement de modèle personnalisé que vous voulez supprimer.
3. Sélectionnez Delete (Supprimer).
4. Dans la boîte de dialogue de confirmation, saisissez le nom du déploiement pour confirmer la suppression.
5. Choisissez Supprimer pour confirmer.

Le statut du déploiement passera à `Deleting` pendant la suppression. Une fois la suppression terminée, le déploiement sera supprimé de la liste.

Supprimer un déploiement de modèle personnalisé (AWS Command Line Interface)

Pour supprimer un déploiement de modèle personnalisé à l'aide du AWS Command Line Interface, utilisez la `delete-custom-model-deployment` commande avec votre identifiant de déploiement.

```
aws bedrock delete-custom-model-deployment \
--custom-model-deployment-identifier "deployment-arn-or-name" \
--region region
```

Suppression d'un déploiement de modèle personnalisé (AWS SDKs)

Pour supprimer un déploiement de modèle personnalisé par programmation, utilisez l'opération d'[DeleteCustomModelDeployment](#) API avec le nom ou l'ARN (Amazon Resource Name) du déploiement. Le code suivant montre comment utiliser le kit SDK for Python (Boto3) pour supprimer un déploiement de modèle personnalisé.

```
def delete_custom_model_deployment(bedrock_client):
    """Delete a custom model deployment

    Args:
        bedrock_client: A boto3 Bedrock client for making API calls

    Returns:
        dict: The response from the delete operation

    Raises:
        Exception: If there is an error deleting the deployment
    """

    try:
```

```
response = bedrock_client.delete_custom_model_deployment(  
    customModelDeploymentIdentifier="Deployment identifier"  
)  
  
print(f"Deployment deletion initiated")  
return response  
  
except Exception as e:  
    print(f"Error deleting deployment: {str(e)}")  
    raise
```

Inférence à la demande sur des modèles personnalisés

L'inférence à la demande (OD) vous permet d'exécuter une inférence sur vos modèles Amazon Nova personnalisés sans avoir à gérer des points de terminaison de débit alloué. Cela vous aide à optimiser vos coûts et à vous mettre à l'échelle efficacement. Avec l'inférence à la demande, vous êtes facturé en fonction de votre utilisation, mesurée en jetons, tant en entrée qu'en sortie.

Exigences de compatibilité

Les exigences de compatibilité suivantes s'appliquent :

- L'inférence OD est prise en charge pour les modèles de compréhension personnalisés Amazon Nova Pro, Lite et Micro. L'inférence OD n'est pas prise en charge pour les modèles de génération de contenu personnalisés Nova.
- L'inférence OD est prise en charge pour les modèles de compréhension personnalisés Amazon Nova entraînés après le 16 juillet 2025. Les modèles personnalisés entraînés avant le 16 juillet 2025 ne sont pas compatibles avec l'inférence OD.
- Personnalisation Amazon Bedrock : l'inférence OD est prise en charge pour les modèles personnalisés avec Amazon Bedrock et pour les modèles étudiants qui ont été distillés à partir d'un modèle enseignant avec Amazon Bedrock.
- SageMaker Personnalisation par l'IA : pour les modèles personnalisés en SageMaker IA, l'inférence OD n'est prise en charge que pour les modèles PEFT (Parameter-Efficient Finetuning) lorsque le modèle est hébergé sur Amazon Bedrock. Cela inclut l'optimisation directe des préférences (Direct Preference Optimization) et le PEFT. L'inférence OD n'est pas prise en charge pour les modèles peaufinés à rang complet.

Entraînement et inférence des modèles

Lorsque vous entraînez un nouveau modèle Amazon Nova Pro, Lite ou Micro personnalisé sur Amazon Bedrock ou SageMaker AI à l'aide de PEFT après le 16 juillet 2025, le modèle sera automatiquement compatible avec les options d'inférence provisionnées et à la demande. Vous pouvez sélectionner votre méthode d'inférence préférée lorsque vous déployez votre modèle.

Pour utiliser l'inférence OD avec un modèle entraîné après le 16 juillet 2025, veuillez suivre les étapes suivantes :

1. Créez une nouvelle tâche de réglage avec l'API de [personnalisation Amazon Bedrock ou l'API de personnalisation SageMaker AI](#).
2. Déployez le modèle nouvellement formé sur Amazon Bedrock à l'aide de l'[CreateCustomModel API](#).
3. Déployez pour une inférence à la demande à l'aide de l' CustomModelDeployment API.

Limites de débit

Les limites suivantes de demandes par minute (RPM) et de jetons par minute (TPM) s'appliquent aux demandes d'inférence à la demande :

Modèle de base pour modèle personnalisé	RPM par déploiement de modèles personnalisés	Déploiement du TPM par modèle personnalisé
Nova 2 Lite	2 000	4 000 000

Pour en savoir plus sur les quotas disponibles pour Amazon Nova, consultez [Quotas pour Amazon Nova](#).

Latence

Vous pouvez vous attendre à une différence de end-to-end latence (c'est-à-dire, Time To First Token (TTFT)) de 20 à 55 % entre l'invocation du modèle de base et l'adaptateur. La valeur exacte de la latence varie en fonction de la taille du modèle et est conforme aux normes de l'industrie.

Limites de la personnalisation des modèles Amazon Nova

La personnalisation d'Amazon Nova ne prend pas en charge les fonctionnalités suivantes sur SageMaker.

- SSH dans l'instance pour trouver les métriques

En raison des contrôles de sécurité en place, vous ne pouvez pas accéder par SSH au nœud principal de l'instance d'entraînement algo-1 pour trouver les statistiques de mémoire ou les statistiques NVIDIA et valider les étapes d'entraînement.

- Support pour le SDK pour SageMaker formateurs

SageMaker le formateur n'est actuellement pas disponible pour commencer un poste de formation. Vous pouvez commencer des tâches d'entraînement uniquement via l'API d'estimateur à ce jour.

- Les piscines chaudes ne sont pas accessibles aux postes SageMaker de formation

En raison des contrôles de sécurité en place, les piscines SageMaker chaudes ne peuvent pas être utilisées pour maintenir l'instance dans la piscine chaude jusqu'au moment de sa mise en service.

- Fusion de modèles personnalisés

La fusion de plusieurs modèles n'est actuellement pas prise en charge. Cela signifie qu'il n'est pas possible de créer plusieurs adaptateurs LoRA et d'effectuer une opération de fusion multiple avec le modèle de base.

- Outil d'observabilité pris en charge

[TensorBoard](#) est le seul outil d'observabilité pris en charge pour consulter les indicateurs relatifs aux postes SageMaker de formation. [MLFlow](#) ou [WandB](#) ne sont pas pris en charge actuellement. Pour plus d'informations sur l'utilisation TensorBoard dans SageMaker, voir [TensorBoard dans SageMaker](#).

Amazon Nova Forge

Amazon Nova Forge est un first-of-its-kind service qui offre aux entreprises le moyen le plus simple et le plus rentable de créer leurs propres modèles de pointe à l'aide de Nova.

Amazon Nova Forge introduit le concept de modèles de « formation ouverte », qui permettent aux entreprises d'accéder à une variété de points de contrôle des premiers modèles et de combiner des données propriétaires avec des ensembles de données sélectionnés par Amazon à chaque étape de la formation des modèles. Cela permet aux modèles de maximiser l'apprentissage à partir de données propriétaires tout en minimisant le risque d'oublier des compétences fondamentales telles que le raisonnement.

Nova Forge fournit les fonctionnalités clés suivantes :

- Accédez aux points de contrôle à toutes les phases du développement des modèles et tirez parti des nouveaux modèles Nova avant qu'ils ne soient largement disponibles
- Combinez vos données propriétaires avec les données de formation sélectionnées par Amazon Nova
- Procédez à un apprentissage par renforcement avec des fonctions de récompense dans votre environnement
- Utilisez des recettes à bouton-poussoir optimisées pour être créées avec Nova via des flux de travail visuels ou une interface de ligne de commande
- Utilisez le Responsible AI Toolkit pour aligner les modèles sur les directives d'Amazon Nova en matière d'IA responsable pendant le processus de formation et mettez en œuvre des contrôles d'exécution pour modérer les réponses des modèles lors de l'inférence.

Conditions préalables

Rubriques

- [Abonnez-vous à Nova Forge](#)
- [Autres prérequis](#)

Abonnez-vous à Nova Forge

Pour demander l'accès au service Amazon Nova Forge, ajoutez la balise suivante à votre rôle IAM dans la console : `key forge-subscription with value true`. Après avoir ajouté ce tag à votre rôle, rendez-vous dans SageMaker AI Console > Entraînement et personnalisation des modèles, puis cliquez sur Nova Forge. Sur cette page, vous trouverez des informations sur le service, les tarifs et les fonctionnalités. Vous pouvez demander un abonnement puis gérer votre abonnement à partir de cette page.

1. Le rôle doit être autorisé à appeler l'`APIListAttachedRolePolicy`, et la réponse doit inclure soit une politique, `AdministratorAccess` soit une `AmazonSageMakerFullAccess` politique.
2. Le rôle de connexion doit être autorisé à appeler l'`APIListRoleTags`, et les balises de réponse doivent inclure `tag.key=forge-subscription`.

Autres prérequis

Assurez-vous également que les conditions préalables suivantes sont remplies :

1. [Prérequis généraux](#)
2. Étapes supplémentaires pour les utilisateurs : ajoutez un groupe d'instances restreint (RIG) à votre SageMaker HyperPod cluster (pour terminer, suivez les étapes [ici](#))

Accès et configuration de Nova Forge

Pour vous intégrer à Nova Forge, suivez ce processus en deux étapes :

- Étape 1 : Abonnez-vous à Nova Forge
- Étape 2 : Configuration de HyperPod l'infrastructure

Obtenir les documents de Nova Forge

Pour obtenir les documents de Nova Forge, suivez les étapes ci-dessous :

```
mkdir NovaForgeHyperpodCLI
cd NovaForgeHyperpodCLI
aws s3 cp s3://nova-forge-c7363-206080352451-us-east-1/v1/ ./ --recursive
```

```
pip install -e .
```

Étape 1 : Abonnez-vous à Nova Forge

Résumé rapide :

1. Vérifiez que vous disposez d'un accès administrateur au compte Amazon Web Services.
2. Accédez à la console SageMaker AI et demandez l'accès à Nova Forge.
3. Attendez que l'équipe Nova vous envoie un e-mail de confirmation une fois votre demande d'abonnement approuvée.
4. Marquez votre rôle d'exécution avec le `forge-subscription` tag. Cette balise est requise pour accéder aux fonctionnalités et aux points de contrôle de Nova Forge. Ajoutez la balise suivante à votre rôle d'exécution :
 - Clé : `forge-subscription`
 - Valeur : `true`

Guide détaillé

Pour s'abonner à Nova Forge et utiliser efficacement le service de personnalisation, un client Amazon Web Services doit disposer d'un accès administrateur à son compte Amazon Web Services ou demander à son administrateur de lui accorder un accès administrateur. Ce document décrit les étapes nécessaires pour :

- Accès administrateur sécurisé
- Définissez des politiques pour vous abonner à Nova Forge
- Accédez à des recettes de personnalisation
- Configuration de la personnalisation
- Surveiller le flux de travail
- Évaluer le point de contrôle du modèle personnalisé

Option A

Flux 1 : L'utilisateur du compte doit contacter l'administrateur du compte pour demander ce qui suit :

- Ajoutez le `forge-subscription` tag au compte via IAM (voir l'annexe A pour les étapes à suivre).

- Ajoutez les ListAttachedRolePolicies autorisations ListRoleTags et via IAM (voir l'annexe B pour les étapes).

Required setup for requesting subscription
To request Nova Forge subscription, your IAM role must have the "forge-subscription" tag attached and the "ListRoleTags" and "ListAttachedRolePolicies" policies added.

[Learn more ↗](#)

Nova Forge

Choose from multiple Nova model checkpoints to introduce your data at the optimal stage, mix Amazon-curated training data from each stage with your data, integrate your proprietary tools as reward functions, customize the responsible AI controls, and build your own custom frontier model.

[Request subscription](#) [Documentation ↗](#)

Subscription

Request subscription to access Nova Forge resources. Once approved, your payer account will be eligible for a 1-month free trial. **Infrastructure costs will apply during the free trial.** See Nova Forge pricing details below.

Status
☐ Not subscribed

Option B

Flux : l'utilisateur du compte doit contacter l'administrateur du compte pour demander l'accès administrateur au compte.

- Une fois que l'accès administrateur est accordé, suivez les étapes de Flow 2.

Flux 2. Compte Amazon Web Services avec accès administrateur

- Ajoutez le tag Forge-Subscription au compte via IAM. Voir les étapes à l'annexe A

Annexe A. Ajouter une politique de forgeage d'abonnement au compte Amazon Web Services

1. Accédez au tableau de bord IAM d'Amazon Web Services. Cliquez sur Rôles sur la gauche. Recherchez admin et cliquez sur le rôle admin

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests

Roles (181) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for s

Search: admin 1 match

Role name	Trusted
libsAdminAccess-DO-NOT-DELETE	Account:

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

X.509 Standard

Use your own existing PKI infrastructure or [Certificate Manager Private Certificate Autl](#) to authenticate identities.

- Sélectionnez < AssumedRoleToUse > (par exemple, libsAdminAccess). Cliquez sur l'onglet Tags.

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests

libsAdminAccess-DO-NOT-DELETE Info

Summary

Creation date
November 06, 2025, 09:29 (UTC-08:00)

Last activity
5 minutes ago

ARN
arn:aws:iam::6s-DO-NOT-DE

Maximum session
1 hour

Permissions | Trust relationships | **Tags (2)** | Last Acce

Tags (2) Info

Tags are key-value pairs that you can add to AWS resources to help identify,

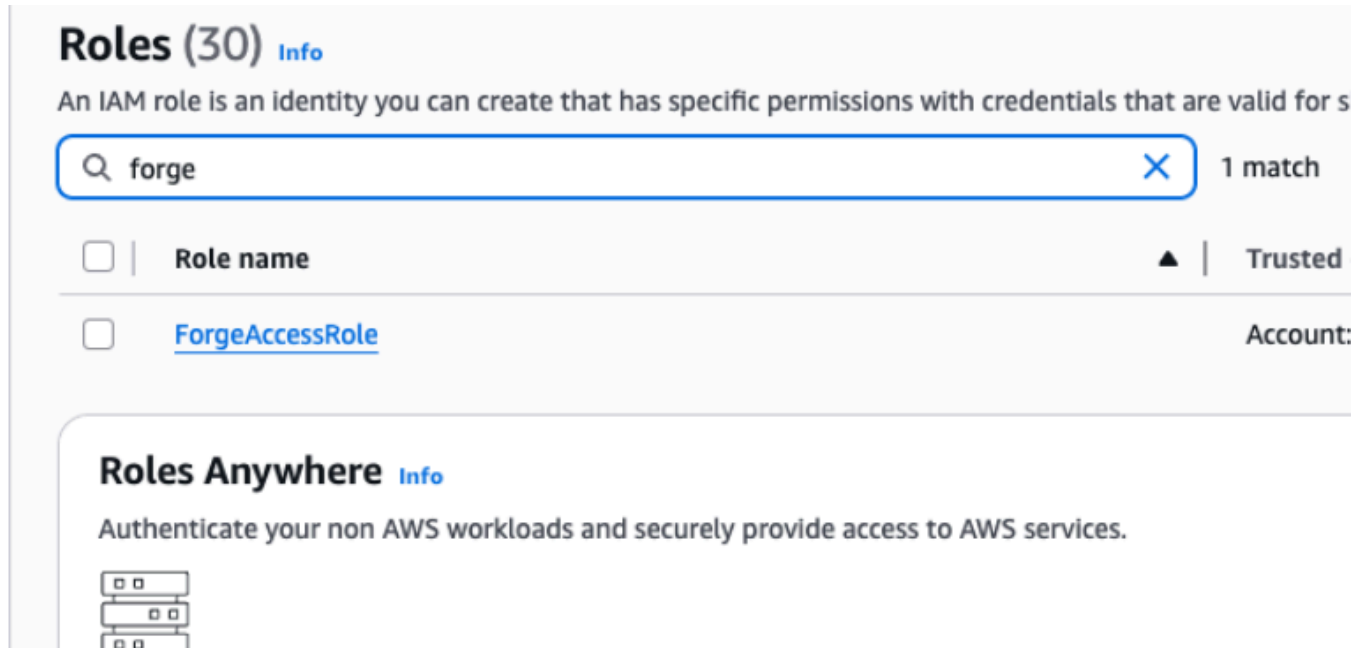
Key
forge-subscription

3. Cliquez sur Gérer les onglets. Ajoutez un nouveau tag. Tapez « forge-subscription » sous Key et cliquez sur Enregistrer les modifications

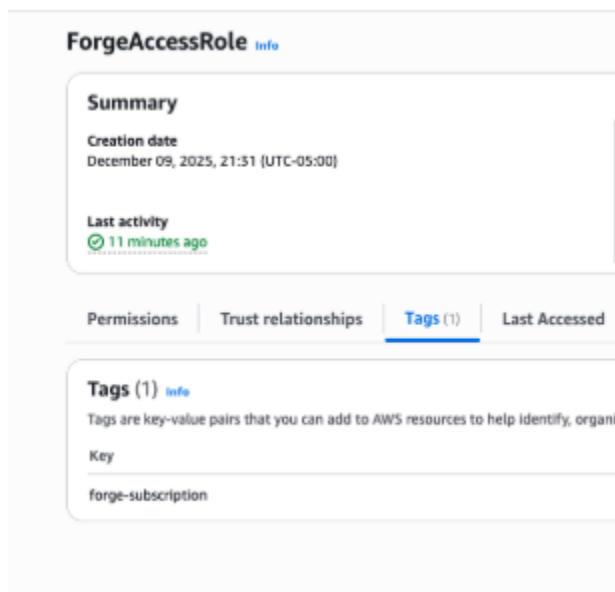
4. Assurez-vous de voir forge-subscription comme une clé dans la section Tags

Annexe B. Ajout ListRoleTags et ListAttachedPolicies politiques au compte Amazon Web Services pour le rôle d'administrateur non administrateur

1. Accédez au tableau de bord IAM d'Amazon Web Services. Cliquez sur Rôles sur la gauche. Recherchez < AssumedRoleToUse > (par exemple, ForgeAccessRole) et cliquez sur le rôle < AssumedRoleToUse > (par exemple, ForgeAccessRole)

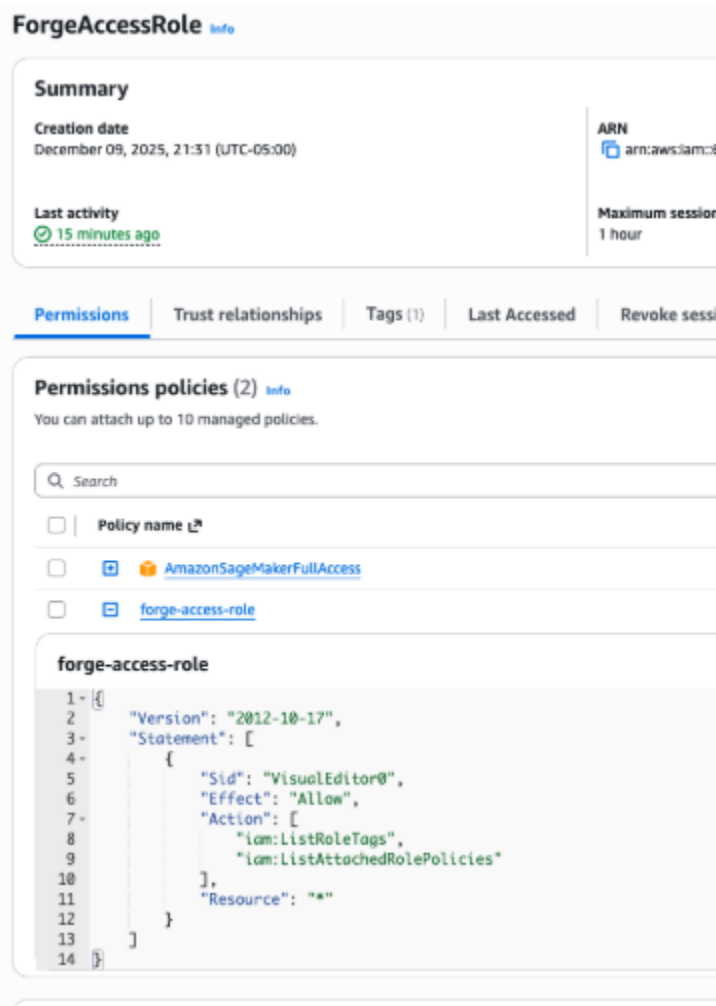


2. Cliquez sur le rôle < AssumedRoleToUse > (par exemple, ForgeAccessRole) et sélectionnez Tags. Ajoutez un nouveau tag avec le type « forge-subscription »



3. Sous Autorisations, ajoutez une nouvelle autorisation : Ajouter des autorisations → Créer une politique en ligne → Ajoutez la politique suivante répertoriée ci-dessous

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "iam:ListRoleTags",
        "iam:ListAttachedRolePolicies"
      ],
      "Resource": "*"
    }
  ]
}
```



Étape 2. Configuration de HyperPod l'infrastructure

Configurez l'infrastructure nécessaire en suivant les [instructions de l'atelier](#) pour configurer l'environnement avec les fonctionnalités compatibles avec Forge.

Paramètres de modération du contenu

Si vous avez besoin d'accéder à Nova Forge, des paramètres de modération de contenu (CCMS) personnalisables sont disponibles pour les modèles Amazon Nova Lite 1.0 et Pro 1.0. Le CCMS permet d'ajuster les contrôles de modération du contenu pour les aligner sur les exigences commerciales spécifiques tout en maintenant les garanties essentielles et responsables de l'IA. Pour déterminer si un modèle commercial convient au CCMS, contactez un responsable de AWS compte.

Pour plus d'informations sur la configuration et l'utilisation de CCMS avec des modèles personnalisés, consultez la [section Responsible AI Toolkit and Content Moderation](#).

Pré-entraînement continu et mi-entraînement

Note

Une documentation détaillée est fournie une fois que vous êtes abonné

Nova Forge CPT offre des fonctionnalités avancées allant au-delà du CPT standard, notamment l'accès à des points de contrôle intermédiaires et le mixage de données avec le corpus de pré-entraînement de Nova. Ces fonctionnalités permettent une adaptation plus efficace des domaines et une meilleure préservation des fonctionnalités générales du modèle.

Quels sont les points de contrôle intermédiaires et pourquoi sont-ils nécessaires ?

Les points de contrôle intermédiaires sont des instantanés du modèle Amazon Nova enregistrés à différentes étapes de la pré-formation, avant que le modèle n'atteigne son état final prêt pour la production. Au cours du développement du modèle, Amazon Nova est soumis à plusieurs phases de formation : pré-formation initiale avec un taux d'apprentissage constant, réduction du taux d'apprentissage, formation à l'extension du contexte et enfin formation à l'alignement et à la sécurité suivant les instructions. Pour le CPT, les points de contrôle intermédiaires sont souvent préférables au point de contrôle final Prod car ils sont plus plastiques et réceptifs à l'adaptation au domaine. Le point de contrôle Prod a fait l'objet d'une formation approfondie sur l'alignement et la sécurité, qui optimise le modèle pour une utilisation conversationnelle générale, mais peut le rendre résistant à l'apprentissage de nouveaux modèles spécifiques à un domaine pendant le CPT. En revanche, les points de contrôle contenant uniquement du texte partiellement ou entièrement préentraîné conservent les caractéristiques du modèle avant l'entraînement. Ils n'ont pas été fortement orientés vers des comportements spécifiques, ce qui en fait des points de départ plus efficaces pour l'adaptation au domaine. Lorsque vous effectuez un CPT à grande échelle (>10 milliards de jetons), le fait de partir de points de contrôle intermédiaires se traduit généralement par une convergence plus rapide, une meilleure stabilité de la formation et une acquisition plus efficace des connaissances du domaine. Toutefois, pour les CPT à petite échelle (<10 milliards de jetons), ou lorsque les capacités de suivi des instructions doivent être préservées, le point de contrôle Prod peut être plus approprié car il permet l'adaptation du domaine tout en préservant les capacités conversationnelles du modèle.

Plusieurs points de contrôle intermédiaires sont nécessaires pour le CPT car ils offrent différents niveaux de plasticité du modèle qui affectent l'efficacité avec laquelle le modèle peut absorber les nouvelles connaissances du domaine. Le point de contrôle final du Prod a fait l'objet d'une formation approfondie sur l'alignement et la sécurité, qui l'optimise pour une utilisation conversationnelle générale, mais le rend résistant à l'apprentissage de nouveaux modèles spécifiques au domaine. En d'autres termes, il a été renforcé après l'entraînement. En revanche, les points de contrôle antérieurs conservent les caractéristiques du modèle avant l'entraînement et n'ont pas été fortement orientés vers des comportements spécifiques, ce qui les rend plus souples et plus réceptifs à l'adaptation au domaine.

Pour optimiser l'efficacité de l'entraînement, plusieurs points de contrôle intermédiaires sont fournis.

Quels sont les points de contrôle disponibles ?

Nova 2.0

Il existe trois points de contrôle Amazon Nova Lite 2.0.

- **PRÉ-ENTRAÎNÉ** - [nova-lite-2/pretraining-text-RD] : Il s'agit du point de contrôle après le taux d'apprentissage constant et les étapes de réduction progressive de la pré-formation d'Amazon Nova, au cours desquelles le modèle est entraîné avec des milliards de jetons.
- **FORMATION INTERMÉDIAIRE** - [nova-lite-2/pretraining-text-CE] : Ce point de contrôle permet d'introduire des volumes intermédiaires de données non structurées avec un taux d'apprentissage plus modéré qu'avant la formation, absorbant ainsi des connaissances spécifiques à un domaine tout en évitant un oubli catastrophique.
- **APRÈS ENTRAÎNEMENT** - [nova-lite-2/prod] : Il s'agit du point de contrôle final entièrement aligné du modèle qui a suivi toutes les étapes pertinentes et postérieures à la formation.

Le tableau suivant décrit les différentes conditions avant et pendant l'entraînement.

Type de données	Exécuter	Avec Checkpoint
Données de domaine brutes non structurées à grande échelle (documents, journaux, articles, code, etc.)	Formation préalable continue	Pré-formé

Type de données	Exécuter	Avec Checkpoint
Données de domaine brutes non structurées à grande échelle (documents, journaux, articles, code, etc.)	Au milieu de l'entraînement	Pré-formé
Petits volumes de données brutes non structurées. Traces de raisonnement structurées/ données CoT	Au milieu de l'entraînement	Entraîné à mi-parcours
Démonstrations structurées (paires entrées-sorties de haute qualité, instructions de tâches sélectionnées, dialogues à plusieurs tours)	Réglage précis complet	Entraîné à mi-parcours
Démonstrations structurées (paires entrées-sorties de haute qualité, instructions de tâches sélectionnées, dialogues à plusieurs tours)	Réglage précis efficace des paramètres	Après avoir suivi une formation

Quel point de contrôle utiliser ?

Les points de contrôle partiellement pré-entraînés uniquement en texte et entièrement pré-entraînés uniquement en texte convergent généralement plus rapidement et nécessitent moins d'étapes de formation pour l'adaptation au domaine. Cependant, ils n'ont pas d'instructions ajustées et devraient suivre des étapes après la formation pour pouvoir effectuer des tâches utiles et suivre les instructions. Le point de contrôle GA peut nécessiter plus d'étapes pour s'adapter, mais constitue un point de départ plus sûr pour les expériences à petite échelle et conservera certaines de ses capacités après la formation, même après la formation du CPT.

En général, avec de grands ensembles de données d'entraînement (>10 milliards de jetons), commencez par des points de contrôle partiellement préentraînés uniquement en texte ou entièrement préentraînés uniquement en texte pour une formation plus efficace et plus stable, car la

base de connaissances du modèle sera considérablement modifiée. Avec de petits ensembles de données (<10 Go de jetons), utilisez le point de contrôle GA pour préserver les capacités de suivi des instructions tout en vous adaptant au domaine.

Comment utiliser le mixage de données pour Nova 2.0 ?

Lors de l'exécution du CPT avec de nouvelles données de domaine, il est très avantageux de mélanger les nouvelles données avec certaines des données utilisées précédemment lors de la phase de pré-apprentissage du modèle. Le mélange d'anciennes données avec de nouvelles données de domaine permet de résoudre deux problèmes :

- **Contrôle de l'oubli** : prévient l'oubli catastrophique en préservant les compétences et les connaissances existantes sur le modèle. Sans mélange de données, l'entraînement exclusivement sur des données de domaine restreint entraîne le remplacement des capacités générales du modèle. Par exemple, un modèle formé uniquement sur des documents juridiques risque de perdre sa capacité à coder ou à effectuer des calculs. Le mélange des ensembles de données du domaine général préserve ces compétences générales lors de l'acquisition du nouveau domaine.
- **Stabilité d'optimisation** : Maintient la stabilité de l'entraînement en ancrant les représentations internes du modèle. Au cours du CPT, les caractéristiques apprises du modèle sont modifiées et le mélange de données fournit des dégradés provenant de diverses sources qui guident cette adaptation en douceur. Sans cela, l'entraînement sur des distributions étroites peut provoquer une instabilité du gradient, les représentations du modèle étant modifiées de manière trop drastique, ce qui entraîne une divergence d'apprentissage, des pics de pertes ou un effondrement des capacités existantes. Il s'agit du compromis entre stabilité et plasticité : le modèle doit être suffisamment souple pour acquérir de nouvelles connaissances dans le domaine, mais suffisamment stable pour ne pas altérer ce qu'il sait déjà.

Capacités de mixage de données Nova CPT

L'accès aux données de pré-formation et aux points de contrôle d'Amazon Nova est l'une des principales offres de personnalisation d'Amazon Nova CPT. La personnalisation d'Amazon Nova CPT permet de mélanger facilement les données de domaine avec le corpus de pré-formation d'Amazon Nova. En outre, le taux d'échantillonnage des catégories de données spécifiques d'Amazon Nova (par exemple, code, mathématiques, raisonnement, etc.) peut être modifié et leurs proportions contrôlées pour compléter les données du domaine. Cela permet de renforcer les fonctionnalités qui correspondent au cas d'utilisation tout en adaptant le modèle au domaine spécifique.

Trouver le rapport de mélange optimal

Le ratio optimal entre les données Amazon Nova et les données de domaine dépend du domaine, de la complexité, de la taille, de la qualité du jeu de données et de l'importance de maintenir les fonctionnalités générales. Ce ratio doit être découvert par l'expérimentation. Voici un cadre d'expérimentation permettant de décider de la quantité de données Amazon Nova à mélanger.

Sélectionnez un sous-ensemble représentatif de données de domaine (par exemple, 5 milliards de jetons) et maintenez-le constant pendant tous les essais expérimentaux.

Exécutez des expériences CPT à petite échelle en faisant varier uniquement la quantité de données Amazon Nova mélangées dans :

- Pas de mixage : 100 % domaine → domaine 5 B uniquement (total 5 B)
- Mélange de lumière : domaine 90 % → domaine 5B + ~0,56 B Amazon Nova (total ~5,56 B)
- Mixage moyen : 70 % de domaine → domaine 5 B + ~2,14 B Amazon Nova (total ~7,14 B au total)
- Mixage intensif : 50 % de domaine → domaine 5 B + 5 Go Amazon Nova (total 10 B)

Évaluez chaque point de contrôle sur les benchmarks internes et généraux du domaine. Évaluez également le point de départ (point de contrôle Amazon Nova avant toute formation).

- Les performances du domaine client restent-elles à peu près constantes d'un cycle à l'autre ? C'est généralement le cas, car chaque exécution a généré le même nombre de jetons de domaine. Si les performances du domaine s'améliorent grâce à un plus grand mixage, les données Amazon Nova fournissent une régularisation utile.
- Les scores de référence généraux s'améliorent-ils à mesure que le mixage augmente ?
 - Le comportement attendu est que les fonctionnalités générales devraient s'améliorer de manière monotone à mesure que de nouvelles données Amazon Nova sont ajoutées.
 - Mesurez plusieurs repères généraux : MMLU (connaissances générales), HumanEval (codage), GSM8 K (mathématiques) ou repères spécifiques qui vous intéressent.
- Sélectionnez le ratio de mixage qui permet de maintenir les performances du domaine tout en offrant des fonctionnalités générales acceptables pour les cas d'utilisation. Tenez compte du coût supplémentaire de la formation en mélangeant davantage les données.

Une fois que le rapport de mélange optimal a été identifié, exécutez le CPT à grande échelle en utilisant l'ensemble de données de domaine complet avec le rapport de mélange sélectionné.

Disséquer les catégories de mixage de données

Nous analysons ci-dessous chaque catégorie disponible dans Data Mixing, afin que vous puissiez décider au mieux quelles catégories de données doivent être représentées dans votre mélange de données global.

Comment activer le mixage des données

Ajoutez la `data_mixing` section à votre recette avec la distribution en pourcentage appropriée entre les catégories de jeux de données. La somme des `nova_data` pourcentages doit être égale à 100.

Configuration Nova 2.0 avec mixage de données

```
# Note:
# This recipe can run on p5.48xlarge

# Run config
display_name: "Nova Lite Pretrain on P5 GPU"
versions: ["2.0"]
instance_types: ["ml.p5.48xlarge"]

run:
  name: "my-cpt-run"      # A descriptive name for your training job
  model_type: "amazon.nova-2-lite-v1:0:256k" # Model variant specification, do not
  change
  model_name_or_path: "nova-lite-2/prod" # Base model path, do not change
  replicas: 8              # Number of compute instances for training, allowed values are 4,
  8, 16, 32
  data_s3_path: ""        # Customer data paths
  validation_data_s3_path: "" # Customer validation data paths
  output_s3_path: ""      # Output artifact path, SageMaker HyperPod job-specific
  configuration - not compatible with standard SageMaker Training jobs

## Training specific configs
training_config:
  task_type: cpt
  max_length: 8192          # Maximum context window size (tokens)
  global_batch_size: 64     # Global batch size, allowed values are 32, 64, 128,
  256.

  trainer:
    max_steps: 10           # The number of training steps to run total
```



```

    val_check_interval: 10      # The number of steps between running validation
    limit_val_batches: 2        # Batches of the validation set to use each trigger

model:
    hidden_dropout: 0.0         # Dropout for hidden states, must be between 0.0 and
1.0
    attention_dropout: 0.0      # Dropout for attention weights, must be between 0.0
and 1.0

optim:
    optimizer: adam
    lr: 1e-5                    # Learning rate
    name: distributed_fused_adam # Optimizer algorithm, do not change
    adam_w_mode: true           # Enable AdamW mode
    eps: 1e-06                  # Epsilon for numerical stability
    weight_decay: 0.0           # L2 regularization strength, must be between 0.0 and
1.0
    adam_beta1: 0.9             # Beta1 for Adam optimizer
    adam_beta2: 0.95            # Beta2 for Adam optimizer
    sched:
        warmup_steps: 10        # Learning rate warmup steps
        constant_steps: 0        # Steps at constant learning rate
        min_lr: 1e-6            # Minimum learning rate, must be lower than lr

data_mixing:
    dataset_catalog: cpt_text_lite
    sources:
        nova_data: # percent inputs for Nova data must sum to 100%; use 0% if you want to
exclude a data grouping
        agents: 20
        business-and-finance: 4
        scientific: 10
        code: 5
        factual-and-news: 5
        longform-text: 6
        health-and-medicine: 1
        humanities-and-education: 1
        legal: 1
        math: 9
        additional-languages: 15
        social-and-personal-interest: 11
        entertainment: 0.5
        reasoning: 10
        other: 0.5

```

```

tables: 1
customer_data: # percent input of customer data. 100 = use only customer data, 0 =
use only the nova_data mix above
percent: 25

```

Que signifient ces catégories

Remarque : Nova 2.0 inclut des catégories supplémentaires spécifiques au raisonnement (par exemple, `reasoning-codereasoning-math`, `reasoning-instruction-following`) qui ne sont pas disponibles dans Nova 1.0.

Résumé des catégories et des étiquettes d'information :

Nom de la catégorie	Détail de l'information
agents	Données de formation axées sur la prise de décision autonome, l'exécution des tâches et le comportement axé sur les objectifs dans les systèmes d'IA
baseline	Données linguistiques fondamentales axées sur la compréhension générale, la communication de base et les capacités linguistiques de base
chat	Des échanges conversationnels démontrant un flux de dialogue naturel, le maintien du contexte et des interactions sociales appropriées
code	Code source de programmation, documentation et discussions techniques à partir de différents langages de programmation et plateformes.
factuality	Matériaux de référence et informations vérifiées axés sur l'exactitude, la validation des sources et l'évaluation de la vérité
identity	Cadres de personnalité et modèles comportementaux axés sur des traits de caractère, des valeurs et des styles d'interaction cohérents
long-context	Des textes étendus et des récits complexes axés sur le maintien de la cohérence et de la pertinence lors de longs échanges
math	Contenu mathématique, y compris les manuels, les problèmes, les solutions et les discussions mathématiques.

Nom de la catégorie	Détail de l'information
rai	Cas et scénarios mettant l'accent sur les principes éthiques de l'IA, les considérations de sécurité et le déploiement responsable des technologies
instruction-following	Exemples d'exécution précise de tâches basées sur différents niveaux d'instructions et de directives utilisateur
stem	Contenu technique couvrant les sciences, la technologie, l'ingénierie et les mathématiques, y compris la résolution de problèmes et les concepts théoriques
planning	Séquences illustrant la réflexion stratégique, la répartition des step-by-step tâches et l'allocation efficace des ressources
reasoning-chat	Scénarios de dialogue analytique axés sur des discussions logiques et des flux de conversation structurés
reasoning-code	Défis de programmation et problèmes algorithmiques axés sur le développement systématique de solutions
reasoning-factuality	Scénarios d'évaluation des informations axés sur les processus critiques d'évaluation et de vérification
reasoning-instruction-following	Analyse de tâches complexes axée sur l'interprétation systématique et l'exécution méthodique
reasoning-math	Scénarios mathématiques de résolution de problèmes axés sur la progression logique et les stratégies de résolution
reasoning-planning	Scénarios de prise de décision stratégique axés sur une approche systématique de la réalisation des objectifs
reasoning-rag	Scénarios de recherche et de synthèse d'informations axés sur la compréhension contextuelle et l'application pertinente

Nom de la catégorie	Détail de l'information
reasoning-rai	Scénarios de prise de décision éthique axés sur l'évaluation systématique de la sécurité et de l'équité de l'IA
reasoning-stem	Scénarios scientifiques de résolution de problèmes axés sur l'analyse méthodique et le développement de solutions
rag	Exemples de combinaison efficace de connaissances externes récupérées avec des réponses générées pour fournir des informations contextuelles précises
translation	Paires de contenus multilingues présentant une traduction précise tout en préservant le contexte, le ton et les nuances culturelles

Guide des paramètres

- `dataset_catalog` : La seule valeur est `cpt_text_lite` pour le moment, jusqu'à ce que nous activions l'entraînement multimodal.
- `nova_data` : pourcentage des catégories individuelles de données Nova lorsqu'elles sont mélangées. Leur somme devrait atteindre 1,0.
- `customer_data` : le pourcentage de données du client mélangées aux données Nova.

Le nombre total de jetons utilisés pendant l'entraînement peut être calculé à partir de $\text{max\_length} * \text{global\_batch\_size} * \text{max\_steps}$

Limitations

Le CPT actuel ne prend en charge que les données texte et ne prend en charge aucun ensemble de données multimodaux des clients.

Réglage précis supervisé

Introduction

Le réglage fin supervisé utilise un ensemble de données avec des paires d'entrées-sorties pour la tâche qui vous intéresse. En d'autres termes, vous fournissez des exemples d'instructions (questions,

instructions, etc.) ainsi que les réponses correctes ou souhaitées et vous poursuivez la formation du modèle sur ces questions. Les poids du modèle sont ajustés pour minimiser les pertes supervisées, généralement une entropie croisée entre ses prédictions et les jetons de réponse cibles.

Quand utiliser SFT ?

SFT est préférable lorsque vous avez une tâche bien définie avec des résultats souhaités clairs. Si vous pouvez dire explicitement « Étant donné une entrée X, la correct/desired sortie est Y » et que vous pouvez recueillir des exemples de tels mappages X-Y, le réglage fin supervisé est un excellent choix. Parmi les scénarios dans lesquels SFT excelle, citons :

- Tâches de classification structurées ou complexes : par exemple, classification de documents internes ou de contrats dans de nombreuses catégories personnalisées. Avec SFT, le modèle peut apprendre ces catégories spécifiques bien mieux que de le demander seul.
- Tâches de réponse à des questions ou de transformation avec des réponses connues : par exemple, affiner un modèle pour répondre aux questions de la base de connaissances d'une entreprise, ou pour convertir des données entre formats, chaque entrée ayant une réponse correcte.
- Cohérence du formatage et du style : si vous souhaitez que le modèle réponde toujours dans un certain format ou dans un certain ton, vous pouvez affiner les exemples de format/de tonalité appropriés. Par exemple, une formation sur des paires prompt-réponse illustrant une voix ou un style de marque en particulier peut apprendre au modèle à utiliser ce style dans ses sorties. Le comportement suivant les instructions est souvent initialement enseigné via SFT sur des exemples sélectionnés de bons comportements d'assistant.

SFT est le moyen le plus direct d'enseigner à un LLM une nouvelle compétence ou un nouveau comportement lorsque vous pouvez spécifier à quoi ressemble le bon comportement. Il tire parti de la compréhension linguistique existante du modèle et le concentre sur votre tâche. N'utilisez pas le SFT lorsque l'écart est lié aux connaissances plutôt qu'au comportement ; cela n'obligera pas le modèle à apprendre de nouveaux faits, du jargon ou des événements récents. Dans ces cas, préférez une formation préalable continue sur de grands corpus internes au domaine ou une génération augmentée par extraction afin d'intégrer des connaissances externes à des fins d'inférence. Lorsque vous pouvez mesurer la qualité mais que vous ne pouvez pas identifier une seule bonne réponse, il LLM-as-judge peut être préférable de peaufiner le renforcement avec des récompenses vérifiables ou une réponse au SFT.

En fonction de la complexité des tâches et des performances du modèle Nova sans réglage, planifiez des milliers, voire des dizaines de milliers de démonstrations par tâche, la qualité, la cohérence et la diversité des données étant plus importantes que le volume brut.

Quand utiliser le paramètre Efficient et quand utiliser le SFT à rang complet ?

Les recettes de personnalisation Nova vous permettent d'exécuter des paramètres efficaces, en particulier LoRa ou SFT de rang complet. Si vous souhaitez une mise à jour de modèle simple et rentable, ou si vous disposez de très peu de données, privilégiez des méthodes économes en paramètres afin de former de petits adaptateurs sans toucher à la majeure partie de l'épine dorsale (le SFT complet met à jour tous les paramètres du modèle).

Mixage de données pour SFT

Le mixage des données vous permet de combiner vos ensembles de données d'entraînement personnalisés avec les données d'entraînement exclusives de Nova. Cette fonctionnalité est disponible pour les modèles Nova 1.0 et Nova 2.0.

Type de données propriétaire Nova : Nova prend en charge les types de données SFT textuels et multimodaux. Il est organisé en plusieurs catégories de données, chacune contenant un mélange de tâches pertinentes pour la catégorie correspondante.

Catégories de données propriétaires Nova : Les ensembles de données textuels incluent plusieurs catégories, notamment : prise de décision autonome, achèvement des tâches, ensembles de données orientés vers les objectifs (agents), ensembles de données d'exécution précise des tâches avec ou sans raisonnement (suivi des instructions) reasoning-instruction-following, séquences démontrant une réflexion stratégique et une répartition des step-by-step tâches (planification), IA responsable (rai), contexte long, factualité, mathématiques, STEM et bien d'autres encore. De même, les ensembles de données multimodaux incluent des vidéos, des captures d'écran, des graphiques et bien d'autres.

La fonction de mixage des données vous permet de mélanger vos propres échantillons d'entraînement avec des échantillons provenant des ensembles de données Nova utilisés pour affiner le Nova. Cela peut éviter de trop adapter votre entraînement personnalisé et d'« oublier de manière catastrophique » les capacités de Nova, ou vous aider à renforcer vos capacités lorsque vous vous entraînez à partir d'un nouveau point de contrôle préentraîné.

Pour intégrer des données Nova, il vous suffit d'ajouter un bloc `data_mixing` à votre fichier YAML de recette, dans la section `training_config`. Les blocs de mélange de texte et de données multimodales ont un contenu différent. Veuillez vous référer aux recettes correspondantes.

Modèles pris en charge

- Nova 2.0 Lite

Modalité prise en charge

- Texte
- Multimodal

Exemples de configuration YAML

Exemple de configuration de Nova 2.0

```
run:
  name: my-lora-sft-run
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: nova-lite-2/prod
  data_s3_path: s3://my-bucket-name/train.jsonl
  replicas: 4
  output_s3_path: s3://my-bucket-name/outputs/
  mlflow_tracking_uri: ""
  mlflow_experiment_name: "my-lora-sft-experiment"
  mlflow_run_name: "my-lora-sft-run"

training_config:
  max_steps: 100
  save_steps: 10
  save_top_k: 5
  max_length: 32768
  global_batch_size: 32
  reasoning_enabled: true
  lr_scheduler:
    warmup_steps: 15
    min_lr: 1e-6
  optim_config:
```

```
lr: 1e-5
weight_decay: 0.0
adam_beta1: 0.9
adam_beta2: 0.95
peft:
  peft_scheme: "lora"
  lora_tuning:
    alpha: 64
    lora_plus_lr_ratio: 64.0
```

Mixage de données texte Nova 2.0

```
data_mixing:
  dataset_catalog: sft_1p5_text_chat
  sources:
    customer_data:
      percent: 50
    nova_data:
      agents: 1
      baseline: 10
      chat: 0.5
      code: 10
      factuality: 0.1
      identity: 1
      long-context: 1
      math: 2
      rai: 1
      instruction-following: 13
      stem: 0.5
      planning: 10
      reasoning-chat: 0.5
      reasoning-code: 0.5
      reasoning-factuality: 0.5
      reasoning-instruction-following: 45
      reasoning-math: 0.5
      reasoning-planning: 0.5
      reasoning-rag: 0.4
      reasoning-rai: 0.5
      reasoning-stem: 0.4
      rag: 1
      translation: 0.1
```


Mélange de données multimodales Nova 2.0

```
data_mixing:
  dataset_catalog: sft_1p5_mm_chat
  sources:
    customer_data:
      percent: 50
    nova_data:
      charts: 1
      chat: 38
      code: 20
      docs: 3
      general: 2
      grounding: 1
      rag: 4
      screenshot: 4
      text: 8
      translation: 4
      video: 15
```

Modéliser les points de contrôle

Points de contrôle Nova 2.0

- **PRÉ-ENTRAÎNÉ** [nova-lite-2/pretraining-text-RD] : Point de contrôle après un taux d'apprentissage constant et des étapes de réduction progressive au cours desquelles le modèle est entraîné sur des milliards de jetons. [Résultat de l'étape 2]
- **FORMATION INTERMÉDIAIRE** [nova-lite-2/pretraining-text-CE] : Permet aux clients disposant de volumes intermédiaires de données non structurées d'introduire leurs données avec un taux d'apprentissage plus modéré qu'avant la formation, absorbant ainsi les connaissances spécifiques au domaine tout en évitant un oubli catastrophique. [Résultat de l'étape 3]
- **FINAL** [nova-lite-2/prod] : Point de contrôle final entièrement aligné qui a suivi toutes les étapes avant et après l'entraînement. [Résultat de l'étape 4]

Étapes de formation :

- Étape 1 : PT Ckpt, pré-formation initiale avec un taux d'apprentissage constant
- Étape 2 : PT Copt, réduction du taux d'apprentissage

- Étape 3 : PT Copt, formation d'extension du contexte
- Étape 4 : alignement suivant les instructions et formation à la sécurité

Approches de formation

Guide de sélection de l'approche de formation

Type de données	Volume de données	Exécuter	Avec Checkpoint
Données de domaine brutes non structurées à grande échelle (documents, journaux, articles, code, etc.)	Jetons 1T+	Formation préalable continue	Fin du taux d'apprentissage constant (CLR)
Données de domaine brutes non structurées à grande échelle	Plus de 100 milliards de jetons	Au milieu de l'entraînement	Fin du CLR
Petits volumes de données brutes non structurées ; traces de raisonnement structurées/données CoT	Plus de 1 milliard de jetons	Au milieu de l'entraînement	Modèle de base Nova
Démonstrations structurées (paires entrées-sorties de haute qualité, instructions de tâches sélectionnées, dialogues à plusieurs tours)	Plus de 1 000 exemples	Réglage fin supervisé (SFT)	Modèle de base Nova

Pré-requis avant de commencer

- Nous supposons que vous avez déjà configuré un cluster SMHP avec un groupe d'instances restreint (RIG) doté d'une capacité active. Si ce n'est pas le cas, veuillez vous référer ici pour terminer la configuration de votre cluster SMHP et de votre RIG [[Docs Link](#), [Workshop Link](#)]
- Vous aurez besoin d'instances EC2 p5.48xlarge pour exécuter cette recette. Le nombre minimum d'instances requis pour exécuter efficacement cette recette est le suivant :
 - Nova Lite 2.0 - 4 p5.48 x large
- Installez la SageMaker HyperPod CLI spécifique à Forge en suivant les instructions fournies [ici](#)
- Vérifiez que vous pouvez vous connecter à votre cluster à l'aide de `hyperpod get-clusters`
 - Notez que cette commande listera tous les clusters SMHP de votre compte
- Vérifiez que votre formation, et éventuellement les données de validation, sont disponibles dans un compartiment S3 accessible par le rôle d'exécution de votre cluster SMHP. Pour la préparation des données, reportez-vous à la section suivante.
- La configuration est terminée. Si vous n'avez pas terminé la configuration, veuillez suivre le [guide](#) ci-dessous.
- Vérification : Après avoir terminé la configuration, confirmez que vous pouvez exécuter avec succès les commandes ci-dessous

```
aws sagemaker describe-cluster --cluster-name <cluster-name> --region <region>

hyperpod connect-cluster --cluster-name cluster-name
```

Une approche systématique pour réussir le SFT

- Préparation des données : suivez les directives établies pour créer, nettoyer ou reformater les ensembles de données selon la structure requise. Assurez-vous que les entrées, les sorties et les informations auxiliaires (telles que les traces de raisonnement ou les métadonnées) sont correctement alignées et formatées.
- Configuration de l'entraînement : définissez la manière dont le modèle sera entraîné. Lorsque vous utilisez Amazon SageMaker HyperPod, cette configuration est écrite dans un fichier de recette YAML qui inclut :

- Chemins des sources de données (ensembles de données de formation et de validation)
- Hyperparamètres clés (nombre d'étapes d'apprentissage, taux d'apprentissage, taille du lot)
- Composants optionnels (paramètres d'entraînement distribués, etc.)
- Paramètre de mixage des données (définit les proportions des catégories de données client et Nova)
- Optimisation des hyperparamètres SFT : les valeurs des paramètres de recette SFT que nous recommandons constituent un excellent point de départ et un choix solide. Si vous souhaitez les optimiser davantage pour votre cas d'utilisation, effectuez plusieurs exécutions SFT avec différentes combinaisons de paramètres et choisissez la meilleure. Vous pouvez sélectionner des combinaisons de paramètres en suivant la méthode d'optimisation des hyperparamètres de votre choix. Une approche simple consiste à faire varier la valeur d'un paramètre (default*0.5, default, default*2) tout en conservant une autre valeur par défaut pour les autres paramètres, à répéter cette opération pour chaque paramètre que vous souhaitez optimiser et à itérer si nécessaire. Les paramètres les plus pertinents pour LoRa sont le taux d'apprentissage, l'alpha (paramètre de mise à l'échelle), le nombre d'époques d'entraînement et les étapes d'échauffement ; pour le grade complet, il s'agit principalement du taux d'apprentissage, du nombre d'époques et des étapes d'échauffement.

Séquençage d'expériences et mélange de données

- Si vous ne disposez que de données SFT (train/dev/test) pour un ensemble de tâches et que vous ne vous souciez que des performances des tests sur ces tâches
 - Effectuez le SFT sans mixer sur le point de contrôle [FINAL] Nova. Utilisez les hyperparamètres SFT par défaut et optimisez-les éventuellement pour votre cas d'utilisation. Surveillez les métriques de validation et and/or évaluez les points de contrôle intermédiaires pour des ensembles de données plus importants.
- Si vous ne disposez que de données SFT (train/dev/test) pour un ensemble de tâches et que vous vous souciez des performances des tests sur ces tâches et des points de référence généraux dans le domaine qui vous intéresse
 - Commencez par mixer les données SFT avec Nova sur un point de contrôle préalable à l'entraînement (point de contrôle PRÉ-ENTRAÎNÉ ou MI-ENTRAÎNÉ, pas FINAL). L'utilisation d'un point de contrôle intermédiaire permet au modèle de mieux intégrer vos données personnalisées aux données propriétaires de Nova tout en conservant de solides capacités générales.

- Exécutez des sessions d'entraînement SFT plus courtes avec une quantité variable de données Nova dans le mix (par exemple, 10 %, 25 %, 50 %, 75 %) et des sélections de catégories de données Nova qui complètent votre cas d'utilisation (par exemple, choisissez les instructions par catégorie si vous vous souciez des instructions générales suivant les capacités). Surveillez les métriques de validation et évaluez si le mixage améliore les performances par rapport aux benchmarks généraux. Sélectionnez le mix d'entraînement et le point de contrôle qui vous permettront d'obtenir la meilleure combinaison de performance sur votre tâche et de performance générale. Selon le cas d'utilisation, les tâches et les performances générales peuvent être encore améliorées grâce au réglage fin du renforcement (RFT).

Préparer le jeu de données pour SFT

Nova 2.0 : utilisez le format <https://docs.aws.amazon.com/bedrock/latest/userguide/conversation-inference-call.html> de l'API Converse. Le format de données Nova 2.0 peut contenir des champs de raisonnement supplémentaires : https://docs.aws.amazon.com/bedrock/latest/APIReference/APIReasoningContentBlock_runtime_.html

Le contenu du raisonnement capture les étapes de réflexion intermédiaires du modèle avant de générer une réponse finale. À votre assistant tour, utilisez le `reasoningContent` champ pour inclure des traces de raisonnement. Utilisez du texte brut pour raisonner le contenu, évitez les balises de balisage telles `</thinking>` que `<thinking>` et sauf si cela est spécifiquement requis par votre tâche, et assurez-vous que le contenu du raisonnement est clair et pertinent pour le processus de résolution des problèmes.

Méthodes d'évaluation

Conditions préalables

- Vérifiez l'URI S3 depuis le `manifest.json` fichier de votre tâche de formation (pour les modèles entraînés)
- Ensemble de données d'évaluation chargé dans S3 dans le bon format
- Chemin de sortie S3 pour les résultats d'évaluation

Benchmarks prêts à l'emploi : utilisez des benchmarks prêts à l'emploi pour valider les performances sur des tâches générales. Pour plus de détails, rendez-vous ici : <https://docs.aws.amazon.com/sagemaker/latest/dg/nova-hp-evaluate.html>

Apportez vos propres données

Vous pouvez également fournir vos données personnalisées en les formatant dans le format ci-dessous, puis en utilisant les conteneurs mentionnés ci-dessous pour obtenir des résultats d'inférence ainsi que des probabilités logarithmiques pour les étalonnages si nécessaire.

Créez jsonl par tâche avec la structure suivante :

```
{
  "metadata": "{key:4, category:'apple'}",
  "system": "arithmetic-patterns, please answer the following with no other words: ",
  "query": "What is the next number in this series? 1, 2, 4, 8, 16, ?",
  "response": "32"
}
```

Les résultats générés pendant la phase d'inférence du travail d'évaluation auront la structure suivante :

```
{
  "prompt": "[{'role': 'system', 'content': 'arithmetic-patterns, please answer the following with no other words: '}, {'role': 'user', 'content': 'What is the next number in this series? 1, 2, 4, 8, 16, ?'}]",
  "inference": "['32']",
  "gold": "32",
  "metadata": "{key:4, category:'apple'}"
}
```

Descriptions des champs :

- **prompt**: entrée formatée envoyée au modèle
- **inference**: réponse générée par le modèle
- **gold**: réponse correcte attendue de l'ensemble de données en entrée, champ de réponse de l'entrée
- **metadata**: métadonnées facultatives transmises depuis l'entrée

Préparer la configuration d'évaluation

Commande pour lancer la tâche d'évaluation. "--override-parameters" À utiliser pour modifier n'importe quelle entrée de la recette.

```
hyperpod start-job -n kubeflow \  
  --recipe evaluation/nova/nova_micro_p5_48xl_bring_your_own_dataset_eval \  
  --override-parameters '{  
    "instance_type": "p5.48xlarge",  
    "container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-  
HP-Eval-latest",  
    "recipes.run.name": "<your-eval-job-name>",  
    "recipes.run.model_name_or_path": "<checkpoint-s3-uri>",  
    "recipes.run.output_s3_path": "s3://<your-bucket>/eval-results/",  
    "recipes.run.data_s3_path": "s3://<your-bucket>/eval-data.jsonl"  
  }'
```

Bonnes pratiques

- Privilégiez la qualité des données au volume : des données de formation de haute qualité, diversifiées et représentatives sont plus précieuses que de grandes quantités de données de faible qualité.
- Inclure reasoning-instruction-following la catégorie : lorsque vous utilisez le mixage des données, incluez la catégorie reasoning-instruction-following « » pour maintenir de bonnes performances génériques entre les tâches.
- Utiliser les taux d'apprentissage par défaut : commencez par les taux d'apprentissage par défaut (1e-5 pour LoRa, 5e-6 pour le SFT de haut rang) et ajustez uniquement si nécessaire en fonction des métriques de validation.
- Mixage de données Balance Nova : mélangez un maximum de 50 % de données Nova pour un équilibre optimal entre latence et performances. Des pourcentages plus élevés peuvent améliorer les capacités générales mais peuvent augmenter le temps d'entraînement.
- Surveillez les mesures de validation : évaluez régulièrement les points de contrôle intermédiaires pendant l'entraînement afin de détecter rapidement le surajustement ou la dégradation des performances.

- Testez sur des ensembles de données représentatifs : assurez-vous que vos ensembles de données d'évaluation représentent précisément vos cas d'utilisation en production pour une évaluation significative des performances.

Préparer la configuration du Job de formation

Hyper paramètres

Ensemble complet d'hyperparamètres autres que le mixage de données :

```
## Run config
run:
  name: my-lora-sft-run
  model_type: amazon.nova-2-lite-v1:0:256k
  model_name_or_path: nova-lite-2/prod
  data_s3_path: s3://my-bucket-name/train.jsonl # SageMaker HyperPod (SMHP) only and
not compatible with SageMaker Training jobs. Note replace my-bucket-name with your
real bucket name for SMHP job
  replicas: 4 # Number of compute instances for training, allowed
values are 4, 8, 16, 32
  output_s3_path: s3://my-bucket-name/outputs/ # Output artifact path
(Hyperpod job-specific; not compatible with standard SageMaker Training jobs). Note
replace my-bucket-name with your real bucket name for SMHP job

## MLFlow configs
mlflow_tracking_uri: "" # Required for MLFlow
mlflow_experiment_name: "my-lora-sft-experiment" # Optional for MLFlow. Note: leave
this field non-empty
mlflow_run_name: "my-lora-sft-run" # Optional for MLFlow. Note: leave this field non-
empty

training_config:
  max_steps: 100 # Maximum training steps. Minimal is 4.
  save_steps: 10 # This parameter suggests after how many training steps the
checkpoints will be saved. Should be less than or equal to max_steps(please override
this value with a numerical value equal or less than max_steps value; min: 4)
  save_top_k: 5 # Keep top K best checkpoints. Note supported only
for SageMaker HyperPod jobs. Minimal is 1.
  max_length: 32768 # Sequence length (options: 8192, 16384, 32768
[default], 65536)
  global_batch_size: 32 # Global batch size (options: 32, 64, 128)
```



```

reasoning_enabled: true          # If data has reasoningContent, set to true;
otherwise False

lr_scheduler:
  warmup_steps: 15               # Learning rate warmup steps. Recommend 15% of
max_steps
  min_lr: 1e-6                  # Minimum learning rate, must be between 0.0 and 1.0

optim_config:                   # Optimizer settings
  lr: 1e-5                      # Learning rate, must be between 0.0 and 1.0
  weight_decay: 0.0             # L2 regularization strength, must be between 0.0
and 1.0
  adam_beta1: 0.9               # Exponential decay rate for first-moment estimates,
must be between 0.0 and 1.0
  adam_beta2: 0.95              # Exponential decay rate for second-moment
estimates, must be between 0.0 and 1.0

peft:                           # Parameter-efficient fine-tuning (LoRA)
  peft_scheme: "lora"           # Enable LoRA for PEFT
  lora_tuning:
    alpha: 64                   # Scaling factor for LoRA weights ( options: 32, 64,
96, 128, 160, 192),
    lora_plus_lr_ratio: 64.0     # LoRA+ learning rate scaling factor (0.0-100.0)

```

Les paramètres les plus pertinents pour LoRa sont le taux d'apprentissage, l'alpha (paramètre de mise à l'échelle), le nombre d'époques d'entraînement et les étapes d'échauffement ; pour le grade complet, il s'agit principalement du taux d'apprentissage, du nombre d'époques et des étapes d'échauffement. Les recettes sont préremplies avec les valeurs par défaut recommandées.

Configurer un bloc de mixage de données

Ajoutez la section `data_mixing` à votre recette avec la distribution en pourcentage appropriée entre les catégories de jeux de données.

Nous décrivons ci-dessous chaque catégorie de données Nova disponible.

Configuration Nova 2.0 avec mixage de données

```

data_mixing:
  dataset_catalog: sft_1p5_text_chat      # Nova text dataset catalog
  sources:
    customer_data:

```

percent: 50	# Percent of overall mix to draw from customer data
nova_data:	# The remainder will be drawn from Nova data. The
categories below must add to 100	
agents: 1	# autonomous decision-making, task completion, goal-
oriented behavior in AI systems	
baseline: 10	# [New in Nova 1.5]
chat: 0.5	# Conversational exchanges demonstrating natural
dialogue flow	
code: 10	# Programming examples and solutions spanning
multiple languages	
factuality: 0.1	# [New in Nova 1.5]
identity: 1	# [New in Nova 1.5]
long-context: 1	# [New in Nova 1.5]
math: 2	# [New in Nova 1.5]
rai: 1	# ethical AI principles, safety considerations, and
responsible technology deployment	
instruction-following: 13	# precise task execution based on varying levels of
user prompts and directives	
stem: 0.5	# Technical content covering science, technology,
engineering, and mathematics	
planning: 10	# Sequences demonstrating strategic thinking and
step-by-step task breakdown	
reasoning-chat: 0.5	
reasoning-code: 0.5	
reasoning-factuality: 0.5	
reasoning-instruction-following: 45	
reasoning-math: 0.5	
reasoning-planning: 0.5	
reasoning-rag: 0.4	
reasoning-rai: 0.5	
reasoning-stem: 0.4	
rag: 1	# combining retrieved external knowledge with
generated responses	
translation: 0.1	

Que signifient ces catégories ?

Catégories de données texte Nova 2.0

Nom de la catégorie	Détail de l'information
agents	Données de formation axées sur la prise de décision autonome, l'exécution des tâches et

Nom de la catégorie	Détail de l'information
	le comportement axé sur les objectifs dans les systèmes d'IA
référence	Données linguistiques fondamentales axées sur la compréhension générale, la communication de base et les capacités linguistiques de base
chat	Des échanges conversationnels démontrant un flux de dialogue naturel, le maintien du contexte et des interactions sociales appropriées
code	Code source de programmation, documentation et discussions techniques à partir de différents langages de programmation et plateformes.
factualité	Matériaux de référence et informations vérifiées axés sur l'exactitude, la validation des sources et l'évaluation de la vérité
une	Cadres de personnalité et modèles comportementaux axés sur des traits de caractère, des valeurs et des styles d'interaction cohérents
contexte long	Des textes étendus et des récits complexes axés sur le maintien de la cohérence et de la pertinence lors de longs échanges
math	Contenu mathématique, y compris les manuels, les problèmes, les solutions et les discussions mathématiques.
rai	Cas et scénarios mettant l'accent sur les principes éthiques de l'IA, les considérations de sécurité et le déploiement responsable des technologies

Nom de la catégorie	Détail de l'information
suivre les instructions	Exemples d'exécution précise de tâches basées sur différents niveaux d'instructions et de directives utilisateur
radical	Contenu technique couvrant les sciences, la technologie, l'ingénierie et les mathématiques, y compris la résolution de problèmes et les concepts théoriques
planification	Séquences illustrant la réflexion stratégique, la répartition des step-by-step tâches et l'allocation efficace des ressources
discussion raisonnée	Scénarios de dialogue analytique axés sur des discussions logiques et des flux de conversation structurés
code de raisonnement	Défis de programmation et problèmes algorithmiques axés sur le développement systématique de solutions
raisonnement-factualité	Scénarios d'évaluation des informations axés sur les processus critiques d'évaluation et de vérification
reasoning-instruction-following	Analyse de tâches complexes axée sur l'interprétation systématique et l'exécution méthodique
mathématiques du raisonnement	Scénarios mathématiques de résolution de problèmes axés sur la progression logique et les stratégies de résolution
planification du raisonnement	Scénarios de prise de décision stratégique axés sur une approche systématique de la réalisation des objectifs

Nom de la catégorie	Détail de l'information
chiffon de raisonnement	Scénarios de recherche et de synthèse d'informations axés sur la compréhension contextuelle et l'application pertinente
Reasoning-Rai	Scénarios de prise de décision éthique axés sur l'évaluation systématique de la sécurité et de l'équité de l'IA
radical de raisonnement	Scénarios scientifiques de résolution de problèmes axés sur l'analyse méthodique et le développement de solutions
chiffon	Exemples de combinaison efficace de connaissances externes récupérées avec des réponses générées pour fournir des informations contextuelles précises
traduction	Paires de contenus multilingues présentant une traduction précise tout en préservant le contexte, le ton et les nuances culturelles

Mixage de données multimodal (Nova 2.0)

```

data_mixing:
  dataset_catalog: sft_1p5_mm_chat          # Nova text dataset catalog
  sources:
    customer_data:
      percent: 50                          # Percent of overall mix to draw from customer data
    nova_data:                             # The remainder will be drawn from Nova data. The
categories below must add to 100
  charts: 1
  chat: 38
  code: 20
  docs: 3
  general: 2
  grounding: 1

```

```
rag: 4
screenshot: 4
text: 8
translation: 4
video: 15
```

Remarque : Nova 2.0 inclut la prise en charge de catégories de données vidéo qui n'est pas disponible dans Nova 1.0.

Que signifient ces catégories ?

Catégories de données multimodales Nova 2.0

Nom de la catégorie	Détail de l'information
graphiques	Représentations visuelles et descriptions de graphiques, de diagrammes à secteurs, de diagrammes à barres, de diagrammes linéaires et d'autres visualisations statistiques pour aider le modèle à comprendre et à communiquer efficacement les informations quantitatives
chat	Des données conversationnelles associées à des éléments visuels axés sur la compréhension du dialogue contextuel et les interactions basées sur des images
code	Interfaces de programmation et environnements de développement axés sur l'interprétation visuelle du code, les captures d'écran IDE et les diagrammes techniques
médecins	Des données centrées sur le document combinant du texte, des images, des mises en page et des mises en forme pour apprendre aux modèles à comprendre et à traiter différents types et structures de documents afin de faciliter l'utilisation de concepts tels que la reconnaissance de contenu PDF

Nom de la catégorie	Détail de l'information
general	Contenu visuel et textuel diversifié axé sur une large compréhension des images, des graphiques et du texte descriptif qui les accompagne
ancrage	Des matériaux de référence visuels et des images étiquetées axés sur le lien entre les concepts linguistiques et les représentations visuelles du monde réel
chiffon	Exemples de récupération multimodale montrant comment combiner et référencer efficacement des connaissances externes visuelles et textuelles afin de générer des réponses contextuelles précises
capture d'écran	Captures d'interface d'application et affichage numérique d'images axées sur la compréhension des interfaces logicielles et des interactions numériques
text	Un pool équilibré de données textuelles contextuelles créées à partir des catégories de jeux de données SFT Nova contenant uniquement du texte afin de fournir des capacités généralistes
traduction	Contenu visuel multilingue axé sur l'interprétation multilingue du texte sous forme d'images et d'éléments visuels culturels
vidéos	Contenu visuel basé sur le mouvement axé sur la compréhension temporelle et la compréhension séquentielle visuo-narrative

Comment lancer une offre d'emploi

Vous pouvez également vous référer au README, si vous n'avez besoin que des informations essentielles pour lancer la première exécution de SFT.

Informations sur le contenant :

Informations sur le conteneur et commandes de lancement

Modèle	Technique	Sous-catégorie	URI de l'image	Commande Hyperpod Launcher
Nova 2.0	Peaufinage	Texte SFT	708977205387.dkr.ecr.us-east-1.amazonaws.com/:SM-HP-SFT-v2-Latest nova-fine-tune-repo	hyperpod start-job \ -n kubeflow \ -- recipe fine- -:SM-HP-SFT-v2-lat est »} 'tuning/ nova/nova_2_0/nova_lite/SFT/nova_lite_2_0_p5_gpu_sft \ --override-parameters '{ "instance_type": "ml.p5.48xlarge", "container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/novafine-tune-repo
Nova 2.0	Peaufinage	Texte SFT + Mixage de données	708977205387.dkr.ecr.us-east-1.amaz	hyperpod start-job \ -n kubeflow \ --

Modèle	Technique	Sous-catégorie	URI de l'image	Commande Hyperpod Launcher
			onaws.com /:SM-HP-SFT-V2-Datamixing-Latest nova-fine-tune-repo	recipe fine- -:SM-HP-SFT-V2-Datamixing-Latest »} 'tuning/nova/ forge/nova_2_0 /nova_lite/SFT/ nova_lite_2_0_p5_gpu_ sft_text_ with_datamix \ -- override-paramet ers '{ "instance _type": "ml.p5.48 xlarge", "container": "70897720 5387.dkr.ecr.us- east-1.amaz onaws.com/ novafine-tune- repo

Modèle	Technique	Sous-catégorie	URI de l'image	Commande Hyperpod Launcher
Nova 2.0	Peaufinage	1 MM DOUX	708977205387.dkr.ecr.us-east-1.amazonaws.com/:SM-HP-SFT-v2-Latest nova-fine-tune-repo	hyperpod start-job \ -n kubeflow \ --recipe fine- -:SM-HP-SFT-v2-latest »} 'tuning/nova/nova_2_0/nova_lite/SFT/nova_lite_2_0_p5_gpu_sft \ --override-parameters '{ "instance_type": "ml.p5.48xlarge", "container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/novafine-tune-repo

Modèle	Technique	Sous-catégorie	URI de l'image	Commande Hyperpod Launcher
Nova 2.0	Peaufinage	SFT MM + Mixage de données	708977205387.dkr.ecr.us-east-1.amazonaws.com /:SM-HP-SFT-V2-Datamixing-Latest nova-fine-tune-repo	hyperpod start-job \ -n kubeflow \ -- recipe fine- -:SM-HP-SFT-V2-Datamixing-Latest »} 'tuning/nova/forge/nova_2_0/nova_lite/SFT/nova_lite_2_0_p5_gpu_sft_mm_with_datamix \ -- override-parameters '{ "instance_type": "ml.p5.48xlarge", "container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/novafine-tune-repo

Une fois que tout est configuré, en commençant par la racine du sagemaker-hyperpod-cli référentiel, accédez au dossier de recettes par défaut de nova sft

- `cd/src/hyperpod_cli/sagemaker_hyperpod_recipes/recipes_collection/recipes/training/nova/`
- Ici, vous pouvez choisir d'exécuter les recettes nova 1 ou nova 2 en fonction du modèle de base choisi.

Pour Nova 2.0 sft :

- Si vous souhaitez utiliser une tâche logicielle normale, vous devriez pouvoir voir une recette dans ce dossier
 - `cd/src/hyperpod_cli/sagemaker_hyperpod_recipes/recipes_collection/recipes/fine-tuning/nova_2_0/nova_lite/SFTet` alors vous devriez pouvoir voir une recette dans ce dossier appelée `nova_lite_2_0_p5x8_gpu_sft.yaml`
- Si vous souhaitez utiliser datamixing sft Job, vous pouvez accéder au dossier des recettes de sft Forge
 - `cd/src/hyperpod_cli/sagemaker_hyperpod_recipes/recipes_collection/recipes/fine-tuning/nova/forge/nova_2_0/nova_lite/SFTet` et vous devriez pouvoir voir une recette dans ce dossier appelée : `nova_lite_2_0_p5x8_gpu_sft_with_datamix.yaml`
- Modifiez les sections de la recette requise par la tâche, telles que `name`, `data_s3_path`, `validation_s3_path`, `output_s3_path` et `max_steps`. Puisque nous exécutons sft, la notion d'époque ne s'applique pas ici.

La configuration de mixage des données aura la même apparence, mais avec une section de mixage de données supplémentaire en bas, similaire à celle-ci

```
data_mixing:
  dataset_catalog: sft_text_lite
  sources:
    nova_data:  # percent inputs for Nova data must sum to 100%; use 0% if you want to
exclude a data grouping
    agents: 20
    business-and-finance: 20
    scientific: 20
    code: 20
    factual-and-news: 20
    longform-text: 0
    health-and-medicine: 0
    humanities-and-education: 0
    legal: 0
    math: 0
    additional-languages: 0
    social-and-personal-interest: 0
    entertainment: 0
    reasoning: 0
```

```

    other: 0
    tables: 0
    customer_data: # percent input of customer data. 100 = use only customer data, 0 =
use only the nova_data mix above
    percent: 25

```

Il existe ici deux catégories de données de premier niveau :

- **nova_data** : Il s'agit du véritable mélange de données, qui est subdivisé en encore plus de catégories. Il est impératif qu'ils se résument à 100 %
 - Vous trouverez ci-dessous une liste complète de ces catégories, y compris le nombre de jetons.
- **customer_data** : Il s'agit de vos données d'entraînement référencées dans la clé `data_s3_path` en haut de votre fichier yaml. Le pourcentage indiqué ici détermine le pourcentage obtenu pour `nova_data`. Par exemple, dans les pourcentages de sélections ci-dessus, pendant la formation, nous utiliserons 25 % de `customer_data` et 75 % de `nova_data`, dont 15 % seront des agents, 15 % seront des agents, 15 % seront des scientifiques business-and-finance, 15 % seront du code et 15 % seront factual-and-news

Conseil : exécutez à nouveau `pip install -e.` et vous êtes prêt à soumettre votre travail !

Nous allons remplacer quelques paramètres ici pour utiliser le mixage des données :

```

hyperpod start-job \
-n kubeflow \
--recipe fine-tuning/nova/forg/nova_2_0/nova_lite/SFT/
nova_lite_2_0_p5x8_gpu_sft_with_datamix \
--override-parameters '{
"instance_type": "ml.p5.48xlarge",
"recipes.run.name": "nova-sft-datamixing",
"container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-HP-
SFT-V2-Datamix",
"recipes.run.data_s3_path": "s3://sft-data/sft_train_data.jsonl",
"recipes.run.validation_data_s3_path": "s3://sft-data/sft_val_data.jsonl",
"recipes.run.output_s3_path": "s3://sft-data/output/
}'

```

Votre sortie doit contenir le nom de la tâche comme suit :

```
# MY Desktop # % hyperpod start-job \
-n kubeflow \
--recipe training/nova/forged/nova_2_0/nova_lite/sft/
nova_lite_2_0_p5x8_gpu_pretrain_with_datamix \
--override-parameters '{
  "instance_type": "ml.p5.48xlarge",
  "recipes.run.name": "nova-sft-datamixing",
  "container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-fine-tune-repo:SM-HP-
SFT-V2-Datamix",
  "recipes.run.data_s3_path": "s3://sft-data/sft_train_data.jsonl",
  "recipes.run.validation_data_s3_path": "s3://sft-data/sft_val_data.jsonl",
  "recipes.run.output_s3_path": "s3://sft-data/output/
}'
```

Le résultat serait le suivant :

```
Final command: python3 /local/home/my/Downloads/sagemaker-hyperpod-cli/
src/hyperpod_cli/sagemaker_hyperpod_recipes/main.py recipes=training/nova/
nova_micro_p5x8_gpu_pretrain cluster_type=k8s cluster=k8s base_results_dir=/local/home/
niphaded/Downloads/sagemaker-hyperpod-cli/results cluster.pullPolicy="IfNotPresent"
cluster.restartPolicy="OnFailure" cluster.namespace="kubeflow"
instance_type="p5d.48xlarge" container="900867814919.dkr.ecr.us-east-1.amazonaws.com/
nova-fine-tune-repo:sft-datamix-rig-final"
Prepared output directory at /local/home/my/Downloads/sagemaker-hyperpod-cli/results/
my-sft-run-wzdyn/k8s_templates
Found credentials in shared credentials file: ~/.aws/credentials
Helm script created at /local/home/my/Downloads/sagemaker-hyperpod-cli/results/my-sft-
run-wzdyn/niphaded-sft-run-wzdyn_launch.sh
Running Helm script: /local/home/my/Downloads/sagemaker-hyperpod-cli/results/my-sft-
run-wzdyn/niphaded-sft-run-wzdyn_launch.sh

NAME: my-sft-run-wzdyn
LAST DEPLOYED: Tue Aug 26 16:21:06 2025
NAMESPACE: kubeflow
STATUS: deployed
REVISION: 1
TEST SUITE: None
Launcher successfully generated: /local/home/my/Downloads/sagemaker-hyperpod-cli/src/
hyperpod_cli/sagemaker_hyperpod_recipes/launcher/nova/k8s_templates/SFT

{
```

```
"Console URL": "https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/cluster-management/hyperpod-eks-ga-0703"
}
```

Vous pouvez consulter le statut de votre tâche en utilisant `hyperpod list-pods -n kubeflow --job-name my-sft-run-wzdyn`

```
hyperpod list-pods -n kubeflow --job-name my-sft-run-wzdyn
{
  "pods": [
    {
      "PodName": "my-sft-run-wzdyn-master-0",
      "Namespace": "kubeflow",
      "Status": "Pending",
      "CreationTime": "2025-08-26 16:21:06+00:00"
    },
    {
      "PodName": "my-sft-run-wzdyn-worker-0",
      "Namespace": "kubeflow",
      "Status": "Pending",
      "CreationTime": "2025-08-26 16:21:06+00:00"
    }
  ]
}
```

ou utilisez directement la commande `kubectl` pour les trouver.

Par exemple,

```
kubectl get pods -o wide -w -n kubeflow | (head -n1 ; grep my-sft-run)
```

NAME	READY	STATUS
RESTARTS AGE IP NODE		NOMINATED NODE
READINESS GATES		
my-sft-run-5suc8-master-0	0/1	Completed 0
3h23m 172.31.32.132 hyperpod-i-00b3d8a1bf25714e4	<none>	<none>
my-sft-run-5suc8-worker-0	0/1	Completed 0
3h23m 172.31.44.196 hyperpod-i-0aa7ccfc2bd26b2a0	<none>	<none>
my-sft-run-5suc8-worker-1	0/1	Completed 0
3h23m 172.31.46.84 hyperpod-i-026df6406a7b7e55c	<none>	<none>

my-sft-run-5suc8-worker-2	0/1	Completed	0
3h23m 172.31.28.68 hyperpod-i-0802e850f903f28f1	<none>		<none>

Conseil de pro : veuillez à toujours utiliser l'indicateur `-o wide`, car le nœud EKS sur lequel la tâche s'exécute vous aidera à retrouver vos journaux encore plus rapidement dans l'interface utilisateur AWS

Comment surveiller Job

Vous pouvez consulter vos journaux de trois manières :

a) Utilisation CloudWatch

Vos journaux sont disponibles sur votre compte Amazon Web Services qui contient le cluster Hyperpod sous CloudWatch. Pour les consulter dans votre navigateur, rendez-vous sur la CloudWatch page d'accueil de votre compte et recherchez le nom de votre cluster. Par exemple, si votre cluster était appelé, `my-hyperpod-rig` le groupe de journaux aurait le préfixe suivant :

- Groupe de journaux : `/aws/sagemaker/Clusters/my-hyperpod-rig/ {UUID}`
- Une fois que vous êtes dans le groupe de journaux, vous pouvez trouver votre journal spécifique à l'aide de l'ID d'instance du nœud, tel que `hyperpod-i-00b3d8a1bf25714e4`.
 - `i-00b3d8a1bf25714e4` représente ici le nom de la machine adaptée aux hyperpodes sur laquelle se déroule votre tâche de formation. Rappelez-vous comment, dans la commande précédente `kubectl get pods -o wide -w -n kubeflow | (head -n1 ; grep my-cpt-run)` output, nous avons capturé une colonne appelée `NODE`.
 - Le nœud « principal » exécuté était dans ce cas exécuté sur `hyperpod-i-00b3d8a1bf25714e4`. Nous allons donc utiliser cette chaîne pour sélectionner le groupe de journaux à afficher. Sélectionnez celui qui dit `SagemakerHyperPodTrainingJob /rig-group/ [NODE]`

Vos journaux devraient ressembler à ceci :

b) Utilisation d' CloudWatch Insights

Si vous avez le nom de votre poste à portée de main et que vous ne souhaitez pas suivre toutes les étapes ci-dessus, vous pouvez simplement rechercher tous les journaux sous `/aws/sagemaker/Clusters/my-hyperpod-rig/ {UUID}` pour trouver le journal individuel.

CPT


```
fields @timestamp, @message, @logStream, @log
| filter @message like /(?!i)Starting CPT Job/
| sort @timestamp desc
| limit 100
```

Pour terminer la tâche, remplacez Starting SFT Job par SFT Job completed

Ensuite, vous pouvez cliquer sur les résultats et choisir celui qui indique « Epoch 0 », car ce sera votre nœud principal.

C) Utilisation de l'interface de ligne de commande aws

Vous pouvez choisir de suivre vos journaux à l'aide du. Avant de le faire, veuillez vérifier votre version d'aws cli à l'aide de `aws --version`. Il est également recommandé d'utiliser ce script utilitaire qui facilite le suivi des journaux en direct sur votre terminal.

pour V1 :

```
aws logs get-log-events \
--log-group-name /aws/sagemaker/YourLogGroupName \
--log-stream-name YourLogStream \
--start-from-head | jq -r '.events[].message'
```

pour V2 :

```
aws logs tail /aws/sagemaker/YourLogGroupName \
--log-stream-name YourLogStream \
--since 10m \
--follow
```

D) Configurer ML Flow :

Vous pouvez suivre les métriques via MLFlow.

Création d'une MLflow application

Utilisation de l'interface utilisateur de Studio : si vous créez une tâche de formation via l'interface utilisateur de Studio, une MLflow application par défaut est créée automatiquement et sélectionnée par défaut dans les options avancées.

Utilisation de la CLI : si vous utilisez la CLI, vous devez créer une MLflow application et la transmettre comme entrée à la demande d'API de la tâche de formation.

```
mlflow_app_name="<enter your MLflow app name>"
role_arn="<enter your role ARN>"
bucket_name="<enter your bucket name>"
region="<enter your region>"

mlflow_app_arn=$(aws sagemaker create-mlflow-app \
  --name $mlflow_app_name \
  --artifact-store-uri "s3://$bucket_name" \
  --role-arn $role_arn \
  --region $region)
```

Accédez à l' MLflow application

À l'aide de la CLI : créez une URL présignée pour accéder à l'interface utilisateur de l' MLflow application :

```
aws sagemaker create-presigned-mlflow-app-url \
  --arn $mlflow_app_arn \
  --region $region \
  --output text
```

Une fois ML Flow configuré, vous pouvez transmettre l'URI dans votre recette ou utiliser le override au démarrage de la tâche. Vous trouverez un exemple de la façon de procéder dans le fichier README.

Comment évaluer votre modèle après SFT ?

Conditions préalables

- Vérifiez l'URI S3 depuis le fichier manifest.json de votre tâche de formation (pour les modèles entraînés)
- Ensemble de données d'évaluation chargé dans S3 dans le bon format
- Chemin de sortie S3 pour les résultats d'évaluation

Benchmarks prêts à l'emploi : utilisez des benchmarks prêts à l'emploi pour valider les performances sur des tâches générales. Pour plus de détails, cliquez [ici](#).

Apportez vos propres données :

Vous pouvez également fournir vos données personnalisées en les formatant dans le format ci-dessous, puis en utilisant les conteneurs mentionnés ci-dessous pour obtenir des résultats d'inférence ainsi que des probabilités logarithmiques pour les étalonnages si nécessaire.

Créez jsonl par tâche avec la structure suivante :

```
{
  "metadata": "{key:4, category:'apple'}",
  "system": "arithmetic-patterns, please answer the following with no other words: ",
  "query": "What is the next number in this series? 1, 2, 4, 8, 16, ?",
  "response": "32"
}
```

Les résultats générés pendant la phase d'inférence du travail d'évaluation auront la structure suivante :

```
{
  "prompt": "[{'role': 'system', 'content': 'arithmetic-patterns, please answer the following with no other words: '}, {'role': 'user', 'content': 'What is the next number in this series? 1, 2, 4, 8, 16, ?'}]",
  "inference": "['32']",
  "gold": "32",
  "metadata": "{key:4, category:'apple'}"
}
```

Descriptions des champs :

- **prompt** : entrée formatée envoyée au modèle
- **inférence** : réponse générée par le modèle
- **gold** : réponse correcte attendue de l'ensemble de données d'entrée, champ de réponse de l'entrée
- **métadonnées** : métadonnées facultatives transmises depuis l'entrée

Préparer la configuration d'évaluation

Commande pour lancer la tâche d'évaluation. Utilisez « `--override-parameters` » pour modifier n'importe quelle entrée de la recette.

```
hyperpod start-job -n kubeflow \
  --recipe evaluation/nova/nova_micro_p5_48xl_bring_your_own_dataset_eval \
  --override-parameters '{
    "instance_type": "p5.48xlarge",
    "container": "708977205387.dkr.ecr.us-east-1.amazonaws.com/nova-evaluation-repo:SM-HP-Eval-latest",
    "recipes.run.name": "<your-eval-job-name>",
    "recipes.run.model_name_or_path": "<checkpoint-s3-uri>",
    "recipes.run.output_s3_path": "s3://<your-bucket>/eval-results/",
    "recipes.run.data_s3_path": "s3://<your-bucket>/eval-data.jsonl"
  }'
```

Lancez votre job d'évaluation

Commandes de lancement de tâches pour différentes recettes avec les images correspondantes.

Commandes de lancement des tâches d'évaluation

Modèle	Technique	Sous-catégorie	URI de l'image	Commande
Nova 2.0	Évaluation	Éval	708977205387.dkr.ecr.us-east-1.amazonaws.com/:SM-HP-Eval-Latest nova-evaluation-repo	hyperpod start-job -n kubeflow \ --recipe -Evaluation-Repo:SM-HP-Eval-Latest »} 'evaluation/nova/nova_2_0/nova_lite/nova_lite_2_0_p5_48xl_gpu_ft_eval \ --override-parameters '{ "instance_type": "ml.p5.48

Modèle	Technique	Sous-catégorie	URI de l'image	Commande
				xlarge", "container": "70897720 5387.dkr.ecr.us- east-1.amaz onaws.com/nova

Leçons apprises et conseils

- La qualité de l'ensemble de données SFT est essentielle. Vous devez tout mettre en œuvre pour filtrer les données de faible qualité. Si vous disposez d'un petit sous-ensemble de données d'une qualité exceptionnelle, à la fois en termes de complexité et de précision, vous pouvez envisager de le placer vers la fin de l'entraînement pour aider le modèle à mieux converger.
- Nous utilisons à la fois des ensembles de données textuels et multimodaux (MM) pour le mixage des données. Nos expériences avec des jeux de données textuels montrent que l'ajout de la catégorie propriétaire « reasoning-instruction-following » de Nova améliore considérablement les performances par rapport aux benchmarks génériques. Nous vous recommandons d'inclure cette catégorie dans votre stratégie de mixage des données si vous vous intéressez à un benchmark générique régressé une fois que vous avez utilisé SFT avec vos ensembles de données.
- Pour les ensembles de données MM, nos expériences indiquent que l'intégration de plus de 20 % de catégories vidéo dans le mix est bénéfique pour maintenir les performances de référence génériques.
- De plus, le SFT avec mixage de données est assez sensible au taux d'apprentissage. Nos résultats suggèrent donc de l'ajuster avec le taux d'apprentissage par défaut, c'est-à-dire 1e-5 pour LoRa et 5e-6 pour FR.
- Enfin, il existe un compromis entre latence et performances si vous mélangez des ensembles de données propriétaires de Nova. Nos résultats suggèrent donc de mélanger 50 % au maximum pour obtenir un bon équilibre.

Apprentissage par renforcement

Note

Une documentation détaillée est fournie une fois que vous êtes abonné

Nova Forge fournit des capacités avancées d'apprentissage par renforcement avec la possibilité d'utiliser des fonctions de récompense à distance dans votre propre environnement. Les clients peuvent choisir d'intégrer leur propre point de terminaison pour exécuter la validation afin d'obtenir des commentaires immédiats dans le monde réel, ou même d'utiliser leur propre orchestrateur pour coordonner les évaluations agentiques en plusieurs étapes dans votre environnement.

Apportez votre propre orchestrateur pour des évaluations agentiques en plusieurs étapes

Pour les utilisateurs de Forge qui ont besoin de conversations à plusieurs tours ou de fonctions de récompense dépassant les 15 minutes, Nova Forge propose des fonctionnalités BYOO (Bring Your Own Orchestration). Cela vous permet de coordonner des évaluations agentiques à plusieurs tours dans votre environnement (par exemple, en utilisant des outils de chimie pour évaluer les conceptions moléculaires, ou des simulations robotiques qui récompensent l'exécution efficace des tâches et pénalisent les collisions).

Rubriques

- [Présentation de l'architecture](#)
- [Configuration et exécution](#)

Présentation de l'architecture

L'architecture BYOO fournit un contrôle total sur le processus de déploiement et de génération grâce à une infrastructure gérée par le client.

VPC d'entraînement :

- **Déploiement** : coordonne la formation en déléguant la génération du déploiement à l'infrastructure du client
- **Trainer** : effectue des mises à jour du poids des modèles en fonction des déploiements reçus

VPC du client (tel qu'ECS sur EC2) :

- Proxy Lambda : reçoit les demandes de déploiement et assure la coordination avec l'infrastructure du client
- Réponse au déploiement SQS : file d'attente pour le renvoi des déploiements terminés vers l'infrastructure de formation
- Generate Request SQS : file d'attente pour les demandes de génération de modèles
- Generate Response SQS : file d'attente pour les réponses de génération de modèles
- Conteneur client : implémente une logique d'orchestration personnalisée (peut utiliser le kit de démarrage fourni)
- DynamoDB : stocke et récupère l'état tout au long du processus d'orchestration

Flux de travail :

1. Le déploiement délègue la génération du déploiement à Proxy Lambda
2. Proxy Lambda envoie une demande d'API de déploiement pour générer une requête SQS
3. Le conteneur client traite les demandes, gère les interactions à plusieurs tours et appelle les fonctions de récompense
4. Le conteneur stocke et extrait l'état de DynamoDB selon les besoins
5. Container transmet les réponses de déploiement à Rollout Response SQS
6. Rollout envoie les déploiements terminés à Trainer pour qu'il mette à jour le poids

Configuration et exécution

Pour des instructions de configuration détaillées, des configurations de recettes, des formats de demande et de réponse, ainsi que des exemples d'environnement, reportez-vous à la documentation confidentielle fournie aux abonnés de Nova Forge. Pour obtenir les documents de Nova Forge, suivez les étapes ci-dessous :

```
aws s3 cp s3://nova-forge-c7363-206080352451-us-east-1/v1/ ./ --recursive
```

Une fois les ressources téléchargées, vous trouverez toute la documentation dans le docs dossier.

Boîte à outils d'IA responsable et modération du contenu

Boîte à outils pour une IA responsable

Nova Forge fournit une boîte à outils d'IA responsable qui inclut des données de formation et d'évaluation pour aligner les modèles sur les directives d'Amazon Nova en matière d'IA responsable pendant le processus de formation, ainsi que des contrôles d'exécution pour modérer les réponses des modèles lors de l'inférence.

Données de formation — Des cas et des scénarios mettant l'accent sur les principes de l'IA responsable, les considérations de sécurité et le déploiement responsable des technologies sont disponibles pour le mixage des données afin d'aligner les modèles de manière responsable pendant la formation préalable continue.

Évaluations — Les évaluations testant la capacité du modèle à détecter et à rejeter le contenu inapproprié, préjudiciable ou incorrect sont disponibles en tant que tâche de référence pour déterminer le delta entre les performances du modèle de base et les performances du modèle personnalisé.

Contrôles d'exécution : par défaut, le moteur d'exécution d'Amazon Nova contrôle les réponses modérées du modèle lors de l'inférence. Pour modifier ces contrôles d'exécution, demandez les paramètres de modération de contenu personnalisables d'Amazon Nova en contactant un responsable de AWS compte.

La sécurité est une responsabilité partagée entre AWS et ses utilisateurs. La modification du modèle de base ou le recours à une formation préalable continue pour améliorer les performances dans un cas d'utilisation spécifique peuvent avoir un impact sur la sécurité, l'équité et d'autres propriétés du nouveau modèle. Une méthode d'adaptation robuste minimise les modifications apportées à la sécurité, à l'équité et aux autres protections intégrées aux modèles de base tout en minimisant l'impact sur les performances du modèle pour les tâches pour lesquelles le modèle n'a pas été personnalisé. End-to-end tests d'applications sur des ensembles de données représentatifs de cas d'utilisation sont nécessaires pour déterminer si les résultats des tests répondent aux attentes spécifiques en matière de sécurité, d'équité et d'autres propriétés, ainsi que d'efficacité globale. Pour plus d'informations, consultez le guide d'utilisation responsable de l'IA d'Amazon Web Services, la politique d'Amazon Web Services en matière d'intelligence artificielle responsable, la politique d'utilisation acceptable d'Amazon Web Services et les conditions de service d'Amazon Web Services.

Utilisation responsable

La mise en place de mesures de sûreté, de sécurité et de confiance à l'aide de modèles d'IA est une responsabilité partagée entre nos clients AWS et nos clients. Notre objectif est d'aligner nos modèles sur la [Politique d'utilisation acceptable d'AWS](#) et d'atténuer les résultats indésirables tout en offrant une expérience client agréable. Notre approche de l'IA responsable (RAI) s'articule autour de nos [dimensions fondamentales de l'IA responsable](#), qui sont présentées dans la liste suivante. Pour chacune de ces dimensions, nous avons élaboré des lignes directrices qui régissent notre prise de décision tout au long du cycle de vie du développement du modèle. Ce cycle de vie englobe toutes les étapes, de la collecte initiale des données et de l'entraînement à la mise en œuvre de mesures d'atténuation d'exécution après le déploiement.

- Équité : prise en compte des impacts sur différents groupes de parties prenantes
- Explicabilité : compréhension et évaluation des résultats du système
- Confidentialité et sécurité : obtention, utilisation et protection appropriées des données et des modèles
- Sécurité : empêcher les résultats préjudiciables et les utilisations abusives
- Contrôlabilité : disposer de mécanismes permettant de surveiller et de diriger le comportement du système d'IA
- Véracité et robustesse : obtenir des résultats corrects du système, même en cas d'entrées inattendues ou hostiles
- Gouvernance : intégrer les bonnes pratiques dans la chaîne d'approvisionnement de l'IA, y compris les fournisseurs et les dépoyeurs
- Transparence : permettre aux parties prenantes de faire des choix éclairés concernant leur engagement avec un système d'IA

Consignes

Les directives que nous utilisons pour orienter le développement de nos modèles comprennent, sans s'y limiter, la modération des contenus qui glorifient, facilitent ou encouragent les éléments suivants :

- Participation à des activités dangereuses, automutilation ou utilisation de substances dangereuses.
- Utilisation, abus ou commerce de substances contrôlées, de tabac ou d'alcool.
- Violence physique ou sanglante.

- Maltraitance d'enfants ou matériel pédopornographique.
- Maltraitance ou trafic d'animaux.
- Désinformation qui présente des individus ou des groupes comme responsables d'une tromperie délibérée, qui sape la crédibilité d'une institution auprès du grand public ou qui met en danger la santé ou les moyens de subsistance des personnes.
- Logiciels malveillants, contenus malveillants ou tout contenu qui facilite la cybercriminalité.
- Manque de respect, discrimination ou stéréotypes à l'égard d'un individu ou d'un groupe.
- Insultes, blasphèmes, gestes obscènes, langage sexuellement explicite, pornographie, symboles haineux ou groupes haineux.
- Nudité intégrale en dehors d'un contexte scientifique, éducatif ou de référence.
- Préjugés à l'encontre d'un groupe sur la base d'une caractéristique démographique.

Recommandations

Pertinence d'utilisation : les résultats des modèles d'IA étant probabilistes, Amazon Nova peut produire des contenus inexacts ou inappropriés. Les clients doivent évaluer l'exactitude et la pertinence des résultats pour leur cas d'utilisation, en particulier s'ils sont directement présentés aux utilisateurs finaux. En outre, si Amazon Nova est utilisé dans des flux de travail clients qui aboutissent à des décisions corrélatives, les clients doivent évaluer les risques potentiels liés à leur cas d'utilisation et mettre en œuvre une supervision humaine appropriée, des tests et d'autres mesures de protection spécifiques au cas d'utilisation pour atténuer ces risques.

Optimisation des invites : en cas de modération par Amazon Nova, envisagez d'examiner les invites utilisées par rapport aux directives ci-dessus. **Optimisations rapides :** en cas de modération de la part d'Amazon Nova, pensez à examiner les instructions utilisées conformément aux directives ci-dessus. L'optimisation des invites afin de réduire le risque de générer des résultats indésirables est la stratégie recommandée pour produire les résultats attendus à l'aide des modèles Amazon Nova. Soyez attentif aux cas où les entrées sont contrôlées par les utilisateurs, y compris le contenu en pixels qui pourrait avoir un impact sur les performances du modèle. Veuillez consulter la section consacrée aux directives relatives aux invites dans ce guide d'utilisation pour plus de détails.

Confidentialité : Amazon Nova est disponible en Amazon Bedrock. Amazon Bedrock est un service géré qui ne stocke ni ne passe en revue les instructions des clients ou les réponses rapides des clients. Les instructions et les réponses ne sont jamais partagées entre les clients ou avec les partenaires. Amazon Bedrock AWS n'utilise pas les entrées ou les sorties générées par le Amazon

Bedrock service pour entraîner Amazon Bedrock des modèles, y compris Amazon Nova. Consultez la section [50.3](#) des conditions de AWS service et la [FAQ sur la confidentialité AWS des données](#) pour plus d'informations. Pour obtenir des informations de confidentialité spécifiques au service, consultez la section Confidentialité et sécurité de la [Amazon Bedrock FAQs](#) documentation. Amazon Nova prend des mesures pour éviter de compléter des invites qui pourraient être interprétées comme des demandes d'informations privées. Si un utilisateur craint que ses informations privées aient été incluses dans une complétion Amazon Nova, il doit nous contacter [ici](#).

Sécurité : tous les Amazon Bedrock modèles, y compris Amazon Nova, sont dotés d'une sécurité d'entreprise qui permet aux clients de créer des applications d'IA génératives conformes aux normes de sécurité et de conformité des données communes, notamment le RGPD et l'HIPAA. Les clients peuvent l'utiliser AWS PrivateLink pour établir une connectivité privée entre Amazon Nova personnalisé et les réseaux sur site sans exposer le trafic client à Internet. Les données des clients sont toujours cryptées en transit et au repos et les clients peuvent utiliser leurs propres clés pour chiffrer les données, par exemple en utilisant AWS Key Management Service. Les clients peuvent l'utiliser Gestion des identités et des accès AWS pour contrôler en toute sécurité l'accès aux Amazon Bedrock ressources, y compris Amazon Nova personnalisé. Amazon Bedrock Offre également des fonctionnalités complètes de surveillance et de journalisation qui peuvent répondre aux exigences de gouvernance et d'audit des clients. Par exemple, Amazon CloudWatch peut aider à suivre les statistiques d'utilisation requises à des fins d'audit, à surveiller l'activité des API et à résoudre les problèmes, car Amazon Nova est intégré à d'autres AWS systèmes. AWS CloudTrail Les clients peuvent également choisir de stocker les métadonnées, les instructions et les complétions dans leur propre compartiment chiffré Amazon Simple Storage Service (Amazon S3).

Propriété intellectuelle : AWS [offre une couverture d'indemnisation illimitée en matière de propriété intellectuelle \(IP\) pour les sorties des modèles Amazon Nova généralement disponibles \(voir la section 50.10 des conditions de service\)](#). Cela signifie que les clients sont protégés contre les réclamations de tiers alléguant une violation ou un détournement de la propriété intellectuelle (y compris les réclamations relatives aux droits d'auteur) par les résultats générés par ces modèles Amazon Nova. En outre, notre indemnisation standard en matière de propriété intellectuelle pour l'utilisation des services protège les clients contre les réclamations de tiers alléguant une violation de la propriété intellectuelle (y compris les réclamations relatives aux droits d'auteur) par les services (y compris les modèles Amazon Nova) et les données utilisées pour les entraîner.

Quotas pour Amazon Nova

Vous Compte AWS disposez de quotas par défaut, anciennement appelés limites, pour Amazon Bedrock. Pour afficher les quotas de service pour Amazon Bedrock, procédez de l'une des manières suivantes :

- Suivez les étapes décrites dans [Affichage des quotas de service](#) et sélectionnez Amazon Bedrock comme service.
- Reportez-vous aux [points de terminaison et aux quotas d'Amazon Bedrock](#) dans le. Références générales AWS

Note

Si un quota est marqué comme non ajustable par le biais de Service Quotas, vous pouvez soumettre une demande d'augmentation de [limite par le biais du formulaire d'augmentation](#) des limites pour qu'une augmentation soit prise en compte.

Pour maintenir les performances du service et garantir une utilisation appropriée d'Amazon Bedrock, les quotas par défaut attribués à un compte peuvent être mis à jour en fonction de facteurs régionaux, de l'historique des paiements, de l'utilisation frauduleuse ou de l'and/or approbation d'une demande d'augmentation de quota.

Demander une augmentation des quotas Amazon Nova

Vous pouvez demander une augmentation du quota de votre compte en suivant les étapes ci-dessous :

- Si un quota est marqué Oui dans la colonne Réglable des [points de terminaison Amazon Bedrock et des quotas](#) dans le Références générales AWS, vous pouvez l'ajuster en suivant les étapes décrites dans la section [Demander une augmentation de quota](#) dans le guide de l'utilisateur des quotas de service dans le guide de l'utilisateur [des quotas de service](#).
- Les quotas d'invocation du modèle à la demande dans les points de terminaison [Amazon Bedrock et les quotas dans les points de terminaison](#) Références générales AWS ne sont pas ajustables via

Service Quotas. Contactez votre Compte AWS responsable pour qu'une augmentation soit prise en compte.

 Note

En raison de la forte demande, la priorité sera accordée aux clients dont le trafic consomme le quota qui leur est alloué. Votre demande peut être refusée si vous ne remplissez pas cette condition.

- Vous pouvez soumettre une demande via le [formulaire d'augmentation de limite](#) afin que votre demande d'augmentation soit prise en compte, même si un quota est marqué comme Non dans la colonne Ajustable dans [Points de terminaison et quotas Amazon Bedrock](#) dans la Références générales AWS,

Surveillance d'Amazon Nova

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'Amazon Nova et de vos autres AWS solutions. AWS fournit les outils de surveillance suivants pour surveiller Amazon Nova, signaler un problème et prendre des mesures automatiques le cas échéant :

Amazon Nova étant disponible via Amazon Bedrock, veuillez suivre toutes les instructions de surveillance indiquées dans [Surveiller l'intégrité et les performances d'Amazon Bedrock](#).

Sécurité dans Amazon Nova

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de [AWS conformité Programmes](#) de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon Nova, consultez la section [AWS Services concernés par programme de conformitéAWS](#) .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, notamment la sensibilité de vos données, les exigences de votre entreprise et les lois et réglementations applicables.

Amazon Nova étant disponible via Amazon Bedrock, veuillez suivre toutes les consignes de sécurité figurant dans la section [Sécurité dans Amazon Bedrock](#).

Créez des ressources Amazon Nova avec AWS CloudFormation

Lorsque vous déployez des modèles Amazon Nova via Amazon Bedrock with CloudFormation, suivez toutes les CloudFormation instructions de la section [Ressources Create Amazon Bedrock with. CloudFormation](#)

Pour plus d'informations, notamment des exemples de modèles JSON et YAML pour, consultez le document [YOUR-SERVICE-RESOURCE-TOPNODE-IN-CLOUDFORMATION YOUR-RESOURCE-PLURAL](#), modifiez ce lien dans le fichier local-phrases.ent dans le guide de l'utilisateur.AWS CloudFormation

En savoir plus sur CloudFormation

Pour en savoir plus CloudFormation, consultez les ressources suivantes :

- [AWS CloudFormation](#)
- [AWS CloudFormation Guide de l'utilisateur](#)
- [Référence de l'API CloudFormation](#)
- [AWS CloudFormation Guide de l'utilisateur de l'interface de ligne de commande](#)

Historique du document pour le guide du développeur Amazon Nova

Le tableau suivant décrit les versions de documentation pour Amazon Nova 2.0.

Modification	Description	Date
Version initiale d'Amazon Nova 2.0	Première publication du guide du développeur Amazon Nova 2.0 avec de nouveaux modèles : Nova 2 Lite, Amazon Nova Sonic et. Ce guide présente des fonctionnalités multimodales améliorées, des fonctionnalités de raisonnement et des performances améliorées à tous les niveaux du modèle.	2 décembre 2025

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.