Implementation Guide

# Innovation Sandbox on AWS

# Innovation Sandbox on AWS: Implementation Guide

# Table of Contents

**Solution overview ..... 1**
Features and benefits ..... 2
Use cases ..... 3
Concepts and definitions ..... 4
**Architecture overview ..... 8**
Architecture diagram ..... 8
AWS Well-Architected design considerations ..... 10
Operational excellence ..... 10
Security ..... 10
Reliability ..... 11
Performance efficiency ..... 12
Cost optimization ..... 12
Sustainability ..... 13
**Architecture details ..... 14**
Authentication mechanism ..... 14
CloudFormation stacks ..... 14
Stack dependencies ..... 15
AccountPool Organizational Units ..... 18
Web application ..... 21
Event infrastructure ..... 22
Event schemas reference ..... 24
Account Cleaner components ..... 36
Account lifecycle ..... 37
AWS services in this solution ..... 39
**Plan your deployment ..... 42**
Supported AWS Regions ..... 42
Choosing the deployment accounts ..... 45
Accounts ..... 45
Home Region ..... 46
Configuring an external identity provider (Optional) ..... 47
Group Management ..... 47
User Management ..... 47
Attribute mapping examples ..... 48
Control Tower managed organizations ..... 49

Creating sandbox accounts ..... 49
Cost ..... 50
Example cost table ..... 50
Security ..... 51
IAM roles ..... 52
IAM Identity Center and SAML authentication ..... 52
AWS Key Management Service ..... 52
AWS WAF ..... 52
Amazon CloudFront ..... 53
Amazon DynamoDB ..... 53
AWS Lambda ..... 54
Amazon CloudWatch Alarms ..... 54
Log retention and monitoring ..... 54
AWS CloudTrail ..... 54
Amazon S3 security features ..... 54
Custom client security considerations ..... 55
Quotas ..... 55
Quotas for AWS services in this solution ..... 55
AWS CloudFormation quotas ..... 56
AWS Lambda quotas ..... 56
AWS CodeBuild quotas ..... 56
**Deploy the solution ..... 57**
Deployment process overview ..... 57
Prerequisites ..... 57
AWS CloudFormation templates ..... 58
AccountPool stack ..... 58
IDC stack ..... 59
Data stack ..... 59
Compute stack ..... 59
Launch the stacks ..... 60
Step 1: Deploy the AccountPool stack ..... 60
Step 2: Deploy the IDC stack ..... 62
Step 3: Deploy the Data stack ..... 67
Step 4: Deploy the Compute stack ..... 68
**Post-deployment configuration tasks ..... 71**
Configure IAM Identity Center ..... 71

Create a SAML 2.0 application ..... 71
Map application attributes ..... 72
Assign groups to your application ..... 73
Assign users to groups ..... 74
Configure the web application ..... 75
Update configuration using AWS AppConfig ..... 76
Update values in AWS Secrets Manager ..... 78
**Using the web UI ..... 80**
Logging into the web UI ..... 84
Administrator Guide ..... 85
Adding new accounts to the account pool ..... 85
Managing existing accounts ..... 88
Registering and managing blueprints ..... 89
Viewing or modifying Innovation Sandbox settings ..... 101
Manager Guide ..... 107
Creating lease templates ..... 107
Updating lease templates ..... 113
Deleting lease templates ..... 115
Assigning leases to users ..... 116
Approving and rejecting leases ..... 117
Choosing the right budget and duration configuration ..... 117
Managing leases ..... 120
Viewing your lease costs ..... 122
Accessing user accounts for troubleshooting ..... 123
User Guide ..... 123
Requesting a new account lease ..... 124
Logging in to an account ..... 125
Requesting a lease extension ..... 125
**Monitoring the solution ..... 126**
Overview ..... 126
Amazon CloudWatch Application Insights ..... 126
Cloudwatch log queries ..... 127
AWS X-Ray ..... 128
**Update the solution ..... 129**
Overview ..... 129
Update via CloudFormation templates ..... 129

Update via source code (Git) ..... 130
**Troubleshooting ..... 132**
Application sign in error ..... 132
Incorrect global configuration ..... 132
Authentication failed, Invalid document signature ..... 132
Authentication failed, SAML assertion audience mismatch ..... 133
Investigating accounts in Quarantine state ..... 133
Resolving cleanup failures ..... 134
Viewing a specific Lease history ..... 134
Viewing a specific User history ..... 135
403 Permissions error ..... 135
Unexpected server errors ..... 135
Lease assignment issues ..... 136
User not found error ..... 136
Unfreezing a lease ..... 136
Group Cost reporting issues ..... 137
Re-running the group cost reporting function ..... 137
Running cost reporting for a previous month ..... 137
Blueprint deployment issues ..... 138
StackSet not found error ..... 138
Blueprint deployment timeout ..... 139
Concurrent deployment failure (OperationInProgressException) ..... 139
Blueprint permission errors ..... 140
StackSet not appearing in the Innovation Sandbox application ..... 141
Contact AWS Support ..... 142
Create a case ..... 142
How can we help? ..... 142
Additional information ..... 142
Help us resolve your case faster ..... 142
Solve now or contact us ..... 143
**Uninstall the solution ..... 144**
End leases and eject accounts ..... 144
Enable maintenance mode ..... 144
End all Active and Frozen leases ..... 145
Eject accounts ..... 145
Move accounts out of the Organizational Unit ..... 146

Uninstall solution stacks ..... 147
Using the AWS Management Console ..... 147
Resources retained after deletion ..... 148
Delete the custom application in IAM Identity Center ..... 149
**Developer guide ..... 151**
Source code ..... 151
List of solution API endpoints ..... 151
**Reference ..... 152**
Data collection ..... 152
Contributors ..... 153
**Revisions ..... 155**
**Notices ..... 156**
Terms of Use for Admins ..... 156

# Create temporary sandbox environments with configurable security and spend monitoring controls

Publication date: *May 2025. For updates, refer to CHANGELOG.md file in the GitHub repository.*

The Innovation Sandbox on AWS solution allows cloud administrators to set up and recycle temporary sandbox environments by automating the implementation of security and governance policies, spend management mechanisms, and account recycling preferences through a web user interface (UI). Using the solution, customers can empower their teams to experiment, learn, and innovate with AWS services in production-isolated AWS accounts that are recycled after use.

> **Note**
>
> The solution does not create any new, or close existing AWS accounts; it only allows you to manage existing AWS accounts for sandbox experiments, and recycles accounts to promote reuse.

The solution automates the setup of a sandbox Organizational Unit (OU) structure that comes preconfigured with best practices for workload isolation, by automatically deploying a standard set of policies, guardrails, and controls across sandbox accounts. The solution:

1. Enables cost optimization by sending alerts and initiating automated actions when spend reaches budget threshold limits.
2. Enables account recycling by providing the ability to use accounts for a predefined duration or spend threshold, and cleaning up the account at the end of its sandbox use.
3. Limits and controls excessively expensive, or sensitive actions within sandbox accounts.

This implementation guide provides an overview of the Innovation Sandbox on AWS solution, its reference architecture and components, considerations for planning the deployment, and configuration steps for deploying the solution to the AWS Cloud. It is intended for solution architects, DevOps engineers, AWS account administrators, and cloud professionals who want to implement Innovation Sandbox on AWS in their environment.

Use this navigation table to find answers to these common questions:

| If you want to ... | Read ... |
| --- | --- |
| Know the cost for running this solution.<br><br>The average estimated cost for running this solution in the US East (N. Virginia) Region is **USD $65.25 per month**. | Cost |
| Understand the security considerations for this solution. | Security |
| Know how to plan for quotas for this solution. | Quotas |
| Know which AWS Regions support this solution. | Supported AWS Regions |
| View the instructions to automatically deploy the infrastructure resources (the "stacks") for this solution. | Deploy the solution |

# Features and benefits

**Automate the creation of a sandbox environment**

Transforms the sandbox setup process with automated deployment of organizational unit (OU) structures that adhere to best practices for workload isolation.

**Accelerate environment setup with blueprints**

Deploys pre-configured infrastructure to sandbox accounts automatically through CloudFormation StackSets, allowing users to get started with ready-to-use resources.

**Enhanced operational efficiency**

Reduces administrative overhead by implementing standardized policies, guardrails, and security controls across sandbox accounts automatically, ensuring consistent governance while saving valuable cloud administration time.

**Establish cost governance**

Maintains better cost control and takes necessary action to reduce unnecessary spend; monitors spend patterns, sends automated alerts at defined thresholds, and restricts access or clean up resources when budget thresholds are approached.

**Gain visibility into sandbox usage**

Centrally monitors all sandbox accounts, tracks sandbox usage metrics, and makes informed decisions with detailed visibility of sandbox environments using the web User Interface (UI).

**Recycle and reuse AWS accounts**

Efficiently reuses AWS accounts using a clean-up mechanism that is automatically initiated when the spend or time period reaches predefined limits. This systematic approach ensures that sandbox environments are recycled and ready for new experiments, while minimizing administrative overheads.

# Use cases

**Development and innovation experiments**

Developers who want to build a proof of concept on new AWS services, or run innovation experiments and prove the business value, before moving to a CI/CD pipeline.

**Pre-configured development environments**

Development teams provide standardized development environments with pre-installed tools, frameworks, and configurations through blueprints, ensuring consistency across team members and reducing manual environment setup.

**Train and test GenAI models**

Machine learning engineers and data scientists who want to train, test, fine-tune, and establish reinforcement learning on foundation models to improve the model's accuracy and reduce bias.

**Test environment**

Quality Assurance/Test engineers who want a disposable and isolated cloud environment to run integration tests, regression tests, reproduce bugs, and test API changes, before pushing tested code to a CI/CD pipeline.

**Higher education training labs**

Educators, such as Head of Department, professors, and teachers at universities who want to train students by creating and managing disposable cloud environments (classroom labs, exams, and more).

**Research and Development (R&D)**

Educators at universities, colleges, and high schools or R&D teams at enterprises who want to run cloud research experiments in a controlled environment to verify their hypotheses.

**Employee onboarding and training**

Training leads at enterprises who want to provide a secure and short-lived cloud environment to deliver hands-on learning, workshops or onboarding experiences for employees.

**Hackathons**

IT teams (at healthcare companies, investment firms, and other enterprises) who want to run hackathons in AWS accounts owned by them, so that they can host sensitive and proprietary data.

**Demo environments**

Engineers and solution architects at enterprises who want an environment to run demos.

**Software vendors**

Companies that sell software and want to stand up time or budget limited demos of their software solutions and make them available to their customers to try.

# Concepts and definitions

**Sandbox environment**

A controlled, isolated environment where teams can experiment with AWS services without impacting production systems. It provides a safe space for learning, testing, and innovation.

**Organizational Unit (OU)**

A grouping of AWS accounts that allows you to organize accounts into a hierarchy and apply policies. This solution creates dedicated OUs for active and recycled sandbox accounts.

**Service Control Policies (SCPs)**

Policy documents that specify the maximum available permissions for accounts within an AWS Organization. They help enforce security boundaries and service restrictions across sandbox accounts.

**Lease**

A lease is a temporary allocation of an AWS account to a user for a specified budget or lease duration to run innovation experiments. Leases can be created through two methods: user-initiated requests (traditional self-service model) or manager-initiated assignments (direct assignment model).

**Lease template**

A lease template provides the ability to define conditions that govern the use of the account - such as approval for a user to use a given account, budget and threshold actions, lease duration and threshold actions, and template visibility controls. Admins and managers can create lease templates with public or private visibility settings, and sandbox users can request new sandbox leases by choosing from available public templates. Private templates are only accessible to administrators and managers for direct lease assignment purposes.

**Blueprint**

A registered CloudFormation StackSet that deploys pre-configured infrastructure to sandbox accounts when you provision a lease. Blueprints enable administrators to provide users with ready-to-use environments containing standardized resources, tools, and configurations, eliminating manual setup time and ensuring consistency across sandbox accounts.

**Template visibility**

A configuration setting that controls whether a lease template appears in the general template listing for user requests. Public templates are visible to all users for self-service lease requests, while private templates are only accessible to administrators and managers for direct lease assignment purposes.

**Budget threshold**

A predefined customer-defined spending limit that triggers specific actions when reached. The solution uses thresholds to send alerts, stop resources, and prevent new resource creation.

**Account recycling**

The process of cleaning up and reusing sandbox accounts when they reach customer-defined limits. This helps optimize account management and reduce administrative overhead.

**AWS Nuke**

AWS-nuke is an open-source tool designed for the purpose of cleaning up and deleting AWS resources in a systematic and automated way.

**Guardrails**

Preventive or detective controls that protect your AWS environment. They help ensure sandbox accounts maintain security, compliance, and operational standards.

**Hub Account**

A centralized AWS account that hosts the sandbox resources and configuration, and orchestrates actions across sandbox accounts.

**Cost Report Group**

A configurable identifier used to categorize and aggregate sandbox costs for organizational reporting and chargeback purposes. Cost report groups enable administrators to attribute sandbox usage costs to specific departments, teams, or cost centers, facilitating accurate cost allocation and budget management across the organization.

**Permission set**

A collection of administrator-defined policies that AWS IAM Identity Center uses to determine a user's access permissions to AWS accounts.

**Resource controls**

Mechanisms that manage AWS resource lifecycle, including creation, modification, and termination based on defined policies and budget thresholds.

**Least privilege access**

A security principle where users and resources are granted the minimum permissions necessary to perform their tasks. The solution enforces this through automated policy deployment.

**Note**

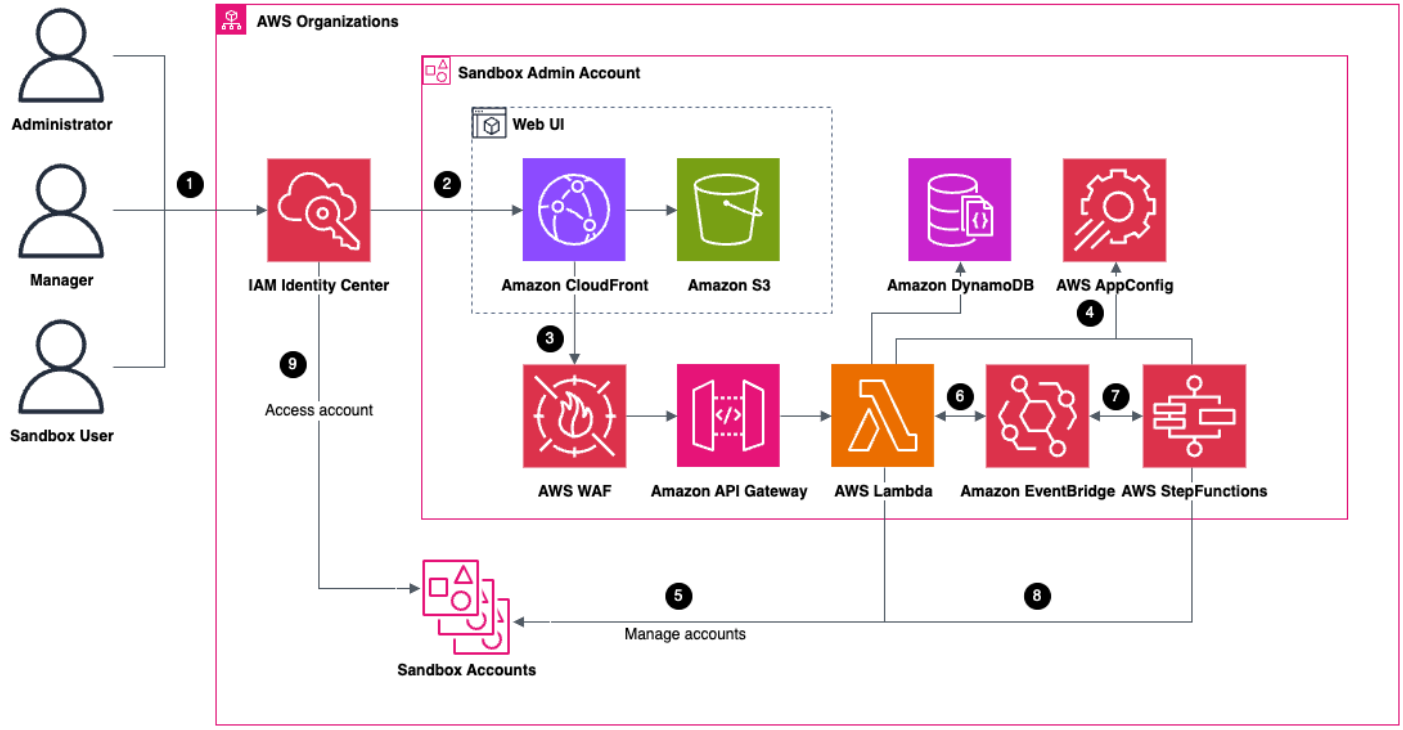
For a general reference of AWS terms, see the AWS Glossary.

# Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

## Architecture diagram

Deploying this solution with the default parameters builds the following environment in your AWS account.




### Innovation Sandbox on AWS architecture

The high-level process flow for the solution components deployed with the AWS CloudFormation templates is as follows:

1. Users access the solution (SAML2.0 application) using AWS IAM Identity Center authentication. You can configure IAM Identity Center to use its own internal user store, or integrate it with an external identity provider such as Okta or Microsoft Entra ID.

2. The web User Interface (UI) is hosted in an Amazon CloudFront distribution. It uses an Amazon Simple Storage Service (Amazon S3) bucket to host and serve the web frontend, including the HTML pages, CSS stylesheets, and the JavaScript code.
3. The web UI calls Amazon API Gateway REST API resources (resource, method, model) to fetch and mutate the solution data. AWS Lambda functions authorize the requests using role-based access, based on identities assigned by solution administrators to user groups in IAM Identity Center. AWS WAF protects the Amazon API Gateway from common exploits and bots that can affect availability, compromise security, or consume excessive resources.
4. AWS Lambda functions handle the API requests by reading, and writing status and configuration data to an Amazon DynamoDB table. These Lambda functions also fetch global configurations from AWS AppConfig to manage solution parameters including lease preferences, account cleanup setting, customer worded "terms of service", and auth configurations.
5. AWS Lambda functions manage the lifecycle of accounts using the AWS Organizations API, and move them between organizational units (OUs) based on the account status. Service control policies (SCPs) attached to OUs prevent sensitive, expensive, or difficult to clean up services and resources from being used by sandbox users.
6. The solution's backend includes an event-based architecture built on Amazon EventBridge for routing events. The solution monitors sandbox account leases using AWS Lambda for breaches in configured lease budget and duration thresholds and creates events that produce email notifications via Amazon Simple Email Service and invoke Lambda functions that are responsible for the management of lease and account lifecycle.
7. Accounts going through the onboarding process or leases being terminated will invoke the account cleanup AWS Step Functions, which is responsible for recycling the accounts back into the account pool, ready for reuse.
8. AWS Step Functions run an AWS CodeBuild project responible for deleting resources in the account. AWS Lambda functions monitor active account leases and issues actions such as moving an AWS account between Organizational Units (OUs), attaching/detaching an IAM Identity Center permission set to the account giving user access, or initiating the cleanup of an AWS account which deletes all user-created resources using AWS Nuke.
   - If the clean up process is successful, the account is moved to the **available** account pool, or
   - If some resources cannot be deleted, the account is moved to a **quarantine** state, for manual investigation and remediation.
9. Users access assigned sandbox accounts via IAM Identity Center access portal console, or programmtically using credentials. The solution provides a link in the web UI to directly access the AWS account with Single Sign-On (SSO).

# AWS Well-Architected design considerations

This solution uses the best practices from the AWS Well-Architected Framework which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework benefit this solution.

## Operational excellence

We architected this solution using the principles and best practices of the operational excellence pillar to benefit this solution.

The Innovation Sandbox on AWS solution implements operational excellence through:

- **Automated operations**
  - Automates sandbox environment setup, configuration, and infrastructure deployment.
  - Deploys standardized policies and guardrails across accounts.
  - Reduces manual intervention in account lifecycle management.
- **Event response**
  - Implements automated responses to budget thresholds.
  - Provides a Cloudwatch Application Insights dashboard for monitoring and alerts.
  - Enables quick identification and resolution of issues, using predefined CloudWatch Log Insight queries and X-Ray traces.
- **Standard definitions**
  - Creates consistent Organizational Unit (OU) structure across implementations.
  - Establishes standardized security policies.
  - Maintains uniform budget control mechanisms.

## Security

We architected this solution using principles and best practices of the security pillar to benefit this solution.

The solution implements comprehensive security controls:

- **Identity and access management**
  - Integrates with AWS IAM Identity Center for centralized access control.
  - Automatically implements least privilege permissions.
  - Enforces role-based access across sandbox accounts.
- **Network security**
  - Isolates sandbox environments from production systems.
  - Restricts access to internal networks.
  - Controls network traffic through automated WAF policies.
- **Data protection**
  - Prevents access to sensitive corporate resources.
  - Implements service control policies for data protection.
  - Maintains isolation between sandbox environments.

## Reliability

We architected this solution using principles and best practices of the reliability pillar to benefit this solution.

The solution ensures reliability through:

- **Distributed design**
  - Implements multi-account architecture.
  - Uses AWS Organizations for management.
  - Maintains separation of concerns across accounts.
- **Automated recovery**
  - Implements automated resource management.
  - Enables account recycling and clean-up.
  - Provides consistent environment configuration.
- **Change management**
  - Automates policy deployment.
  - Maintains consistent controls across accounts.
  - Enables standardized environment updates.

## Performance efficiency

We architected this solution using principles and best practices of the performance efficiency pillar to benefit this solution.

The solution maintains performance efficiency by:

- **Resource selection**
  - Allows administrators to specify approved services and Regions.
  - Enables right-sizing of resources for sandbox environments.
  - Provides flexibility in resource configuration.
- **Automated environment provisioning**
  - Blueprint deployment automates infrastructure setup for sandbox accounts.
  - Allows users to get started with pre-configured resources.
- **Monitoring**
  - Creates a centralized CloudWatch Application Insights dashboard.
  - Tracks resource utilization across accounts.
  - Enables performance optimization through metrics.

## Cost optimization

We architected this solution using principles and best practices of the cost optimization pillar to benefit this solution.

The solution optimizes costs through multiple mechanisms:

- **Resource management**
  - Automatically manage accounts (clean-up or freeze) when budget thresholds are reached.
  - Freeze: Prevents creation of new resources at budget limits.
  - Clean-up: Enables account recycling to optimize usage.
- **Cost controls**
  - Implements multi-tier budget threshold monitoring.
  - Provides visibility into spending across accounts.
  - Reduces monthly cost overruns through automated controls.

> **Note**
>
> Identification of cost/budget overrun per account is best effort due to Cost Explorer service limitation.

- **Resource lifecycle**
  - Manages resource termination based on budget limits and/or lease duration.
  - Enables account reuse through automated clean-up.
  - Optimizes account utilization through recycling.

## Sustainability

We architected this solution using principles and best practices of the sustainability pillar to benefit this solution.

- The solution uses managed and serverless services where possible to minimize the environmental impact.

# Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

Customers can deploy the Innovation Sandbox on AWS solution into their accounts either:

- From CloudFormation templates served from public S3 buckets managed by AWS, or
- Building the solution from the open source code available on GitHub.

## Authentication mechanism

Innovation Sandbox on AWS uses the Single Sign On service from AWS IAM Identity center for authentication. The authentication is done via a browser with redirects using the SAML 2.0 protocol.

You must register the web application with the custom SAML 2.0 application, and update the application with the SAML 2.0 application details. Users log in to the web UI from the Applications tab on the AWS access portal, or programmatically using credentials provided by the IAM Identity Center.

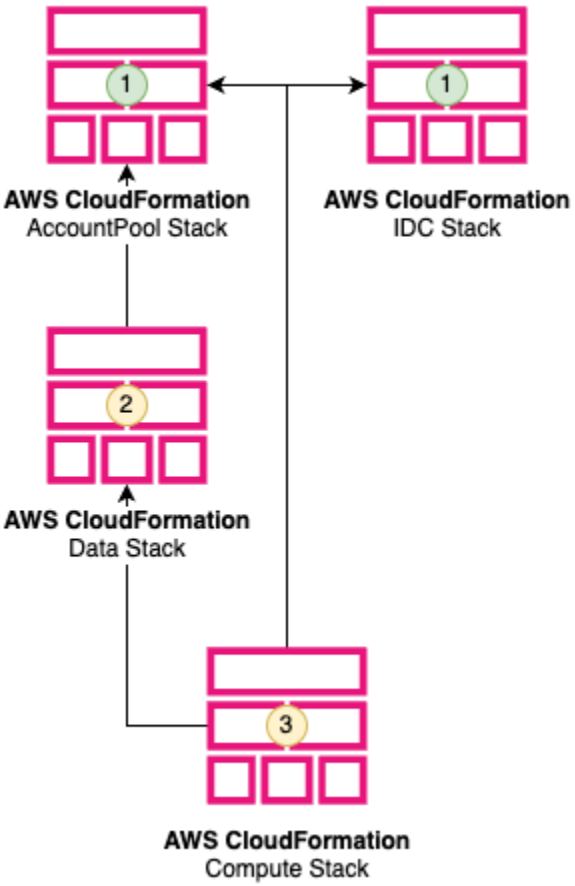
## CloudFormation stacks

The solution is composed of several CloudFormation templates that are deployed into these accounts:

- The AWS Organizations management account
- The account containing the organizations' AWS IAM Identity Center instance
- A designated Hub account for the solution to be deployed into.

Sandbox accounts have a CloudFormation StackSets instance deployed in the account, managed by AWS Organizations.

# Stack dependencies

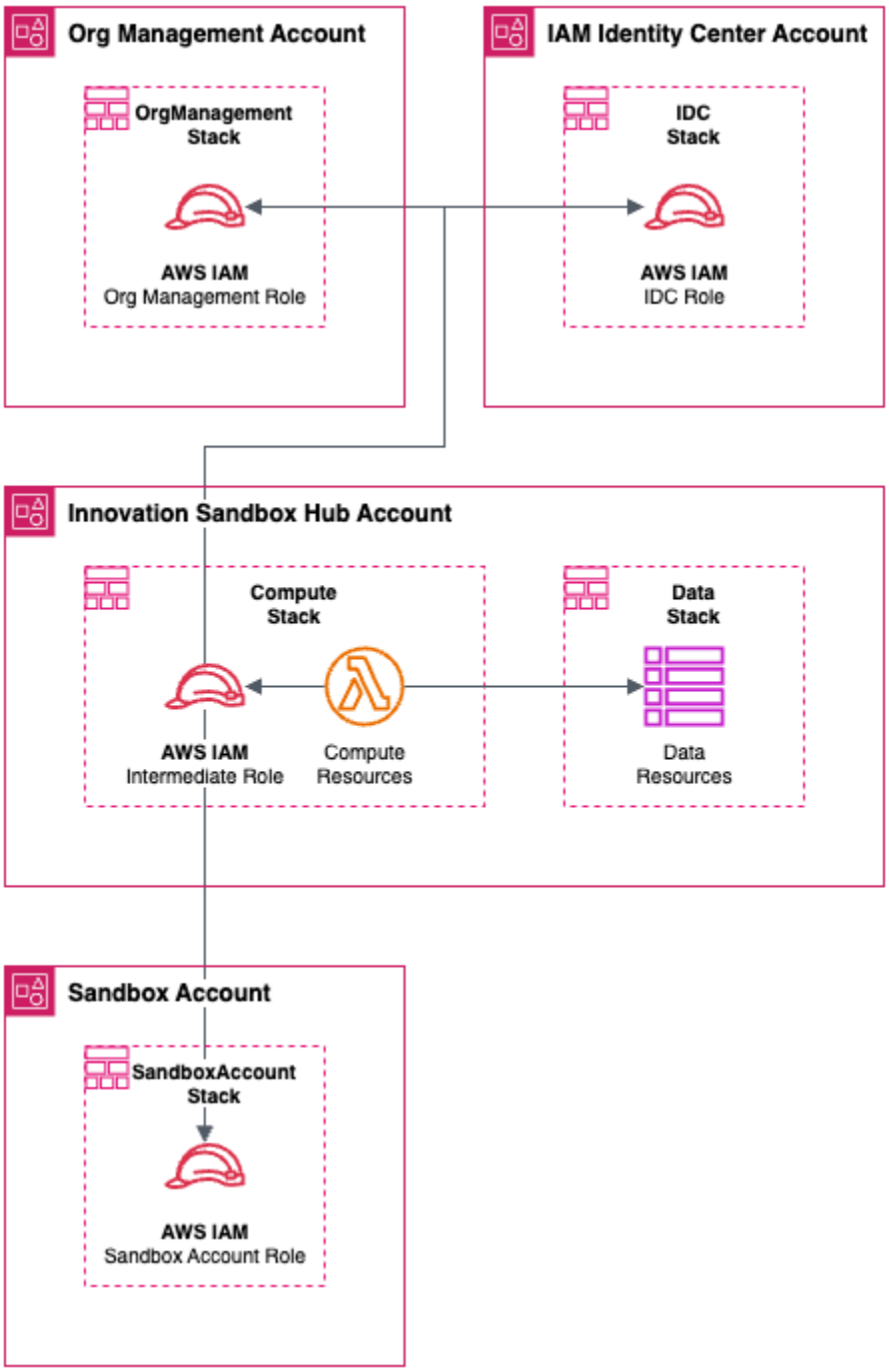



### Solution stack dependencies

This diagram shows the order in which the stacks should be deployed. Some stacks depend on resources created by another stack to successfully deploy, so it is critical to deploy the stacks in the correct order for a successful deployment.

1. Deploy the **AccountPool** and **IDC** stacks as they do not depend on each other. The **SandboxAccount** stack is created as a service-managed StackSet resource. The StackSet instances will be deployed in the SandboxAccount when the accounts are moved into Sandbox OU. This occurs for all accounts that are moved into the AccountPool organizational unit.
2. Deploy the **Data** stack, as it requires the **AccountPool** stack.
3. Deploy the **Compute** stack as it requires all other stacks to be deployed before it is deployed.

As part of the deployment, the solution deploys these CloudFormation stacks.



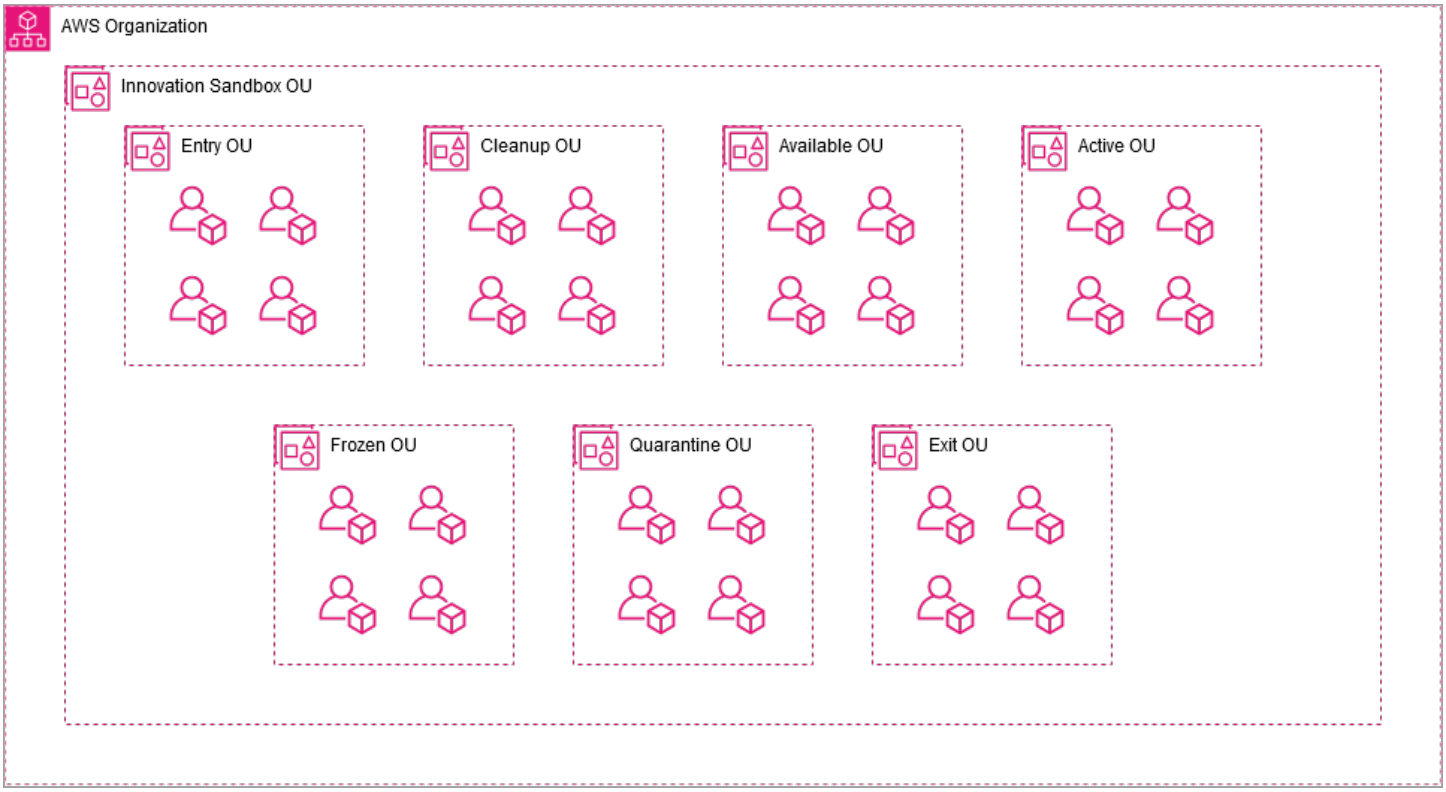

### ISB CloudFormation stacks

- The **AccountPool** stack, deployed into the AWS Organizations management account, is used to manage the lifecycle on sandbox accounts controlled by the solution. This stack contains three major types of resources:

  - **Organizational Units** (OUs) representing the lifecycle of the sandbox accounts (available, active, frozen, clean-up, quarantine, entry, exit),
  - **Service Control Policies** (SCPs) that limit what actions can be taken in accounts in each OU and by what principals, and
  - An **IAM** role. The permissions on this role are least privileged to only allow read actions from Cost Explorer, read actions on the account pool OUs, and move account actions on the account pool OUs. The trust policy on the role only allows for a single Intermediate IAM role from the Compute stack to assume into it.
  - In addition to account management, a **Spoke** role in this account is used to read the account cost data from the Cost Explorer service API.
- The **IAM Identity Center** (IDC) stack deployed into the AWS Account containing the organizations AWS IAM Identity Center instance, is used to manage the solution web UI and sandbox account access. This stack initializes user groups and corresponding permission sets in the instance that administrators can manually add users to. The IDC stack also contains an IAM Role. The permissions on this role are least privileged to only allow the actions required by the solution. The trust policy on the role only allows for a single Intermediate IAM role from the Compute stack to assume into it.
- The **Data** stack is deployed into the account containing the core of the solution (Hub account). It contains Amazon DynamoDB tables containing the stateful data of the solution, including records for sandbox accounts, lease templates, and leases. This stack also contains the AWS AppConfig hosted configurations for the solution's global configurations, nuke configurations, and reporting configurations.
- The **Compute** stack is deployed into the account containing the core of the solution (Hub account). This stack contains all of the stateless (compute) resources used by the solution. The stack contains these two parts:
  - The **web application** is composed of the Amazon CloudFront distribution that serves the static assets of the UI, Amazon API Gateway REST API, a custom AWS Lambda Request Authorizer, and AWS Lambda function handlers for each of the API resources.
  - The **event infrastructure** is composed of producers and consumers on an Amazon EventBridge Event Bus. Produces consist of AWS Lambda functions running on a EventBridge schedule, API initiated events, and Account Cleaner step function initiated events. Event consumers consist of Lambda functions handling account lifecycle, sending emails via SES, and the account cleaner StepFunction. The account cleaner Step Function (state machines) consists of AWS Lambda functions and a CodeBuild project that uses a custom public image containing the AWS Nuke binary.

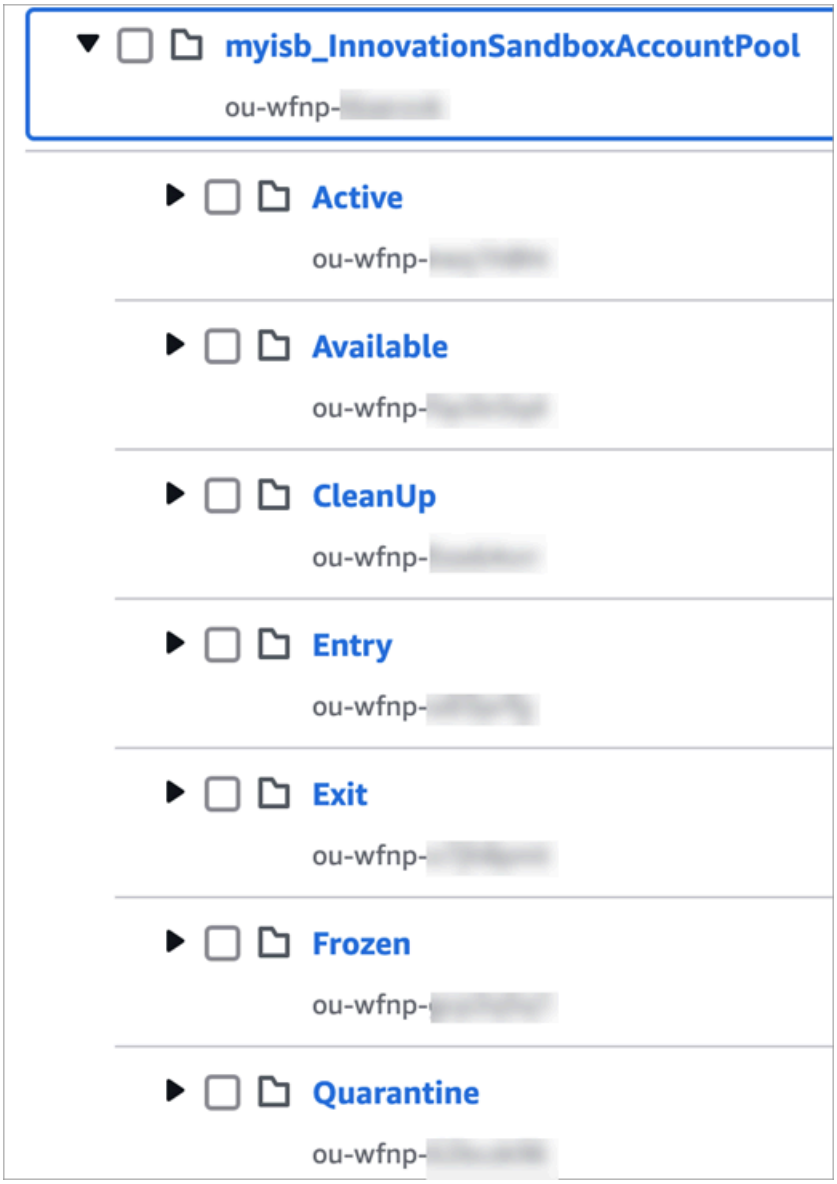
> **Note**
>
> The **SandboxAccount** stack is automatically configured as a service-managed CloudFormation StackSet resource in the AccountPool stack using the Sandbox OU as the deployment target. The stack contains a single spoke role that is assumed into by compute resources in the compute stack to run the account clean-up job. SCPs established in the AccountPool stack protect the deletion of the role as well as the StackSet instance.

# AccountPool Organizational Units




### AccountPool OUs

The AccountPool Organizational Unit (OU) structure defines the sandbox account lifecycle through the solution, and allows for Service Control Policies (SCPs) to restrict actions within the accounts at different phases of the account lifecycle.



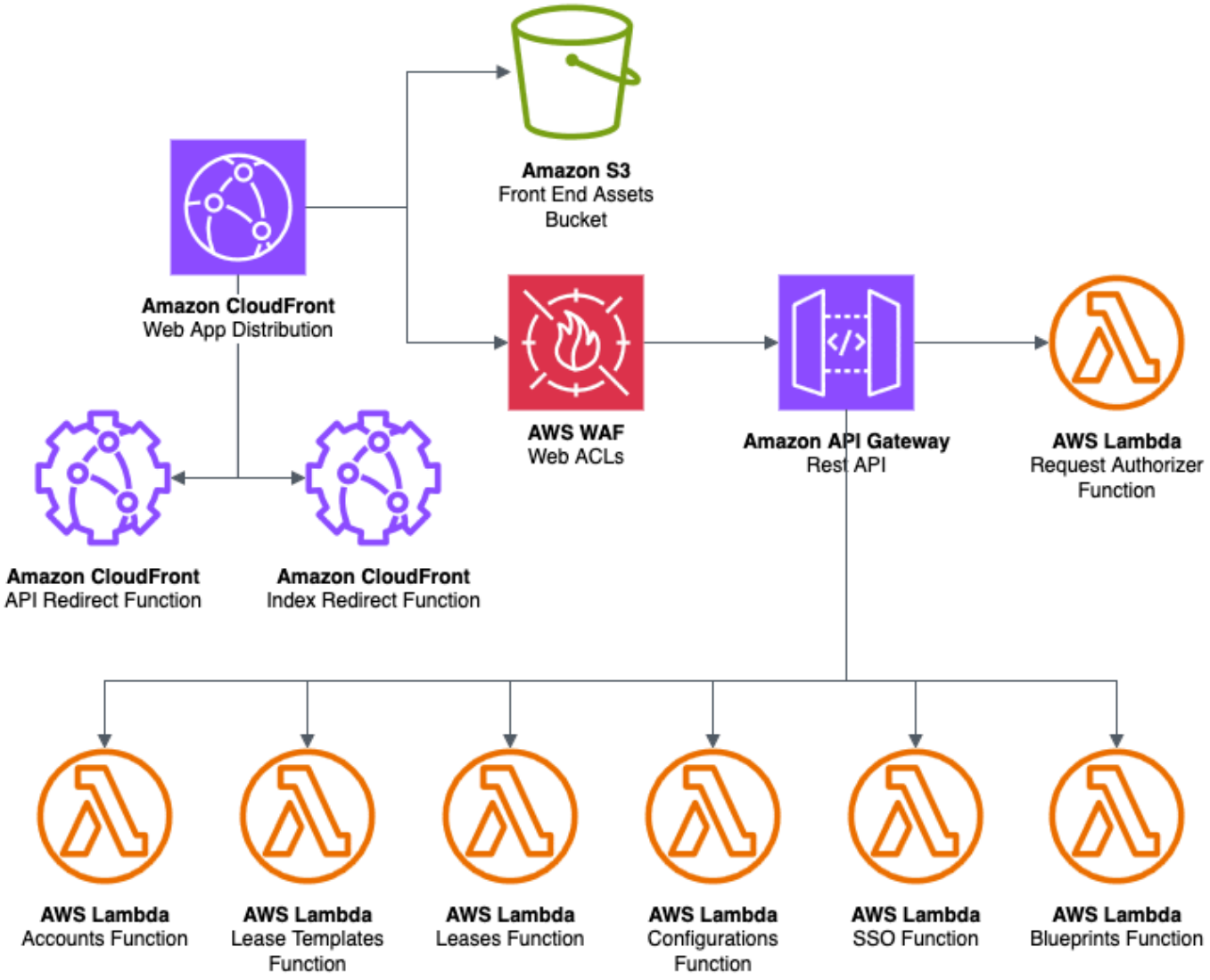

**AccountPool OUs list**

The following OUs are created when the solution is deployed:

| Organizational Unit (OU) | Description |
| --- | --- |
| *<nameSpace>* _InnovationSandboxAccountPool | Parent OU that all other solution OUs are contained in. |

| Organizational Unit (OU) | Description |
| --- | --- |
| Active | Sandbox accounts that are associated with an active lease (claimed). |
| Available | Sandbox accounts that are available for lease (unclaimed). |
| CleanUp | Sandbox accounts that are currently in clean-up. |
| Entry | Staging OU for accounts that are to be registered with the solution. |
| Exit | Staging OU for accounts that have been ejected from the solution. |
| Frozen | Sandbox accounts where the users access has been revoked, but administr ators still have access in order to review resources within the account. |
| Quarantine | Sandbox accounts that have failed the clean-up process due to an undeletab le resource or was detected as solution state drift and needs manual remediation from an administrator. |

# Web application



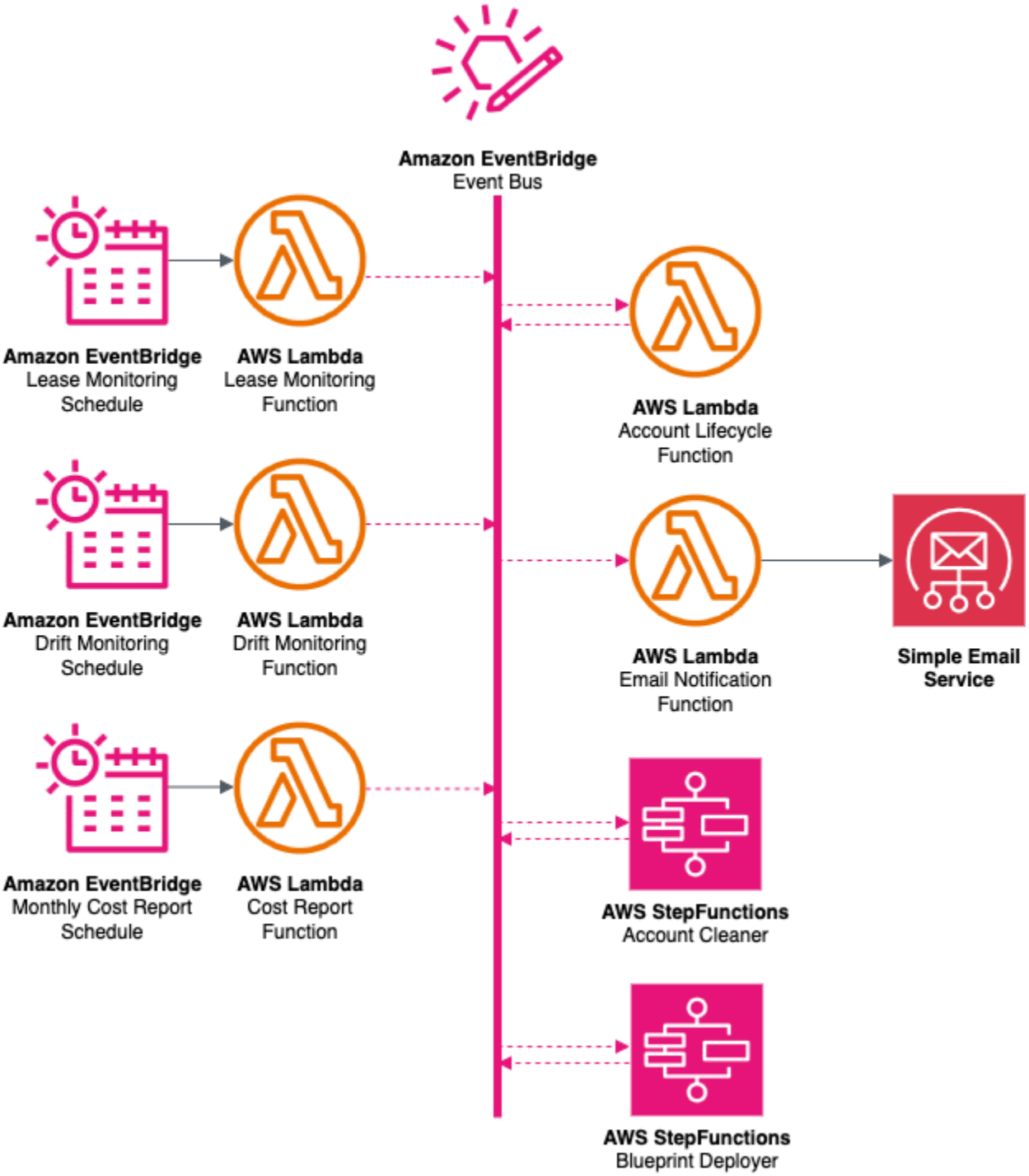

### Web UI and storage management components

The web app infrastructure consists of an Amazon CloudFront distribution with an Amazon Simple Storage Service (Amazon S3) bucket origin for hosting the static assets for the web UI, and an API origin.

The API uses an AWS WAF protected Amazon API Gateway REST API, with proxy AWS Lambda function integrations for each of the API resources (leases, lease-templates, accounts, configurations, users, auth, blueprints) and an AWS Lambda authorizer function for authorizing API

requests. Each of the Lambda function integrations is responsible for interacting with one or more of the underlying data stores in the Data stack.

# Event infrastructure




**Event infrastructure**

The event infrastructure performs the solution's monitoring actions and responds to the events produced. The resources in the diagram can be categorized into event **producers** and **consumers**.

Producers:

- The **lease monitoring** Lambda checks all active leases against their configured terms and generates any alerts or lifecycle events that occur. To check the cost usage, the Lambda function assumes the intermediate role in the Hub account and the Spoke role in the Org management account to retrieve cost and usage data from the AWS Cost Explorer service. This Lambda runs hourly via an EventBridge schedule.
- The **drift monitoring** Lambda checks for situations where the internal state of the solution (DynamoDB) does not match the actual location of an account within the org (for instance an account may be in an active state, but is located in the cleanup OU). In this scenario, an event is produced to move the account into quarantine for manual investigation by an administrator. This lambda runs every 6 hours via an EventBridge schedule.
- The **cost report** Lambda generates monthly cost reports aggregated by cost report groups configured in lease templates. This function queries DynamoDB for all lease records from the previous month, retrieves cost and usage data from AWS Cost Explorer, and generates CSV reports with cost breakdowns by cost report group. The reports are stored in a designated S3 bucket and administrators are notified via email when reports are available. This Lambda is run at the beginning of every month.
- The **account lifecycle** Lambda produces an event to start the account clean-up process after an account has been successfully moved to the CleanUp OU, and produces blueprint deployment request events when a lease with an attached blueprint is approved.
- The **account cleaner** AWS Step Functions (state machine) emits an event once an account has been processed.
- The **blueprint deployment** Step Functions (state machine) emits success or failure events once a blueprint deployment process has been completed.

Consumers:

- The **account lifecycle** function receives events from the account cleaner, blueprint deployment, and lease monitoring Lambdas to move accounts from one lifecycle state to another. This process involves moving the account between OUs, updating the state in DynamoDB, and revoking/granting user access to an account within the IAM Identity Center.

- The **account cleaner** function receives events indicating that an account should be processed for clean-up.
- The **email notification** Lambda receives events from all producers and sends human readable emails to the appropriate users, managers, and administrators for the event. This Lambda uses Amazon Simple Email Service (SES) for these notifications. It is expected that customers will configure this outside of the solution deployment.

## Event schemas reference

Innovation Sandbox on AWS publishes events to Amazon EventBridge to enable integration with external systems. This section provides a comprehensive reference of all event schemas that customers can use to build integrations with their own systems.

All events are published to the Innovation Sandbox EventBridge custom bus with a consistent structure and follow the AWS EventBridge event pattern.

### Event structure

All Innovation Sandbox events follow this standard EventBridge structure:

```
{
  "version": "0",
  "id": "event-id",
  "detail-type": "EventDetailType",
  "source": "innovation-sandbox",
  "account": "123456789012",
  "time": "2024-01-15T14:30:25Z",
  "region": "us-east-1",
  "detail": {
    // Event-specific payload (documented below)
  }
}
```

### Event categories

Events are organized into the following categories:

- **Lease Lifecycle Events** - Lease creation, approval, denial, and termination
- **Lease Monitoring Events** - Budget and duration threshold alerts

- **Account Management Events** - Account cleanup and drift detection
- **Cost Reporting Events** - Cost report generation and failures
- **Blueprint Deployment Events** - Blueprint deployment requests, successes, and failures

## Lease lifecycle events

### LeaseRequested

Published when a user submits a new lease request.

**Detail Type:** `LeaseRequested`

**Schema:**

```
{
  "detail-type": "LeaseRequested",
  "detail": {
    "leaseId": {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
    "comments": "Need AWS environment for ML experimentation",
    "userEmail": "developer@company.com",
    "requiresManualApproval": true
  }
}
```

### LeaseApproved

Published when a lease request is approved and the user gains access to their sandbox account.

**Detail Type:** `LeaseApproved`

**Schema:**

```
{
  "detail-type": "LeaseApproved",
  "detail": {
    "leaseId": "550e8400-e29b-41d4-a716-446655440000",
    "approvedBy": "manager@company.com",
    "userEmail": "developer@company.com"
  }
```

```
}
```

### LeaseDenied

Published when a lease request is denied by an approver.

**Detail Type:** `LeaseDenied`

**Schema:**

```
{
  "detail-type": "LeaseDenied",
  "detail": {
    "leaseId": "550e8400-e29b-41d4-a716-446655440000",
    "deniedBy": "manager@company.com",
    "userEmail": "developer@company.com"
  }
}
```

### LeaseTerminated

Published when a lease is terminated for any reason. The reason field provides specific details about why the lease was terminated.

**Detail Type:** `LeaseTerminated`

**Schema:**

```
{
  "detail-type": "LeaseTerminated",
  "detail": {
    "leaseId": {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
    "accountId": "123456789012",
    "reason": {
      "type": "BudgetExceeded",
      "budget": 100.00,
      "totalSpend": 105.50
    }
  }
}
```

**Reason Types:** `Expired`, `BudgetExceeded`, `ManuallyTerminated`, `AccountQuarantined`, `Ejected`

## Lease monitoring events

### LeaseBudgetThresholdAlert

Published when a lease's spending reaches a configured budget threshold percentage.

**Detail Type:** `LeaseBudgetThresholdAlert`

**Schema:**

```
{
  "detail-type": "LeaseBudgetThresholdAlert",
  "detail": {
    "leaseId": {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
    "accountId": "123456789012",
    "budget": 100.00,
    "totalSpend": 80.00,
    "budgetThresholdTriggered": 0.8,
    "actionRequested": "ALERT"
  }
}
```

### LeaseBudgetExceeded

Published when a lease's spending exceeds the configured budget limit.

**Detail Type:** `LeaseBudgetExceeded`

**Schema:**

```
{
  "detail-type": "LeaseBudgetExceeded",
  "detail": {
    "leaseId": {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
    "accountId": "123456789012",
```

```
    "budget": 100.00,
    "totalSpend": 105.50
  }
}
```

### LeaseDurationThresholdAlert

Published when a lease reaches a configured duration threshold.

**Detail Type:** `LeaseDurationThresholdAlert`

**Schema:**

```
{
  "detail-type": "LeaseDurationThresholdAlert",
  "detail": {
    "leaseId": {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
    "accountId": "123456789012",
    "triggeredDurationThreshold": 24,
    "leaseDurationInHours": 168,
    "actionRequested": "ALERT"
  }
}
```

### LeaseFreezingThresholdAlert

Published when a lease reaches a threshold that triggers a freezing action. This event uses the same schema as LeaseFrozen.

**Detail Type:** `LeaseFreezingThresholdAlert`

**Schema:**

```
{
  "detail-type": "LeaseFreezingThresholdAlert",
  "detail": {
    "leaseId": {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
```

```
    "accountId": "123456789012",
    "reason": {
      "type": "BudgetExceeded",
      "triggeredBudgetThreshold": 0.95,
      "budget": 100.00,
      "totalSpend": 95.00
    }
  }
}
```

### LeaseExpired

Published when a lease expires due to reaching its maximum duration.

**Detail Type:** `LeaseExpired`

**Schema:**

```
{
  "detail-type": "LeaseExpired",
  "detail": {
    "leaseId": {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
    "accountId": "123456789012",
    "leaseExpirationDate": "2024-01-22T14:30:25Z"
  }
}
```

### LeaseFrozen

Published when a lease is frozen (access suspended but not terminated).

**Detail Type:** `LeaseFrozen`

**Schema:**

```
{
  "detail-type": "LeaseFrozen",
  "detail": {
    "leaseId": {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
```

```
    },
    "accountId": "123456789012",
    "reason": {
      "type": "BudgetExceeded",
      "triggeredBudgetThreshold": 0.95,
      "budget": 100.00,
      "totalSpend": 95.00
    }
  }
}
```

#### LeaseUnfrozen

Published when a previously frozen lease is unfrozen and access is restored.

**Detail Type:** `LeaseUnfrozen`

**Schema:**

```
{
  "detail-type": "LeaseUnfrozen",
  "detail": {
    "leaseId": {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
    "accountId": "123456789012",
    "maxBudget": 100.00,
    "leaseDurationInHours": 168,
    "reason": "Budget threshold resolved, access restored"
  }
}
```

### Account management events

#### CleanAccountRequest

Published when an account cleanup process is initiated.

**Detail Type:** `CleanAccountRequest`

**Schema:**

```
{
```

```
  "detail-type": "CleanAccountRequest",
  "detail": {
    "accountId": "123456789012",
    "reason": "Lease terminated - budget exceeded"
  }
}
```

### AccountCleanupSucceeded

Published when an account cleanup process completes successfully.

**Detail Type:** `AccountCleanupSucceeded`

**Schema:**

```
{
  "detail-type": "AccountCleanupSucceeded",
  "detail": {
    "accountId": "123456789012",
    "reason": "LEASE_TERMINATION",
    "cleanupExecutionContext": {
      "stateMachineExecutionArn": "arn:aws:states:us-
east-1:123456789012:execution:cleanup-state-machine:execution-id",
      "stateMachineExecutionStartTime": "2024-01-15T14:30:25Z"
    }
  }
}
```

### AccountCleanupFailed

Published when an account cleanup process fails.

**Detail Type:** `AccountCleanupFailed`

**Schema:**

```
{
  "detail-type": "AccountCleanupFailed",
  "detail": {
    "accountId": "123456789012",
    "reason": "LEASE_TERMINATION",
    "cleanupExecutionContext": {
      "stateMachineExecutionArn": "arn:aws:states:us-
east-1:123456789012:execution:cleanup-state-machine:execution-id",
```

```
      "stateMachineExecutionStartTime": "2024-01-15T14:30:25Z"
    }
  }
}
```

### AccountQuarantined

Published when an account is quarantined due to cleanup failures or other issues.

**Detail Type:** `AccountQuarantined`

**Schema:**

```
{
  "detail-type": "AccountQuarantined",
  "detail": {
    "awsAccountId": "123456789012",
    "reason": "Account cleanup failed after multiple attempts"
  }
}
```

### AccountDriftDetected

Published when an account is detected to be in an unexpected organizational unit.

**Detail Type:** `AccountDriftDetected`

**Schema:**

```
{
  "detail-type": "AccountDriftDetected",
  "detail": {
    "accountId": "123456789012",
    "actualOu": "ou-root-1234567890",
    "expectedOu": "ou-sandbox-0987654321"
  }
}
```

## Cost reporting events

### GroupCostReportGenerated

Published when a cost report is successfully generated for a group of accounts.

**Detail Type:** `GroupCostReportGenerated`

**Schema:**

```
{
  "detail-type": "GroupCostReportGenerated",
  "detail": {
    "reportMonth": "2024-01",
    "fileName": "cost-report-2024-01.csv",
    "bucketName": "innovation-sandbox-reports-bucket",
    "timestamp": "2024-02-01T09:00:00Z"
  }
}
```

### GroupCostReportGeneratedFailure

Published when cost report generation fails.

**Detail Type:** `GroupCostReportGeneratedFailure`

**Schema:**

```
{
  "detail-type": "GroupCostReportGeneratedFailure",
  "detail": {
    "reportMonth": "2024-01",
    "timestamp": "2024-02-01T09:00:00Z"
  }
}
```

## Blueprint deployment events

### BlueprintDeploymentRequest

Published when a lease with an attached blueprint is approved, triggering the blueprint deployment workflow.

**Detail Type:** `BlueprintDeploymentRequest`

**Schema:**

```
{
```

```
  "detail-type": "BlueprintDeploymentRequest",
  "detail": {
    "blueprintId": "550e8400-e29b-41d4-a716-446655440000",
    "blueprintName": "Development Environment",
    "leaseId": "650e8400-e29b-41d4-a716-446655440000",
    "userEmail": "developer@company.com",
    "accountId": "123456789012",
    "stackSetId": "arn:aws:cloudformation:us-east-1:123456789012:stackset/dev-env-
stackset:abc123",
    "regions": ["us-east-1", "us-west-2"],
    "regionConcurrencyType": "SEQUENTIAL",
    "deploymentTimeoutMinutes": 30,
    "maxConcurrentPercentage": 100,
    "failureTolerancePercentage": 0,
    "concurrencyMode": "STRICT_FAILURE_TOLERANCE"
  }
}
```

**Note:** `maxConcurrentPercentage`, `failureTolerancePercentage`, and `concurrencyMode` are optional fields.

**BlueprintDeploymentSucceeded**

Published when a blueprint deployment completes successfully and the user can access the account with pre-configured resources.

**Detail Type:** `BlueprintDeploymentSucceeded`

**Schema:**

```
{
  "detail-type": "BlueprintDeploymentSucceeded",
  "detail": {
    "leaseId": {
      "uuid": "650e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
    "blueprintId": "550e8400-e29b-41d4-a716-446655440000",
    "accountId": "123456789012",
    "operationId": "abc123-def456-ghi789",
    "duration": 15
  }
}
```

### BlueprintDeploymentFailed

Published when a blueprint deployment fails due to validation errors, deployment failures, or timeouts. The lease transitions to `ProvisioningFailed` status and the account is sent to cleanup.

**Detail Type:** `BlueprintDeploymentFailed`

**Schema:**

```
{
  "detail-type": "BlueprintDeploymentFailed",
  "detail": {
    "leaseId": {
      "uuid": "650e8400-e29b-41d4-a716-446655440000",
      "userEmail": "developer@company.com"
    },
    "blueprintId": "550e8400-e29b-41d4-a716-446655440000",
    "accountId": "123456789012",
    "operationId": "abc123-def456-ghi789",
    "errorType": "DEPLOYMENT_TIMEOUT",
    "errorMessage": "StackSet deployment failed: Resource creation timeout"
  }
}
```

**Note:** `operationId` is optional and may not be present for validation failures that occur before deployment starts.

**Common failure reasons:**

- StackSet validation failures (not found, wrong permission model, inactive status)
- CloudFormation deployment errors (resource creation failures, insufficient permissions)
- Deployment timeout exceeded (default: 30 minutes, configurable per blueprint)
- Lambda function errors (crashes, timeouts, throttling)

## Schema evolution

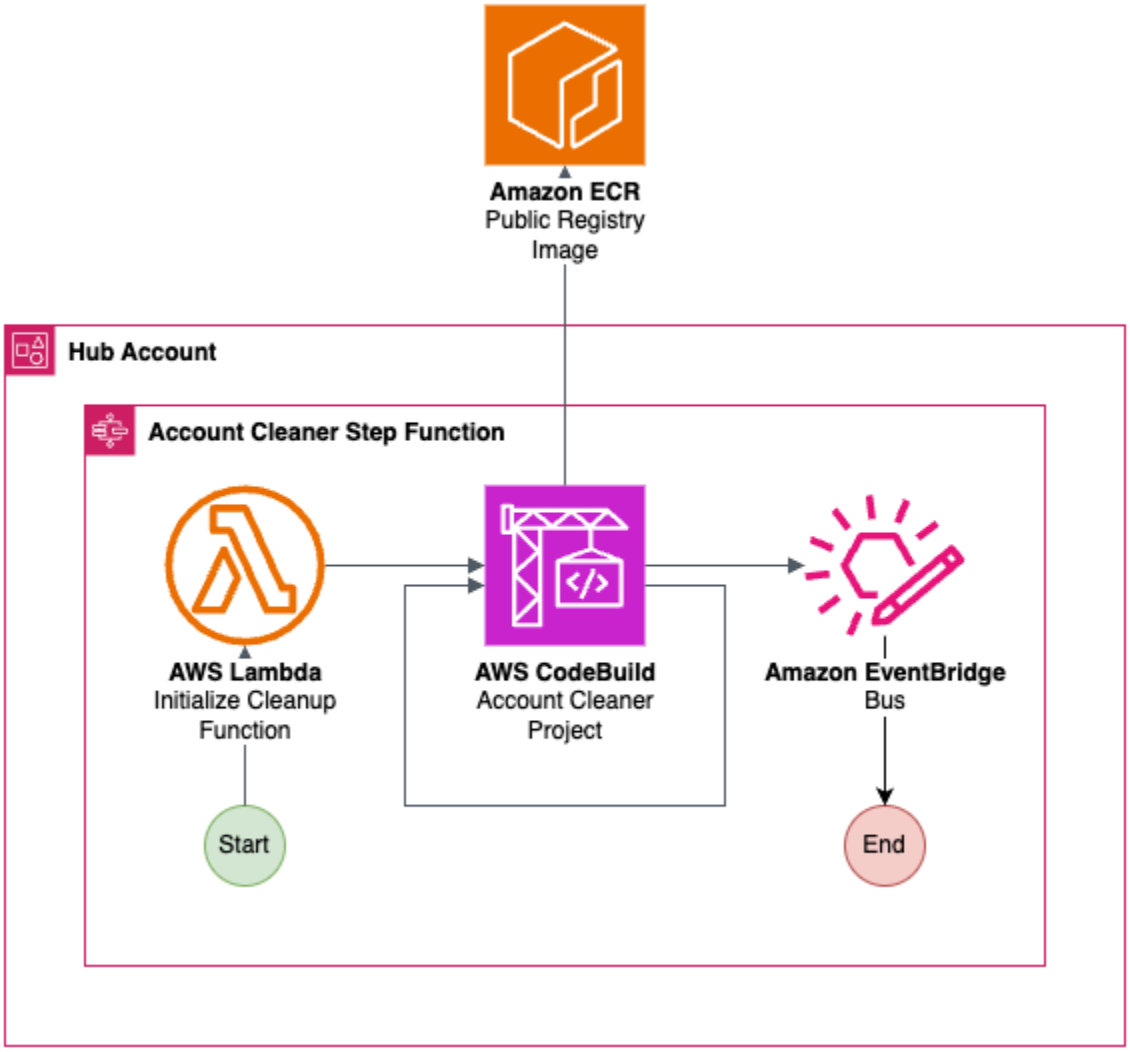
Event schemas may evolve over time. Design your integration to be resilient:

- Ignore unknown fields in event payloads
- Provide default values for missing optional fields

- Version your event processing logic when breaking changes occur

For the complete source code, refer to the Innovation Sandbox on AWS GitHub repository.

## Account Cleaner components




### ISB Account Cleaner components

The **Account Cleaner** is used to clean sandbox accounts either during onboarding to Innovation Sandbox, or after a lease has expired and the account needs to be recycled for reuse. It is composed of an AWS StepFunction with these steps:

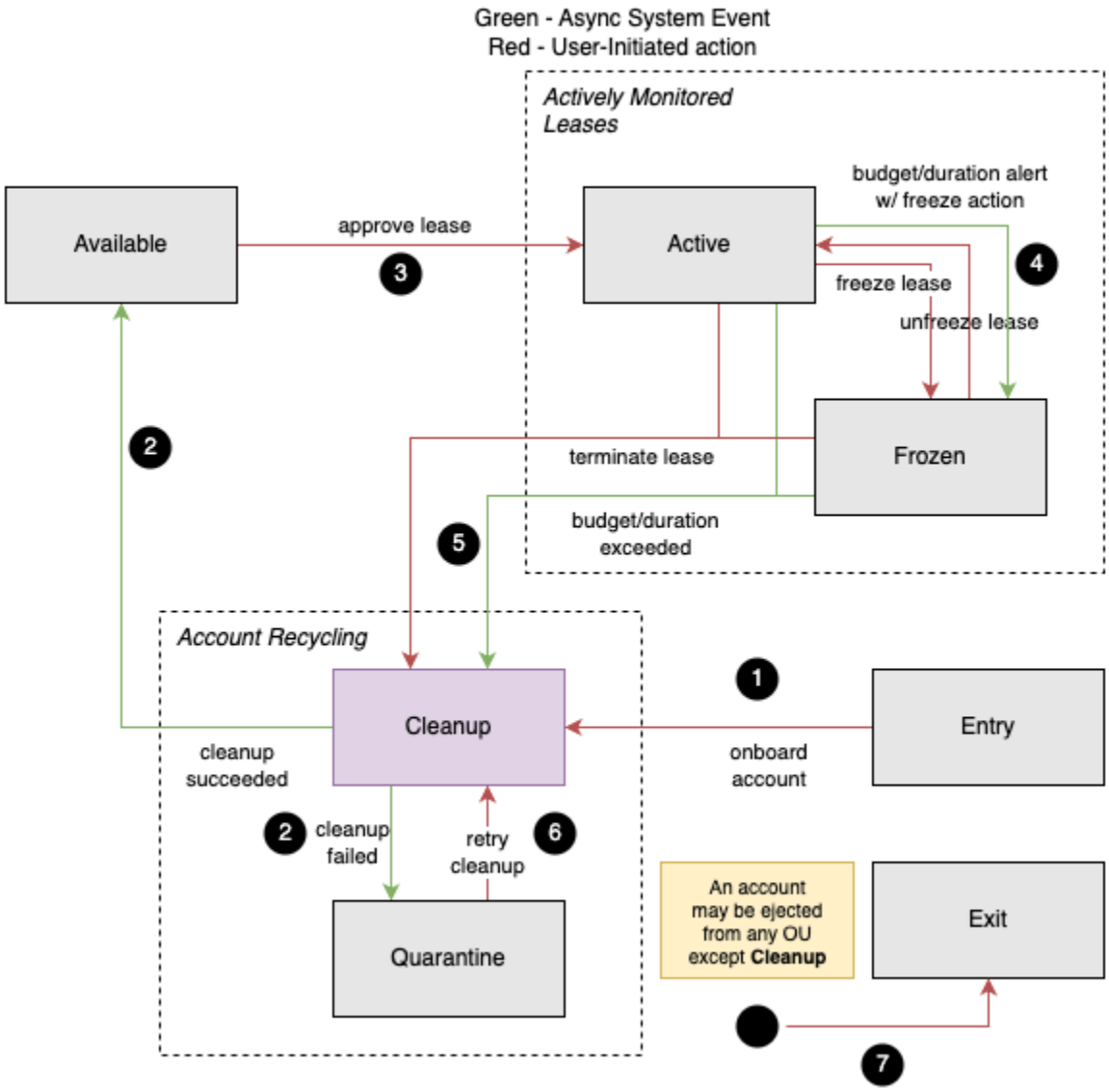
1. The account cleaner invokes the **initialize cleanup** Lambda which performs pre-cleanup actions.

2. An AWS CodeBuild project is initiated that assumes into the sandbox account and runs AWS Nuke on the account to delete all supported resources (Innovation Sandbox configures AWS Nuke to ignore protected solution assets). The CodeBuild project uses a public ECR image with the AWS Nuke binary installed and is managed by the development team.
3. The workflow enters a loop where it attempts clean-up multiple times. This is so that any deletion failures due to resource dependencies eventually resolve themselves and that any resources that are created during clean-up (db snapshots, logs, custom resources OnDelete) are deleted. By default, the solution performs three successful passes of the clean-up loop, but customers can configure this value using AWS AppConfig.

> **Note**
>
> If the clean-up fails, the Step Function (state machine) exits and sends a clean-up failure event to the event bus which moves the account to **Quarantine** OU. If the clean-up is successful, a success event is sent to the event bus which will move the account to **Available** OU so that it can be reused.

# Account lifecycle

This section describes how an account statuses and OU location changes throughout its lifecycle.




## Account lifecycle

1. The account is onboarded into the solution from the **Entry** OU. This is a manual action performed by an administrator that sends the account to clean-up. This sanitizes the account to ensure no previously existing resources make it into the sandbox environment.
2. If the clean-up is successful, an event is produced detailing that the account has been successfully cleaned up. This moves the account to **Available** state.
3. Once available, the lease approval flow will attempt to claim an available sandbox account. Lease approval is a manual API action. During this, the account will move to **Active** state, and a user will be granted access, and the lease's incurred cost and duration will start being monitored.

4. (Optional) **Frozen** status: The account can be sent to a **Frozen** status either manually via an API request, or when the monitoring process detects that a configured threshold for the lease was breached. This revokes account access for the sandbox user and allows the Admin and Manager to review the contents of the account. From the Frozen state, administrators and managers can unfreeze the lease to restore user access, returning the account to **Active** status.
5. From Frozen or Active status, the account can be manually cleaned up by an API request, or the lease monitoring service detecting that the account has reached the end of its configured lease terms. The account will be moved back to **CleanUp** to delete resources that were created under the previous lease so that the account can be used again.
6. During clean-up, if deletion fails (resources were unable to be deleted or an unexpected failure occurs), the account is moved to **Quarantine**. Accounts in Quarantine require manual remediation from the administrator and can only return to the account pool by retrying cleanup and succeeding. Accounts can also be quarantined if drift between the expected account location and actual OU location is detected by the drift monitor Lambda. In this case the account will bypass clean-up, and move straight to Quarantine.
7. From any lifecycle status except accounts going through ongoing clean-up, the account can be **ejected** from the solution. During the ejection process, the solution relinquishes control over the account and places it in a boundary OU named **Exit** where an administrator can safely move it from the account pool. This is useful for preserving work in an account indefinitely, removing a problematic account from the solution, or downsizing the account pool.

# AWS services in this solution

| AWS service | Description |
| --- | --- |
| Amazon CloudFront | **Core**. This solution uses CloudFront with an Amazon S3 bucket as the origin. This restricts access to the Amazon S3 bucket so that it is not publicly accessible and prevents direct access from the bucket. |
| AWS IAM Identity Center | **Core**. The solution uses AWS IAM to authentic ate users for the web application, and role based access to sandbox accounts for solution users. |

| AWS service | Description |
| --- | --- |
| AWS AppConfig | **Core**. The solution uses AWS AppConfig to store configuration data for the solution. |
| AWS Organizations | **Core**. The solution uses AWS Organizations to centrally manage and govern multiple AWS accounts required by the solution. |
| Amazon DynamoDB | **Core**. This solution uses DynamoDB to store state for the solution. |
| AWS Secrets Manager | **Core**. This solution uses AWS Secrets Manager to manage, and store secrets for the SAML2.0 application. |
| AWS Lambda | **Core.** This solution uses serverless Lambda functions, with Node.js to handle API calls. |
| AWS CodeBuild | **Core**. This solution uses CodeBuild for the account clean-up process. |
| Amazon Simple Storage Service | **Core.** This solution uses Amazon S3 for frontend and backend storage purposes. |
| AWS Key Management Service (AWS KMS) | **Core**. This solution uses AWS KMS to manage creation and control of encryption keys, required to encrypt various AWS resources used in the solution. |
| Amazon Simple Queue Service (Amazon SQS) | **Core**. This solution uses Amazon SQS to manage message queues. |
| AWS Step Functions | **Core**. This solution uses Step Functions to orchestrate the account cleanup process and blueprint deployment workflows. |

| AWS service | Description |
| --- | --- |
| Amazon CloudWatch | **Supporting.** This solution uses CloudWatch to collect and visualize real-time logs, metrics, and event data in automated cases. Additiona lly, you can monitor the deployed solution's resource usage and performance issues. |
| AWS Systems Manager | **Supporting.** This solution uses AWS Systems Manager for solution configuration and sharing cross account/stack parameters using the RAM service. |
| AWS WAF | **Supporting.** This solution uses AWS WAF to protect the Amazon API Gateway from common exploits and bots that can affect availability, compromise security, or consume excessive resources. |
| AWS Cost Explorer | **Supporting.** This solution uses AWS Cost Explorer to retrieve cost and usage data for accounts and leases. |

# Plan your deployment

This section describes the Regions, cost, security, and other considerations prior to deploying the solution.

## Supported AWS Regions

Innovation Sandbox on AWS is available in the following AWS Regions. Learn more about enabling regions.

| Region Name | Region Code |
|---|---|
| US East (Ohio) | us-east-2 |
| US East (N. Virginia) | us-east-1 |
| US West (N. California) | us-west-1 |
| US West (Oregon) | us-west-2 |
| Africa (Cape Town) | af-south-1 |
| Asia Pacific (Hong Kong) | ap-east-1 |
| Asia Pacific (Tokyo) | ap-northeast-1 |
| Asia Pacific (Seoul) | ap-northeast-2 |
| Asia Pacific (Osaka) | ap-northeast-3 |
| Asia Pacific (Mumbai) | ap-south-1 |
| Asia Pacific (Hyderabad) | ap-south-2 |
| Asia Pacific (Singapore) | ap-southeast-1 |
| Asia Pacific (Sydney) | ap-southeast-2 |
| Asia Pacific (Jakarta) | ap-southeast-3 |

| Region Name | Region Code |
| --- | --- |
| Asia Pacific (Melbourne) | ap-southeast-4 |
| Canada (Central) | ca-central-1 |
| Europe (Frankfurt) | eu-central-1 |
| Europe (Zurich) | eu-central-2 |
| Europe (Stockholm) | eu-north-1 |
| Europe (Milan) | eu-south-1 |
| Europe (Spain) | eu-south-2 |
| Europe (Ireland) | eu-west-1 |
| Europe (London) | eu-west-2 |
| Europe (Paris) | eu-west-3 |
| Middle East (UAE) | me-central-1 |
| Middle East (Bahrain) | me-south-1 |
| South America (São Paulo) | sa-east-1 |

Innovation Sandbox on AWS is **not** available in the following AWS Regions:

| Region Name | Region Code |
| --- | --- |
| Asia Pacific (Malaysia) | ap-southeast-5 |
| Asia Pacific (Thailand) | ap-southeast-7 |
| Canada West (Calgary) | ca-west-1 |
| China (Beijing) | cn-north-1 |

| Region Name | Region Code |
| --- | --- |
| China (Ningxia) | cn-northwest-1 |
| Israel (Tel Aviv) | il-central-1 |
| Mexico (Mexico City) | mx-central-1 |
| AWS GovCloud (US-East) | us-gov-east-1 |
| AWS GovCloud (US-West) | us-gov-west-1 |

For the most current availability of AWS services by Region, see the AWS Regional Services List.

**Important**

**CloudFront Access Logging Limitation**

As of September 2025, CloudFront access logging is automatically disabled in the following regions due to lack of support for standard logging (legacy):

- Africa (Cape Town) - `af-south-1`
- Asia Pacific (Hong Kong) - `ap-east-1`
- Asia Pacific (Hyderabad) - `ap-south-2`
- Asia Pacific (Jakarta) - `ap-southeast-3`
- Asia Pacific (Melbourne) - `ap-southeast-4`
- Canada West (Calgary) - `ca-west-1`
- Europe (Milan) - `eu-south-1`
- Europe (Spain) - `eu-south-2`
- Europe (Zurich) - `eu-central-2`
- Israel (Tel Aviv) - `il-central-1`
- Middle East (Bahrain) - `me-south-1`
- Middle East (UAE) - `me-central-1`

If you deploy the solution in one of these regions, the CloudFront distribution will function normally but will not generate access logs. If access logging is required for your use case,

you can manually configure CloudFront Standard Logging V2 after deployment. For more information, refer to the CloudFront Standard Logging V2 documentation.

# Choosing the deployment accounts

## Accounts

To deploy this solution, you will need access to these accounts.

**Organizations Management account**

The **AccountPool** stack, deployed into the AWS Organizations management account, is used to manage the lifecycle on sandbox accounts controlled by the solution.

This stack consists of a single IAM role that will be assumed by the Hub stack's Lambda function and grants minimal required permissions to access data of the Organization. The permissions on this role are least privileged to only allow read actions from Cost Explorer, read actions on the account pool OUs, and move account actions on the account pool OUs. The trust policy on the role only allows for a single Intermediate IAM role from the Compute stack to assume into it.

**IAM IDC account**

The **IAM Identity Center (IDC)** stack deployed into the AWS Account containing the organizations AWS IAM Identity Center instance, is used to manage the solution web UI and sandbox account access.

This stack initializes user groups and corresponding permission sets in the instance that administrators can manually add users to. The IDC stack also contains an IAM Role. The permissions on this role are least privileged to only allow the actions required by the solution. The trust policy on the role only allows for a single Intermediate IAM role from the Compute stack to assume into it.

**Hub account**

The **Data** and **Compute** stacks contain all data, compute, and storage resources for the solution to serve the frontend application, handle API requests, facilitate scans, and manage the account lifecycle.

Select a member account within your AWS Organization to deploy these stacks. This account will have administrative access to the spoke accounts to enable the Account Cleaner component for account recycling operations. Due to these elevated permissions, treat the Hub account as a highly sensitive asset. We strongly recommend using a dedicated account with stringent access controls and limiting the number of users who can access it. Implement robust security measures to protect this account, similar to accounts you would use for your most critical AWS environments.

> **Important**
>
> We do not recommend using the Organizations Management account to keep the management account free from operational workloads.

#### Sandbox account

The **SandboxAccount** stack is automatically configured as a service-managed StackSet resource in the AccountPool stack, using the **AccountPool OU** as the deployment target. This stack contains a single **Spoke** role, which is crucial for the account clean-up process. The Spoke role is automatically created by the service-managed StackSet after onboarding the sandbox accounts. It is assumed by compute resources in the Compute stack to run the account clean-up job.

> **Important**
>
> These sandbox accounts are strictly intended for non-production usage and should never run production workloads.

## Home Region

Identifying the home Region is crucial for the successful deployment of the ISB solution. For the solution to work as expected:

- Deploy all four stacks in the same Region.
- Enable IDC in the same home Region. Identify the Region where IDC is enabled in your AWS Organization, as this will be the home Region for for the ISB solution.

**Note**

The home Region is only for deployment resources. The sandbox accounts can use any Regions that are defined in the managed Regions list (CFN Param).

# Configuring an external identity provider (Optional)

## Group Management

Innovation Sandbox on AWS uses three different user groups that align with the different personas. These groups must be created following your normal process within the external provider. The group names must be exactly the same as they are specified in the IDC CloudFormation Stack parameters.

Personas and corresponding groups:

| Persona | Default Group Name | Responsibility |
|---|---|---|
| Admin | <namespace>_IsbAdminsGroup | The Admin persona is responsible for deploying and managing the solution and managing the AWS accounts used in the solution. |
| Manager | <namespace>_IsbManagersGroup | The Manager persona is responsible for the creation and management of the Lease Templates (Sandbox thresholds and actions) and the Leases (active Sandbox accounts). |
| User | <namespace>_IsbUsersGroup | The User persona is responsible for requesting and using Leases (Sandbox Accounts) |

## User Management

Users will be managed according to your normal process within your provider by adding the appropriate users into the one of the 3 ISB user groups.

Requirements:

- **Email**: Ensure that the primary email field in the provider is populated with the correct email address.
  - Microsoft Entra: `mail`
  - Okta: `email`
- The primary email field must be configured within your provider to be passed to IAM Identity Center.

You can confirm that a user's email attribute has been successfully mapped and passed to the correct field in IAM Identity Center by running the following command in the **IDC Account** (Management or delegated account):

```
aws identitystore list-users --identity-store-id $(aws sso-admin list-instances --query
 "Instances[0].IdentityStoreId" --output text)
```

You can confirm that the correct email address is populated in the Emails array as shown below. The Email value should be correct and Primary should be set to true.

```
"Emails": [
    {
       "Value": "example@amazon.com",
       "Type": "work",
       "Primary": true
    }
]
```

## Attribute mapping examples

The attribute mappings within your provider must be configured to map the user's primary email field (from provider) to `emails[type eq "work"]` (to IAM Identity Center).

| External identity provider | Provider attribute | IAM Identity Center attribute |
|---|---|---|
| Microsoft Entra | mail | emails[type eq "work"] |
| Okta | email | emails[type eq "work"] |

# Control Tower managed organizations

The Innovation Sandbox on AWS solution creates an account pool organizational unit (default: InnovationSandboxAccountPool) when you deploy the Account Pool Stack. This OU is created through AWS Organizations and is not managed by Control Tower. This OU and all nested OUs do not need to be registered with Control Tower.

If you choose to register the OU within Control Tower, or deploy the OU as a nested OU in an already Control Tower-managed OU, the parent OU (InnovationSandboxAccountPool OU), nested OUs, and accounts will appear in a drifted state in the Control Tower console. This is expected behavior because the solution moves accounts between the nested OUs.

# Creating sandbox accounts

The Innovation Sandbox on AWS solution works with existing AWS accounts and does not create new accounts. Create new accounts using AWS Organizations. For more information, refer to Creating a member account in an organization with AWS Organizations.

Create the number of accounts that you want to start with in your Account Pool based on the number of expected concurrent users. For example, if you expect 10 concurrent users, create 10 sandbox accounts using AWS Organizations.

**Note**

The size of the account pool can be adjusted at any time. If you are unsure of how many accounts you will need, you can start with a smaller number and expand the pool as necessary (Adding new accounts to the account pool). You can also reduce the pool size in the future (Managing existing accounts).

**Note**

If you use accounts that are created or managed from AWS Control Tower, they will show as drifted in the AWS Control Tower console because the solution moves the accounts between OUs.

# Cost

You are responsible for the cost of the AWS services provisioned while running this solution. As of this revision, the cost for running this solution using the single instance deployment option in the US East (N. Virginia) Region is approximately **USD $65.25 per month**.

> **Note**
>
> The cost for running Innovation Sandbox on AWS in the AWS Cloud depends on the deployment configuration you choose. The following examples provide cost breakdown for various deployment configurations in the US East (N. Virginia) Region. AWS services listed in the example tables below are billed (in US$) on a monthly basis.

We recommend creating a budget through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

## Example cost table

| Deployment type | Small deployment | Medium deployment | Large deployment |
|---|---|---|---|
| Example | 50 accounts, 30 leases (per month), 10 lease templates | 300 accounts, 150 leases (per month), 80 lease templates | 1000 accounts, 500 leases (per month), 100 lease templates |
| **AWS Services** | Cost (USD) | Cost (USD) | Cost (USD) |
| Amazon DynamoDB | $0.25 | $1.20 | $3.71 |
| AWS Lambda | $4.41 | $4.51 | $4.81 |
| AWS KMS | $4.91 | $4.91 | $4.92 |
| Amazon API Gateway | $1.05 | $1.05 | $1.05 |
| AWS WAF | $11.18 | $11.18 | $11.18 |
| AWS CodeBuild | $6.75 | $33.75 | $112.50 |

| Deployment type | Small deployment | Medium deployment | Large deployment |
|---|---|---|---|
| AWS Step Functions | $0.18 | $0.91 | $3.04 |
| Amazon CloudFront | $0.21 | $0.22 | $0.22 |
| Amazon Simple Email Service | $0.02 | $0.11 | $0.35 |
| AWS CostExplorer | $7.20 | $7.20 | $7.20 |
| **Total Cost per month (USD)** | **~$36.40** | **~$65.25** | **~$149.20** |

**Important**

This estimate does not include the costs incurred by sandbox account usage or blueprint deployments. Customers are responsible for setting appropriate lease configurations, monitoring spend of sandbox accounts, and considering the cost of resources deployed through blueprints.

**Note**

Blueprint deployments may incur additional costs depending on the resources defined in your CloudFormation StackSets. Consider the cost of blueprint resources when planning your deployment and setting lease budget limits. For example, a blueprint that deploys Amazon RDS databases, Amazon ElastiCache clusters, or Amazon EC2 instances will incur ongoing costs for the duration of the lease.

# Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared responsibility model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization

layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit the AWS Security Center.

## IAM roles

IAM roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. Multiple roles are required to run Innovation Sandbox on AWS and discover resources in AWS accounts.

## IAM Identity Center and SAML authentication

AWS IAM Identity Center provides a central way to manage access to multiple AWS accounts and business applications using SAML 2.0-based authentication. By configuring SAML authentication through IAM Identity Center, you can allow your users to sign in to the solution's web UI using their existing corporate credentials. This eliminates the need to manage separate user accounts and passwords within the solution.

## AWS Key Management Service

This solution creates four KMS Customer Managed Keys (one for each stack - AccountPool, IDC, Data, and Compute) to encrypt various AWS resources. The encrypted services include CloudWatch Logs, Amazon Simple Queue Service (SQS) queues, EventBridge event buses, Secrets Manager secrets, CodeBuild projects, and DynamoDB tables.

Each CMK is specifically tailored to its stack's requirements, with appropriate key policies that grant necessary permissions to relevant services and IAM roles. This approach of using separate CMKs per stack follows the principle of separation of concerns and allows for more granular control over encryption permissions across different components of the solution.

## AWS WAF

In this solution, AWS WAF (Web Application Firewall) is implemented to protect the API Gateway endpoints through multiple layers of security controls. The solution creates a regional WAF web ACL that combines four AWS managed rule groups and two custom rules.

The default action of the web ACL is set to **allow** and the rule actions are set to **block**, so any request that does not satisfy all rules will be blocked. This comprehensive WAF configuration helps protect the API Gateway against common web exploits, malicious bots, and unauthorized access while allowing legitimate traffic from approved sources.

**Note**

WAF SizeRestrictions_QUERYSTRING Rule Modification*
The solution disables the `SizeRestrictions_QUERYSTRING` rule from the `AWSManagedRulesCommonRuleSet` to accommodate legitimate large pagination tokens from the AWS Organizations API. The GET `/accounts/unregistered` endpoint retrieves accounts from AWS Organizations, which can return pagination tokens that exceed the default WAF query string size limit when handling large numbers of accounts (>20).
This modification is necessary for the solution to function properly with large account pools. If you require additional query string size protection for other endpoints, you can manually implement a custom rule that excludes the `/accounts/unregistered` endpoint while applying size restrictions to other API endpoints.

## Amazon CloudFront

This solution deploys a web UI hosted in an Amazon S3 bucket that is distributed by Amazon CloudFront. To help reduce latency and improve security, this solution includes a CloudFront distribution with an origin access identity, which is a CloudFront user that provides public access to the solution website's bucket contents. By default, the CloudFront distribution uses TLS 1.2 to enforce the highest level of security protocol. For more information, refer to Restricting access to an Amazon S3 origin in the *Amazon CloudFront Developer Guide*.

CloudFront activates additional security mitigations to append HTTP security headers to each viewer response. For more information, refer to Adding or removing HTTP headers in CloudFront responses.

This solution uses the default CloudFront certificate which has a minimum supported security protocol of TLS v1.0. To enforce the use of TLS v1.2 or TLS v1.3, you must use a custom SSL certificate instead of the default CloudFront certificate. For more information, refer to How do I configure my CloudFront distribution to use an SSL/TLS certificate.

## Amazon DynamoDB

All user data stored in DynamoDB is encrypted at rest using customer managed keys (CMK) stored in AWS KMS.

## AWS Lambda

By default, the Lambda functions are configured with the most recent stable version of the language runtime. No sensitive data or secrets are logged. Service interactions are carried out with the least required privilege. Roles that define these privileges are not shared between functions.

## Amazon CloudWatch Alarms

The solution provides CloudWatch Alarms through CloudWatch Application Insights to monitor for Lambda errors, throttling, and execution duration.

To set up SNS notifications to detect changes in these alarms, refer to Acting on alarm changes. You can configure additional alarms based on metrics reported by the different services within the solution.

## Log retention and monitoring

By default, Innovation Sandbox retains all compute logs for 90 days in Amazon CloudWatch Logs. AWS recommends retaining security-relevant logs for 10 years to support compliance and forensic analysis requirements. You can modify the default log retention period by adjusting the `cloudWatchLogRetentionInDays` value in the CloudFormation template mapping before deployment.

All logs are encrypted at rest using AWS KMS customer-managed keys and are automatically archived to Amazon S3 for long-term retention following a multi-tier strategy (CloudWatch Logs → S3 Standard → S3 Glacier).

## AWS CloudTrail

AWS CloudTrail is not automatically enabled by the Innovation Sandbox solution. AWS recommends enabling organization-level CloudTrail in your Organization Management Account to monitor API calls and administrative actions across all accounts.

## Amazon S3 security features

The solution uses Amazon S3 for storing cost reports, log archives, and operational data. By default, the solution only enables S3 access logging and versioning on critical buckets to reduce costs from redundant logs. AWS recommends enabling these features on all solution buckets for enhanced security monitoring if required for your compliance needs.

If desired, you can manually enable S3 access logging to monitor all bucket access, S3 versioning to protect against accidental deletion or modification, and S3 event notifications for real-time alerts on critical bucket operations.

### Custom client security considerations

The Innovation Sandbox on AWS API allows certain free-text fields (such as lease template names and descriptions) to contain characters that may lead to cross-site scripting (XSS) vulnerabilities in insecure client implementations. The included React-based web client implements proper security controls and safely handles all user-provided data. If you develop a custom client application that integrates with the solution's API, ensure your implementation includes appropriate input validation, output encoding, and XSS protection measures following secure coding practices for your chosen technology stack.

## Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

### Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, refer to AWS service quotas.

Use the following links to view service quotas. To view the service quotas for all AWS services in the documentation without switching pages, refer to the Service endpoints and quotas page.

| | |
|---|---|
| Amazon EventBridge | AWS CodeBuild |
| Amazon CloudFront | Amazon API Gateway |
| AWS AppConfig | AWS CloudFormation |
| Amazon DynamoDB | AWS IAM Identity Center |
| AWS KMS | AWS Lambda |
| Amazon CloudWatch Logs | AWS Organizations |

| | |
|---|---|
| AWS RAM | Amazon S3 |
| AWS Secrets Manager | Amazon SQS |
| AWS Systems Manager Parameter Store | AWS Step Functions |

## AWS CloudFormation quotas

Make sure you are aware of AWS CloudFormation quotas when launching the stack in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, refer to AWS CloudFormation quotas in the *AWS CloudFormation User's Guide*.

## AWS Lambda quotas

Your account has an AWS Lambda concurrent execution quota of 1000. If the solution is used in an account where there are other workloads running and using Lambda, set this quota to an appropriate value. This value is adjustable; for more information, see AWS Lambda quotas in the *AWS Lambda User's Guide*.

## AWS CodeBuild quotas

Make sure you are aware of AWS CodeBuild quotas when launching the stack in this solution.

> **Note**
>
> By default, concurrent CodeBuild quotas are low. To efficiently handle account recycling with this solution, we recommend you request a higher concurrent build quota before you launch the solution.

# Deploy the solution

This solution uses AWS CloudFormation templates and stacks to automate its deployment.

1. The CloudFormation template specifies the AWS resources included in this solution and their properties.
2. The CloudFormation stacks provision the resources that are described in the template.

## Deployment process overview

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

**Time to deploy:** Approximately 60 minutes

Before you launch the solution, review the Cost, Architecture, Network security, and other considerations discussed in this guide.

## Prerequisites

Before launching the stacks, you must meet the following prerequisites:

1. **Identify the AWS account where you want to deploy the solution**: Use the AWS Management Console to identify and name this as the **Hub** account. We recommend you dedicate this account for running the solution with no other workloads running in the account.
2. **Verify your home Region**: You must deploy all the stacks in the same AWS Region, and enable the Identity Center (IDC) in the same home Region. If you have already enabled IDC, use that Region as your home Region.
3. **Ensure you have set up an AWS Organization to deploy the solution into**: AWS Organizations helps you centrally manage and govern your environment as you grow and scale your AWS resources. For more information on how to get started, refer to the Creating and configuring an organization tutorial.
4. **Ensure you have enabled Service Control Policies with Organizations**: For more information, refer to Managing organization policies with AWS Organizations.
5. **Ensure you have enabled and set up AWS IAM Identity Center**: AWS IAM Identity Center is used to centrally manage access to your AWS accounts and applications. Enable IAM Identity Center

at the organizational level, either using the Organization Management account or a delegated administration account.

- To enable IAM Identity Center, open the IAM Identity Center console, select your home Region, and on the main page, for Enable IAM Identity Center, choose **Enable**.

6. **Configure Amazon SES for the application to send email notifications**: Set up SES for the solution and request production access using the Hub account. For more information, refer to Setting up Amazon SES and Requesting production access.
7. **Enable resource sharing using AWS Resource Access Manager (RAM)**: For more information on how to set this up, refer to Enable resource sharing within AWS Organizations.
8. **Activate trusted access for CloudFormation StackSets**: AWS CloudFormation StackSets extends the capability of stacks by allowing you to create, update, or delete stacks across multiple accounts and AWS Regions with a single operation. For more information on how to activate trusted access, refer to Activate trusted access for stacksets with AWS Organizations.
9. **Enable Cost Explorer on the Org Management account**: Ensure that you have enabled Cost Explorer for tracking costs. For more information, refer to Enable Cost Explorer. Note that Cost Explorer requires approximately 24 hours to be enabled for your account.
10. **Dedicated AWS Lambda concurrent executions limit**: Use AWS Service Quotas in your AWS console to verify your AWS Lambda concurrent executions.
    - The Applied quota value in your account should be greater than or equal to the AWS default quota value (which is 1000). If the Applied quota value is less than 1000, select the **Request quota increase** button to request an increase to this value to at least 1000 before deploying the solution. For more information, refer to the AWS Lambda Developer Guide.
11. **Ensure that all accounts used are members of the AWS Organization**: The deployment will fail if this is not the case.

# AWS CloudFormation templates

This solution uses AWS CloudFormation to automate the deployment of Innovation Sandbox on AWS in the AWS Cloud. It includes the following CloudFormation template, which you can download before deployment.

## AccountPool stack

View template

**InnovationSandbox-AccountPool.template** - Use this template to deploy the resources required to set up Organizational Units (OUs), Service Control Policies (SCPs), roles, and Regions.

## IDC stack

View template

**InnovationSandbox-IDC.template** - Use this template to deploy the resources required to set up IDC, including mappings, roles, policies, and other configuration.

## Data stack

View template

**InnovationSandbox-Data.template** - Use this template to deploy the data resources required for the application. This stack also contains the AWS AppConfig hosted configurations for the solution's global configurations and Nuke configurations.

## Compute stack

View template

**InnovationSandbox-Compute.template** - Use this template to deploy the compute resources required for the ISB application. This stack contains all of the stateless (compute) resources used by the solution, including the web application and the event infrastructure.

> ⚠ **Important**
>
> The **SandboxAccount** stack is automatically configured as a service-managed StackSet resource in the **AccountPool** stack using the **AccountPool OU** as deployment target. The stack contains a single **Spoke** role that is assumed into by compute resources in the compute stack to run the account clean-up job.

These AWS CloudFormation templates deploy the Innovation Sandbox on AWS solution in the AWS Cloud.

# Launch the stacks

You must gather deployment parameter details before deploying the stacks. For details, refer to Prerequisites.

**Time to deploy:** Approximately 60 minutes

You must deploy these four stacks for the Innovation Sandbox solution in the following order. Failing to do so will result in deployment failures.

1. Step 1: Deploy the `AccountPool` stack
2. Step 2: Deploy the `IDC` stack
3. Step 3: Deploy the `Data` stack
4. Step 4: Deploy the `Compute` stack

## Step 1: Deploy the AccountPool stack

In this step, you will deploy the resources required to set up Organizational Units (OUs), Service Control Policies (SCPs), roles, and Regions.

> **Important**
>
> Ensure that you log into the **Org Management** account for deploying the AccountPool stack.

> **Note**
>
> Refer to Supported AWS Regions for a list of supported AWS Regions.

1. Sign in to the AWS Management Console and select the button to launch the `AccountPool` stack CloudFormation template.

Launch solution

The template launches in the US East (N.Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

2. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box, and choose **Next**.
3. On the **Specify stack** details page, enter a stack name for your solution stack. For information about naming character limitations, see IAM and AWS STS quotas, name requirements, and character limits in the AWS Identity and Access Management User Guide.
4. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

| Parameter | Default | Description |
| --- | --- | --- |
| **Namespace** | *myisb* | The namespace for this deployment of Innovation Sandbox (must be the same for all member stacks). For example, **myisb**. |
| **Hub Account Id** | *<Requires input>* | The AWS Account Id where the Innovation Sandbox Hub application (Data and Compute stacks) is (to be) deployed. This refers to the Hub account you have identified in the Prerequisites section. |
| **Parent OU Id** | *<Requires input>* | Provide the Root id or organization unit id where Innovation Sandbox OUs will be created. To find the OU Id, navigate to AWS Organizations to view the details of the OU that you would like to use. |

| Parameter | Default | Description |
| --- | --- | --- |
| **ISB Managed Regions** | *<Requires input>* | Provide a comma-separated list of AWS Regions to limit sandbox usage to specific regions. Always include `us-east-1` to enable global services. Example: `us-east-1,eu-west-1` |

5. Choose **Next**.
6. On the **Configure stack options** page, review and select to acknowledge the messages under **Capabilities and transforms**, and choose **Next**.
7. On the **Review and create** page, review and confirm the settings.
8. Choose **Submit** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation Console in the Status column. You should receive a **CREATE_COMPLETE** status in approximately 60 minutes.

> **Note**
>
> Always include `us-east-1` as an ISB Managed Region to enable AWS global services. For example, if you want to enable `eu-west-1`, the parameter value should be `us-east-1,eu-west-1`.

## Step 2: Deploy the IDC stack

In this step, you will deploy the resources required to set up IDC, including mappings, roles, policies, and other configuration.

> **Important**
>
> Ensure that you log in using the account where you have configured the IAM Identity Center Instance for your AWS Organization. This can be either the Organization

Management account or a delegated administration account that has been configured for IAM Identity Center.

**Note**

**Using a Delegated Administration Account for IAM Identity Center**: AWS recommends using a delegated administration account for IAM Identity Center rather than the Organization Management account for security best practices. If you are using a delegated administration account, ensure that:

- The delegated administration account has been properly configured for IAM Identity Center
- You deploy the IDC stack in the delegated administration account
- You provide the Organization Management account ID in the **Org Management Account Id** parameter (not the delegated admin account ID)

For more information on setting up delegated administration for IAM Identity Center, refer to the AWS IAM Identity Center delegated administration documentation.

1. Sign in to the AWS Management Console and select the button to launch the `IDC` stack CloudFormation template.


The template launches in the US East (N.Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

2. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box, and choose **Next**.
3. On the **Specify stack** details page, enter a stack name for your solution stack. For information about naming character limitations, see IAM and AWS STS quotas, name requirements, and character limits in the AWS Identity and Access Management User Guide.

4. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

> **Important**
>
> When using an external identity provider with SCIM integration (such as Microsoft Entra or Okta), you must create the ISB user groups in the external provider using the exact names specified in the group name parameters below, or the default names if left empty.

| Parameter | Default | Description |
| --- | --- | --- |
| **Namespace** | *myisb* | Use the same namespace from the `AccountPool` stack deployment of Innovation Sandbox. For example, **myisb**. |
| **Org Management Account Id** | *<Requires input>* | The AWS Account Id of the Organization Management account. This is always the Organization Management account ID, even if you are using a delegated administration account for IAM Identity Center. |
| **Hub Account Id** | *<Requires input>* | The AWS Account Id where the Innovation Sandbox Hub application (Data and Compute stacks) is (to be) deployed. |

| Parameter | Default | Description |
| --- | --- | --- |
| **Identity Store Id** | *<Requires input>* | The Identity Store Id of the IAM Identity Center Instance. Example: d-XXXXXXXXXX. To obtain the IdentityStoreId value from the IAM Identity Center console:<br><br>- Log in to the account your IDC account is located in.<br><br>- Open the IAM Identity Center console, and from the left pane, select **Settings**.<br><br>- From the Settings page, on the Identity source tab, copy the **Identity Store ID** value. |

| Parameter | Default | Description |
| --- | --- | --- |
| **SSO Instance Arn** | *<Requires input>* | The ARN of the SSO instance in IAM Identity Center. Example: arn:aws:sso:::instance/ssoins- xxxxxxxxx xxxxxxx. To obtain the SsoInstanceArn value from the IAM Identity Center console:<br><br>- Log in to the account your IDC account is located in.<br><br>- Open the IAM Identity Center console, and from the left pane, select **Settings**.<br><br>- From the Settings page, under Details, copy the **Instance ARN** value. |
| **Admin Group Name** | *<Empty>* | A custom name to provide for the admin group. **Note**: If left empty, the group will be created with the name **<namespace>_IsbAdminsGroup**. |
| **Manager Group Name** | *<Empty>* | A custom name to provide for the manager group. **Note**: If left empty, the group will be created with the name **<namespace>_IsbManagersGroup**. |

| Parameter | Default | Description |
| --- | --- | --- |
| **User Group Name** | *<Empty>* | A custom name to provide for the user group. **Note**: If left empty, the group will be created with the name **<namespace>_IsbUsersGroup**. |

5. Choose **Next**.
6. On the **Configure stack options** page, review and select to acknowledge the messages under Capabilities and transforms, and choose **Next**.
7. On the **Review and create** page, review and confirm the settings.
8. Choose **Submit** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation Console in the Status column. You should receive a **CREATE_COMPLETE** status in approximately 60 minutes.

## Step 3: Deploy the Data stack

In this step, you will deploy the data resources required for the ISB application.

> **Important**
>
> Ensure that you are logged in using the **Hub** account for deploying the Data stack.

1. Sign in to the AWS Management Console and select the button to launch the `Data` stack CloudFormation template.


The template launches in the US East (N.Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

2. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box, and choose **Next**.
3. On the **Specify stack** details page, enter a stack name for your solution stack. For information about naming character limitations, see IAM and AWS STS quotas, name requirements, and character limits in the AWS Identity and Access Management User Guide.
4. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

| Parameter | Default | Description |
| --- | --- | --- |
| **Namespace** | *myisb* | Use the same namespace from the Account Pool stack deployment of Innovatio n Sandbox. For example, **myisb**. |

5. Choose **Next**.
6. On the **Configure stack options** page, review and select to acknowledge the messages under Capabilities and transforms, and choose **Next**.
7. On the **Review and create** page, review and confirm the settings.
8. Choose **Submit** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation Console in the Status column. You should receive a **CREATE_COMPLETE** status in approximately 60 minutes.

## Step 4: Deploy the Compute stack

In this step, you will deploy the compute resources required for the ISB application.

> **Important**
>
> Ensure that you are logged in using the **Hub** account for deploying the Compute stack.

1. Sign in to the AWS Management Console and select the button to launch the `Compute` stack CloudFormation template.

**Launch solution**

The template launches in the US East (N.Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

2. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box, and choose **Next**.
3. On the **Specify stack** details page, enter a stack name for your solution stack. For information about naming character limitations, see IAM and AWS STS quotas, name requirements, and character limits in the AWS Identity and Access Management User Guide.
4. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

| Parameter | Default | Description |
|---|---|---|
| **Namespace** | *myisb* | Use the same namespace from the Account Pool stack deployment of Innovatio n Sandbox. For example, **myisb**. |
| **Org Management Account Id** | *<Requires input>* | The AWS Account Id of the org management account where the Account Pool and IDC stacks are deployed. |
| **IDC Account Id** | *<Requires input>* | The AWS Account Id where the IAM Identity Center is configured. |
| **Allow Listed IP Ranges** | 0.0.0.0/1,128.0.0.0/1 | Comma separated list of CIDR ranges that allow access to the API. |
| **Use Stable Tagging** | Yes | Automatically use the most up to date and secure |

| Parameter | Default | Description |
| --- | --- | --- |
| | | account cleaner image up until the next minor release.<br><br>**Note**: Selecting 'No' will pull the image as originall y released, without any security updates. |
| **Accept Solution Terms of Use** | *<Requires input>* | Solution's terms of use statement for review. The solution will not deploy unless you enter **Accept** in the parameter field. |

5. Choose **Next**.
6. On the **Configure stack options** page, review and select to acknowledge the messages under Capabilities and transforms, and choose **Next**.
7. On the **Review and create** page, review and confirm the settings.
8. Choose **Submit** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation Console in the Status column. You should receive a **CREATE_COMPLETE** status in approximately 60 minutes.

# Post-deployment configuration tasks

After you successfully deployed the stacks, complete the following tasks.

- Configure the IAM Identity Center
- Configure the web application

# Configure IAM Identity Center

Log in to the account where IAM Identity Center is enabled (usually the **Org Management** account) and the Innovation Sandbox IDC stack is deployed. Make sure that you are in the correct home Region.

In this section, you will:

- Create a SAML 2.0 application
- Map application attributes
- Assign groups to your application
- Assign users to groups

## Create a SAML 2.0 application

In this step, you federate your Identity Provider (IdP) to IAM Identity Center through SAML 2.0, and use IAM Identity Center to manage user access to the solution.

1. Log in to the AWS IAM Identity Center console.
2. From the left pane, under **Application assignments**, choose **Applications**.
3. On the Applications page, on the **Customer managed** tab, choose **Add application**.
4. On the **Select application type** page, under **Setup preference**, choose **I have an application I want to set up**.
5. Under **Application type**, choose **SAML 2.0**, and choose **Next**.
6. On the **Configure application** page, under **Configure application**,
   - Enter a **Display name** for the application, such as *MyISBApp*,
   - Enter a description.

7. Under **Application metadata**, choose **Manually type your metadata values**, and provide the **Application ACS URL** and **Application SAML audience** values.
   - **Application ACS URL**: The URL of the CloudFront distribution (or alternate domain name associated with the distribution) from the Compute stack output appended with `/api/auth/login/callback`. For example: `<ISB_WEB_URL>/api/auth/login/callback` where `ISB_WEB_URL` is the CloudFront Distribution URL or alternate domain (for example: https://duyXXXXXXeh.cloudfront.net/api/auth/login/callback). To view the Compute stack outputs, navigate to the **AWS CloudFormation > Stacks > Outputs** tab, in the account where you have deployed the Compute stack.
   - **Application SAML audience**: The audience used to identify the service provider (in this case, Innovation Sandbox web application) configured to consume the SAML assertion. For example: `Isb-<NAMESPACE>-Audience`.
8. Choose **Submit**. The Application details page displays.

## Map application attributes

In this step, you map application attributes to the user attribute in IAM Identity Center, using the email address for authentication.

1. From the list of applications, choose the SAML application you set up in the previous step.
2. Under **Actions**, choose **Edit attribute mappings**.
3. For the *Subject* **User attribute in the application** row, fill in the two corresponding fields:

| Field | Value |
|---|---|
| Maps to this string value or user attribute in IAM Identity Center | ${user:email} |
| Format | emailAddress |

4. Choose **Save Changes**.

> **Note**
>
> If you have configured IAM Identity Center to use an external identity provider, you need to ensure that the attribute mappings from external identity provider to IAM Identity Center

are configured correctly. For more information refer to Configuring an external identity provider.

## Assign groups to your application

**Note**

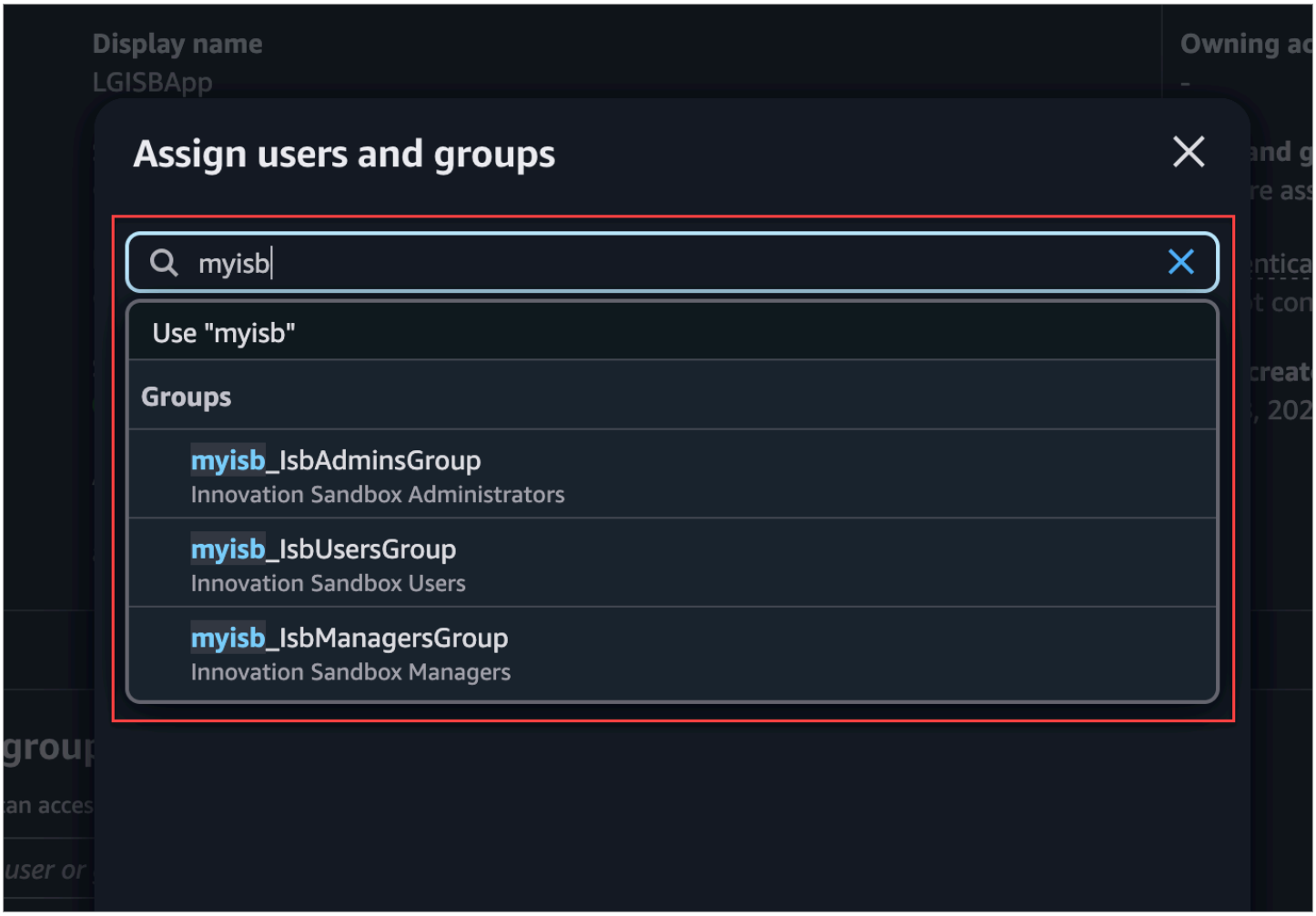
If you have configured your IAM Identity Center instance to use an external identity provider, you will need to manage user groups through that external provider instead of creating them directly in IAM Identity Center.

The IDC stack creates these three user groups in IAM Identity Center (where `NAMESPACE` is the namespace parameter passed to the stack):

- `<NAMESPACE>_IsbUsersGroup`
- `<NAMESPACE>_IsbManagersGroup`
- `<NAMESPACE>_IsbAdminsGroup`

To assign groups to your application:

1. Sign in to the AWS IAM Identity Center console.
2. From the left pane, under **Application assignments**, choose **Applications**.
3. On the Applications page, from the **Customer managed** tab, choose the application you created in the previous steps.
4. Choose **Assigned users and groups**, and choose the three groups. Manually enter the namespace to find the group, as they are not listed by default.




5. Choose **Done** to assign these groups to your application.

## Assign users to groups

As you add new users to IAM Identity Center, you will have to assign them to one of the groups for them to access Innovation Sandbox.

> **Note**
>
> If you have configured IAM Identity Center to use an external identity provider you must assign group access to users through the external identity provider itself and have the changes synced over to your IAM Identity Center instance.

1. Sign in to the AWS IAM Identity Center console.

2. From the left pane, choose **Users**.
3. On the Users page, choose the user name for the user you want to add to a group. The User details page displays.
4. On the **Groups** tab, choose **Add user to groups**.
5. Choose the groups you want to add the user to. You can choose from one of these relevant groups, depending on user role:
   - `<NAMESPACE>_IsbUsersGroup`
   - `<NAMESPACE>_IsbManagersGroup`
   - `<NAMESPACE>_IsbAdminsGroup`
6. Choose **Add user to group**.

Alternatively, you can choose a group and add users to the group.

1. From the left pane, choose **Groups**.
2. On the Groups page, choose the group name you want to add users to. The Group details page displays. You can choose one of these relevant groups:
   - `<NAMESPACE>_IsbUsersGroup`
   - `<NAMESPACE>_IsbManagersGroup`
   - `<NAMESPACE>_IsbAdminsGroup`
3. On the **Users** tab, choose **Add users to group**.
4. Choose the users you want to add to this group.
5. Choose **Add users to group**.

For more information, refer to the Manage identities in IAM Identity Center topic.

# Configure the web application

After setting up SAML 2.0, mapping application attributes, and setting up users and groups, you can configure the web application.

Log in to the AWS account where the solution Hub and data stacks are deployed. Make sure that you are in the correct home Region.

In this section, you will:

- Update configuration using AWS AppConfig
- Update values in AWS Secrets Manager

# Update configuration using AWS AppConfig

In this step, you will collect several configuration values and use them in the authentication configuration section of the solution's GlobalConfig in AWS AppConfig.

## Save the IAM Identity Center application configuration values

1. In the IAM Identity Center console in the account where IAM Identity Center is enabled, navigate to the custom SAML 2.0 application created in the Create a SAML 2.0 application section.
2. On the custom application's page, under **Actions**, choose **Edit configuration**. You do not need to edit anything; however, this page contains the authentication configuration values required by the solution.
3. Save the following values to use in the next step:

| Name | Description |
|---|---|
| IAM Identity Center sign-in URL | Sign-in URL for application authentication |
| IAM Identity Center sign-out URL | Sign-out URL for redirecting to the access portal sign-out page |
| IAM Identity Center Certificate | Certificate that should be downloaded |
| Application SAML audience | SAML audience value that you specified when creating the application |

## Save the IAM Identity Center access portal URL

The IAM Identity Center Access Portal URL is used to provide direct links to access sandbox accounts in the solution UI.

You can locate this value in the IAM Identity Center console in the account where IAM Identity Center is enabled from the **Dashboard** page. This page will contain a **Settings summary** that contains the **AWS access portal URL**. Save this value.

### Save the Web app URL

The Web App URL can be located in the **Hub Account** as an output on the **Compute Stack** in the AWS CloudFormation console. Go to **CloudFormation > Stacks > YourISBComputeStackName** and choose the **Outputs** tab. The Web App URL will be under the output key **CloudFrontDistributionUrl**.

## Updating the global config

After you have collected all the necessary configuration values, you can update the solution's global config with them.

1. Go to the AWS AppConfig console in the **Hub Account**.
2. From the left pane, choose **Applications**.
3. On the Applications page, choose **InnovationSandboxData-Config-Application-XXXXXXX**. The Application details display.
4. Under **Configuration Profiles and Feature Flags**, choose **InnovationSandboxData-Config-GlobalConfigHostedConfiguration-XXXXX** configuration profile, and choose **View details**.
5. Choose **Create version** to begin modifying the current configuration.
6. Set the `maintenanceMode` to `false`. This will allow **manager** and **user** personas to begin to access the solution.
7. In the **auth** section, copy in the corresponding values that you saved in the previous sections (Save the IAM Identity Center application configuration values, Save the IAM Identity Center access portal URL, Save the Web app URL).

```
...
# Authentication Configuration
auth:
  idpSignInUrl: " "
  idpSignOutUrl: " "
  idpAudience: "isb"
  webAppUrl: " "
  awsAccessPortalUrl: " "
  sessionDurationInMinutes: 60
...
```

8. Update the **notification** section. Enter a valid email that can send emails from Amazon Simple Email Service set up in the pre-requisites. If you have not completed this prerequisite step automated email notifications will not be sent.

```
...
# Email Notification controls

notification:
  emailFrom: " "
...
```

9. Choose **Create hosted configuration version**.
10. Choose **Start Deployment**, and choose the latest hosted configuration version you just created.
11. Choose **Start Deployment**.

> **Note**
>
> When updating these configuration values, be mindful of the formatting, white space, and capitalization; otherwise, the solution may not function properly.

## Update values in AWS Secrets Manager

You must sign the SAML requests and responses with SAML certificates to establish trust and verify authenticity. The certificate is created when you create the SAML 2.0 custom application. You will need to configure the solution application with the public key of this certificate.

1. From your AWS console, navigate to AWS Secrets Manager.
2. From the list of secrets, choose the secret named **/InnovationSandbox/<NAMESPACE>/Auth/IDPCert**.
3. On the secret details page, on the **Overview** tab, in the **Secret value** section, choose **Retrieve secret value** and choose **Edit**.
4. Choose **Plaintext**.
5. Copy the value of the IAM Identity Center certificate file (.pem) you downloaded. For more information, refer to the Save application configuration values *Certificate* section.

6. Paste it into the Secrets Manager secret **Plaintext** field and choose **Save**. This will ensure that the application can use SAML authentication.

> **Note**
>
> The Innovation Sandbox on AWS solution is now ready for use. You can now log in to the web UI and start using the solution.

# Using the web UI

This section provides detailed instructions on how to log into the web UI, and use the web UI as an administrator, manager, or a user.

### Administrator Guide

- Adding new accounts to the account pool
- Managing existing accounts
- Registering and managing blueprints
- Viewing or modifying Innovation Sandbox settings

### Manager Guide

- Creating and managing lease templates
- Assigning leases to users
- Approving and rejecting leases
- Choosing the right budget and duration configuration
- Managing leases
- Viewing your lease costs
- Accessing user accounts for troubleshooting

### User Guide

- Requesting a new account lease
- Logging in to an account
- Requesting a lease extension

# Available actions per role

The following table summarizes the actions that can be performed by each Innovation Sandbox role.

### Accounts

| Action | Admin | Manager | User |
|---|---|---|---|
| View all accounts + cost/usage | Yes | No | No |
| Add new AWS accounts to the account pool | Yes | No | No |
| Eject accounts from the account pool | Yes | No | No |
| Retry cleanup process on accounts | Yes | No | No |
| Login to sandbox accounts at any point to troubleshoot or audit | Yes | No | No |

### Blueprints

| Action | Admin | Manager | User |
|---|---|---|---|
| View all blueprints | Yes | Yes | No |
| View blueprint details | Yes | Yes | No |
| Register blueprint | Yes | No | No |
| Update blueprint metadata | Yes | No | No |
| Unregister blueprint | Yes | No | No |

### Lease Templates

| Action | Admin | Manager | User |
|---|---|---|---|
| View all lease templates | Yes | Yes | Yes (public only) |
| View private lease templates | Yes | Yes | No |
| Create lease template (public or private) | Yes | Yes | No |
| Delete lease template | Yes | Yes | No |
| Update lease template (including visibility) | Yes | Yes | No |

**Leases**

| Action | Admin | Manager | User |
|---|---|---|---|
| Request lease (from public templates) | Yes | Yes | Yes |
| Request lease (from private templates) | Yes | Yes | No |
| Assign lease to another user | Yes | Yes | No |
| View all leases | Yes | Yes | No |
| View leases belonging to self | Yes | Yes | Yes |
| Approve lease requests | Yes | Yes | No |

| Action | Admin | Manager | User |
| --- | --- | --- | --- |
| Manually freeze lease | Yes | Yes | No |
| Manually terminate lease | Yes | Yes | No |
| Manually unfreeze lease | Yes | Yes | No |
| Manually extend lease budget/duration or lease | Yes | Yes | No |
| Login to active or frozen sandbox account as manager | Yes | Yes | No |
| Login to active sandbox account as user | Yes | Yes | Yes |

### Settings

| Action | Admin | Manager | User |
| --- | --- | --- | --- |
| View settings page | Yes | No | No |

### Operational

| Action | Admin | Manager | User |
| --- | --- | --- | --- |
| Create managers | Yes | No | No |
| Configure guardrails (such as Service Control Policies) | Yes | No | No |

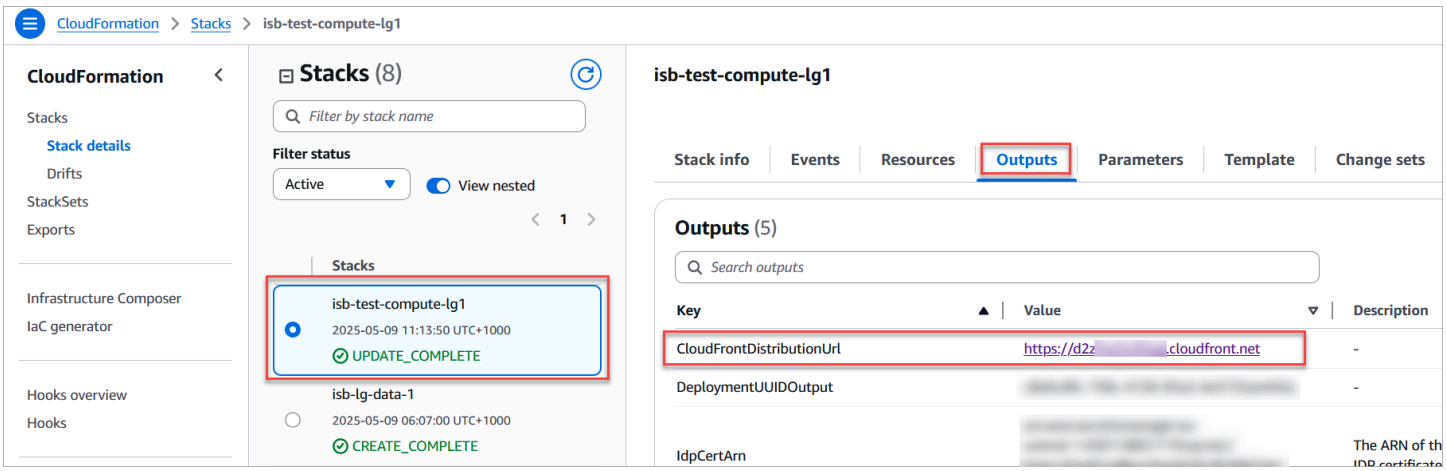
| Action | Admin | Manager | User |
| --- | --- | --- | --- |
| Manage Terms and Conditions content | Yes | No | No |

# Logging into the web UI

> **Important**
>
> Only Admins will have access to the CloudFormation console to retrieve the web UI URL. Admins are responsible for providing the Managers and users with this web UI URL.

After you deploy the Innovation Sandbox on AWS solution:

1. Open AWS CloudFormation console (from the Hub account), and from the left, choose **Stacks**. The list of stacks deployed as part of the solution display.
2. Choose the **Compute** stack to view stack details.
3. On the Stack details page, choose the **Outputs** tab. The web UI URL is the value assigned to the `CloudFrontDistributionUrl` key.




### Web UI URL

4. Choose to open the web UI for the solution. The Sign-in page displays.

The solution uses the Single Sign On service from AWS IAM Identity center for authentication.

# Administrator Guide

This section describes the various actions an Administrator can perform using the web UI.

## Adding new accounts to the account pool

As an Administrator, you can add new AWS accounts to your account pool using the web UI. Adding new accounts will increase the number of accounts you can lease to your end users, allowing them to work with temporary AWS accounts. After you add new accounts to the account pool, you can lease these accounts to users.

> **Important**
>
> You will need to create the AWS accounts and add them to your organization before adding them to the account pool. The Innovation Sandbox solution cannot create new accounts for you.

1. From the AWS Organizations console in your org management account, move accounts that you want to onboard into the **Entry** OU located under the **<NAMESPACE>_InnovationSandboxAccountPool** OU. This will stage them to be registered with the solution.
2. In the solution web UI go to the **Administration** dropdown and choose **Accounts**. This will display the **Accounts** page.
3. From the top right, choose **Add accounts**. The list of available accounts will only include those located in the **Entry** OU.
4. From the list of available accounts, choose the accounts you want to add to the Account pool, and choose **Register**.
5. Review your selections and choose **Submit** to add the selected accounts to your Account pool.

### Resolving Account Cleanup Failures

During the account registration process the sandbox accounts go through an initial cleanup process. In some cases the account may fail cleanup and be placed into the **Quarantine** status.

In most cases the cleanup failure is due to resources created by services that integrate with AWS Organizations that you may have enabled such as AWS CloudTrail, AWS Security Hub, or Amazon GuardDuty.

In the event that the cleanup process fails in your deployment when registering accounts you will need to modify the AWS Nuke configuration file to filter out the protected resources.

First we must discover the resources that should be ignored for your environment:

1. In the **Hub account** navigate to the AWS Step Functions console and choose the account cleaner state machine starting with **AccountCleanerStepFunctionStateMachine**.
2. Choose one of the recent executions with a **Failed** status.
3. From the **Details** tab, copy the executionId provided at the top of the page (It will be in a format like **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx_xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx**).
4. Navigate to the Amazon CloudWatch Logs Insights console.
5. Choose **Saved and sample queries** from the menu on the right.
6. Expand the group name **ISB-<namespace>** and choose the **AccountCleanupLogs** query.
7. In the query editor replace the **PasteStateMachineExecutionIdHere** text with the executionId you copied previously.
8. Ensure that the time range you selected includes the time of the cleanup failure.
9. Choose **Run query**, this will display any resources that failed to be cleaned up.
10. Make a note of any of the resource types in the **resourceType** column that should be filtered out.

Now we must update the AWS Nuke configuration file to ignore these resources:

1. In the **Hub account** navigate to the applications page in the AWS AppConfig console.
2. Choose the application starting with **InnovationSandboxData-Config-Application**.
3. Choose the configuration profile starting with **InnovationSandboxData-Config-NukeConfigHostedConfiguration**. This is the AWS Nuke config file for the solution.
4. To update the configuration file choose **Create version**.
5. Use the resource types noted in the previous steps to modify the filter to ignore them. Refer to the AWS Nuke Config documentation for details on how to update filters.
6. Once you have made your modifications, choose **Create hosted configuration version**.

7. Then choose **Start deployment** to update the nuke configuration for the solution.

Assuming you have appropriately modified the filters for your environment you can now retry the cleanup process:

1. Return to the solution web UI and navigate to the **Accounts** page.
2. Select any accounts that are in the **Quarantine** status.
3. Under the **Actions** menu, choose **Retry cleanup**.

This will reinvoke the cleanup process on the account with the new AWS Nuke configurations.

> **Note**
>
> If the cleanup process continues to fail, you may have missed a resource that needs to be filtered out. Repeat the previous steps to add the appropriate filters to your AWS Nuke config file for other resources failing the cleanup that should be filtered out.

## Account states in Innovation Sandbox

This table explains the various states the account can be in at any given time. Administrators (or anyone) cannot change these states manually.

| State | Description |
|---|---|
| Available | The account is in the pool and ready to be used as part of a lease. |
| Active | The account is being used for a lease. |
| Frozen | The account is being used for a lease but the user no longer has access to the account. Administrators and Managers can still access the account for evaluation and review purposes.<br><br>**Note:** This is an optional state. You will need to configure the account to freeze during the |

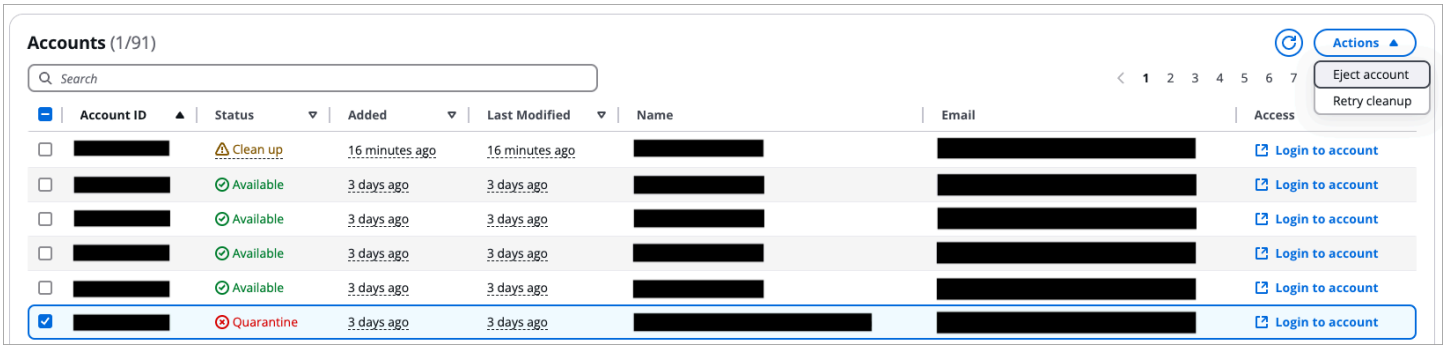
| State | Description |
| --- | --- |
| | lease template creation. See Creating and managing lease templates for more informati on. |
| CleanUp | The account is going through the clean-up process. |
| Quarantine | Accounts that fail to complete the automated clean-up will be quarantined and an Admin will need to manually resolve any resources that failed to delete. After manual remediati on, the account will go back into the clean-up state for a final clean-up process. |

### Account lifecycle in Innovation Sandbox

For more information, refer to the Account lifecycle section.

## Managing existing accounts

As an Administrator, you can manage any existing accounts. This allows you to manually perform account lifecycle actions such as removing accounts from the pool, and retrying the clean-up process.




#### Account management options

To manage accounts:

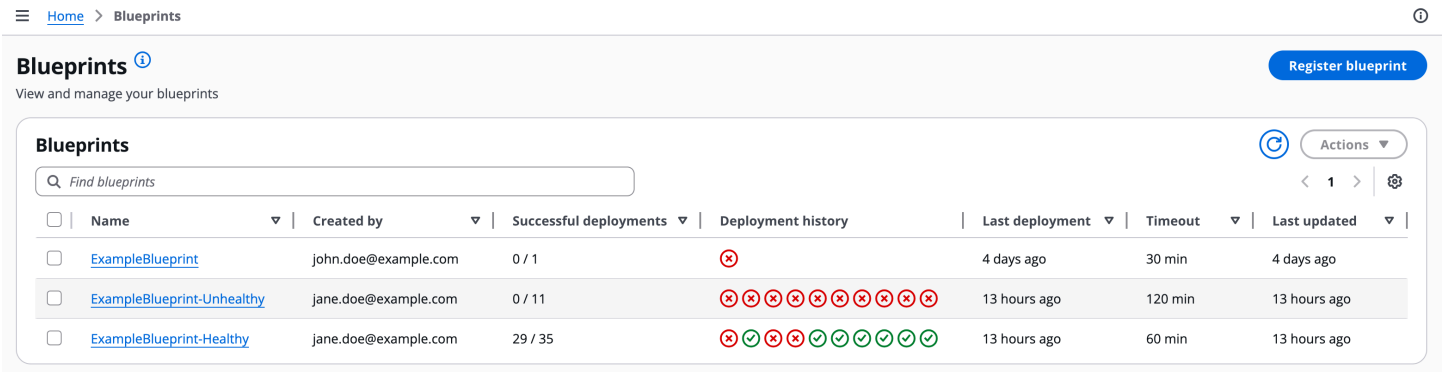
1. From the **Administration** dropdown, navigate to the **Accounts** page.

2. Select the accounts you want to manage to enable the **Actions** dropdown. Using the **Actions** dropdown, you can perform these actions for the selected accounts.

| Action | Description |
| --- | --- |
| Eject account | Removes the account from the pool of available accounts.<br><br>**Note**: Administrators can also eject in-use accounts. For example, they might want to preserve work beyond the lease or move the account away from the management provided by Innovation Sandbox. |
| Retry cleanup | Restarts the clean-up process for that account. By default, lapsed or inactive accounts will be cleaned on a periodic basis. If an account cannot be cleaned, Administrators can manually resolve any issues, and use this option to restart the clean-up process. For example, for accounts in a Quarantine state. |

## Registering and managing blueprints

As an Administrator, you can register CloudFormation StackSets as blueprints to provide pre-configured infrastructure to sandbox accounts. Blueprints enable users to receive accounts with ready-to-use resources, reducing manual setup.



**Blueprints page**

Blueprints are optional. You must create self-managed CloudFormation StackSets outside of Innovation Sandbox before registering them as blueprints.

## Creating self-managed StackSets

Before registering blueprints, create self-managed CloudFormation StackSets in your hub account (the AWS account where Innovation Sandbox is deployed).

**Prerequisites:**

- A CloudFormation template that defines the infrastructure to deploy to sandbox accounts
- IAM roles for cross-account deployment — the solution creates `InnovationSandbox-{NAMESPACE}-IntermediateRole` (administration role) and `InnovationSandbox-{NAMESPACE}-SandboxAccountRole` (execution role)
- Knowledge of your infrastructure requirements and target regions

**Example: Create a StackSet using AWS CLI**

```
aws cloudformation create-stack-set \
  --stack-set-name my-blueprint-stackset \
  --template-url https://s3.us-east-1.amazonaws.com/my-bucket/my-template.yaml \
  --administration-role-arn arn:aws:iam::{ACCOUNT-ID}:role/InnovationSandbox-
{NAMESPACE}-IntermediateRole \
  --execution-role-name InnovationSandbox-{NAMESPACE}-SandboxAccountRole \
  --managed-execution Active=true \
  --capabilities CAPABILITY_IAM \
  --description "My blueprint infrastructure"
```

Replace `{ACCOUNT-ID}` with your hub account ID and `{NAMESPACE}` with your solution namespace.

**Tip**

The `--managed-execution Active=true` flag enables managed execution on your StackSet. This allows CloudFormation to run non-conflicting operations concurrently and

automatically queue conflicting operations, preventing deployment bottlenecks when multiple leases are approved simultaneously.

**Note**

Do not add deployment targets or specify accounts when creating the StackSet. Innovation Sandbox manages StackSet instance creation automatically when leases are approved.

## Blueprint prerequisites

Before registering blueprints, ensure you meet these requirements:

**StackSet Requirements:** * StackSet must use SELF_MANAGED permission model * StackSet must be in ACTIVE status * StackSet template must be valid and tested

**Template Validation:** * Test StackSet deployment to a test account before registration * Verify template completes within your desired timeframe * Ensure template is idempotent (can be deployed multiple times safely) * Validate template works across all target regions

**To validate your StackSet before registration:**

1. Sign in to the CloudFormation console.
2. Navigate to **StackSets** and locate your StackSet.
3. Verify the StackSet status shows **ACTIVE**.
4. Check that the permission model shows **SELF_MANAGED**.
5. Verify the IAM roles match the recommended ISB role names:
   - Administration Role: `InnovationSandbox-{NAMESPACE}-IntermediateRole`
   - Execution Role: `InnovationSandbox-{NAMESPACE}-SandboxAccountRole`
6. Review recent operations to ensure successful deployments.
7. Test deployment to a sandbox account manually before registering as blueprint.

**Note**

For detailed StackSet creation steps, refer to Creating self-managed StackSets.

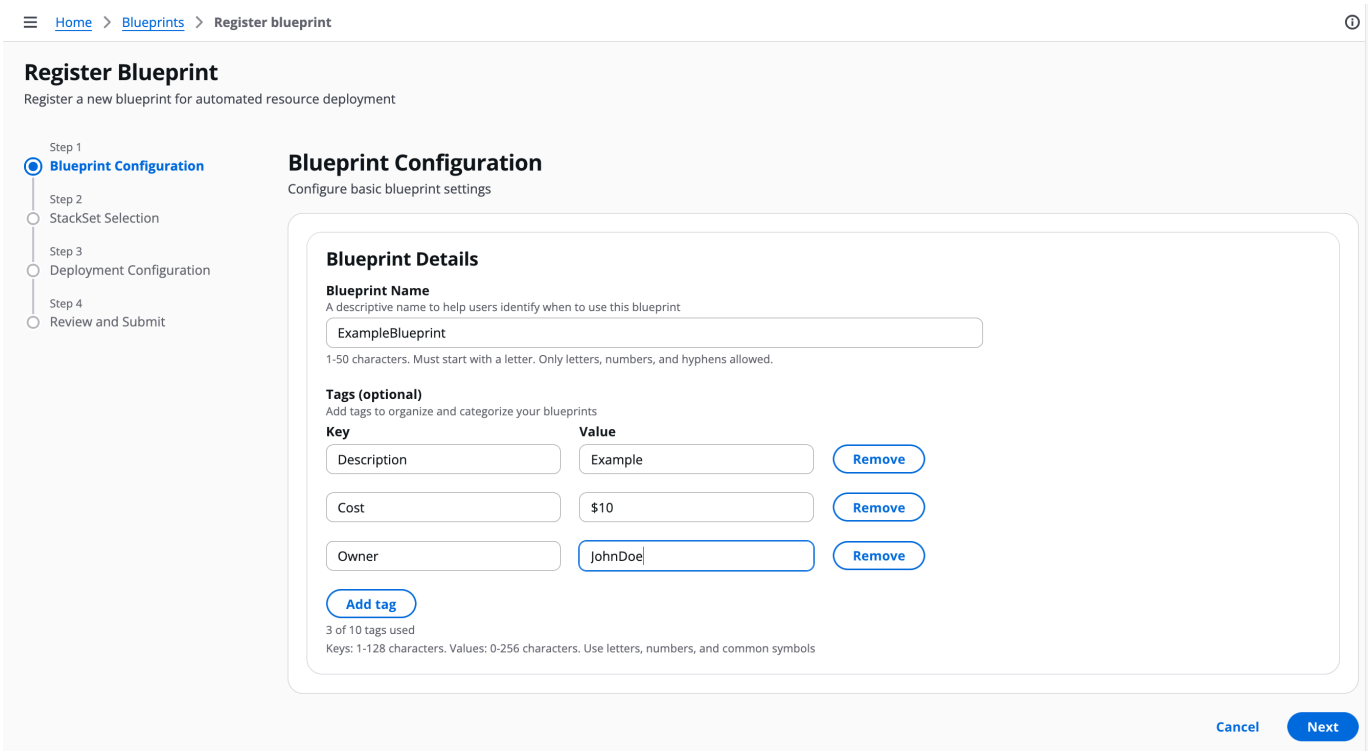
**Important**

Test your StackSet deployment before registering it as a blueprint. Untested StackSets may fail during lease provisioning and prevent users from accessing their accounts. Avoid hardcoding credentials or sensitive data in templates — use AWS Secrets Manager or Parameter Store instead. For general template security guidance, refer to AWS CloudFormation security best practices.

## Registering a new blueprint

To register a blueprint using the registration wizard:

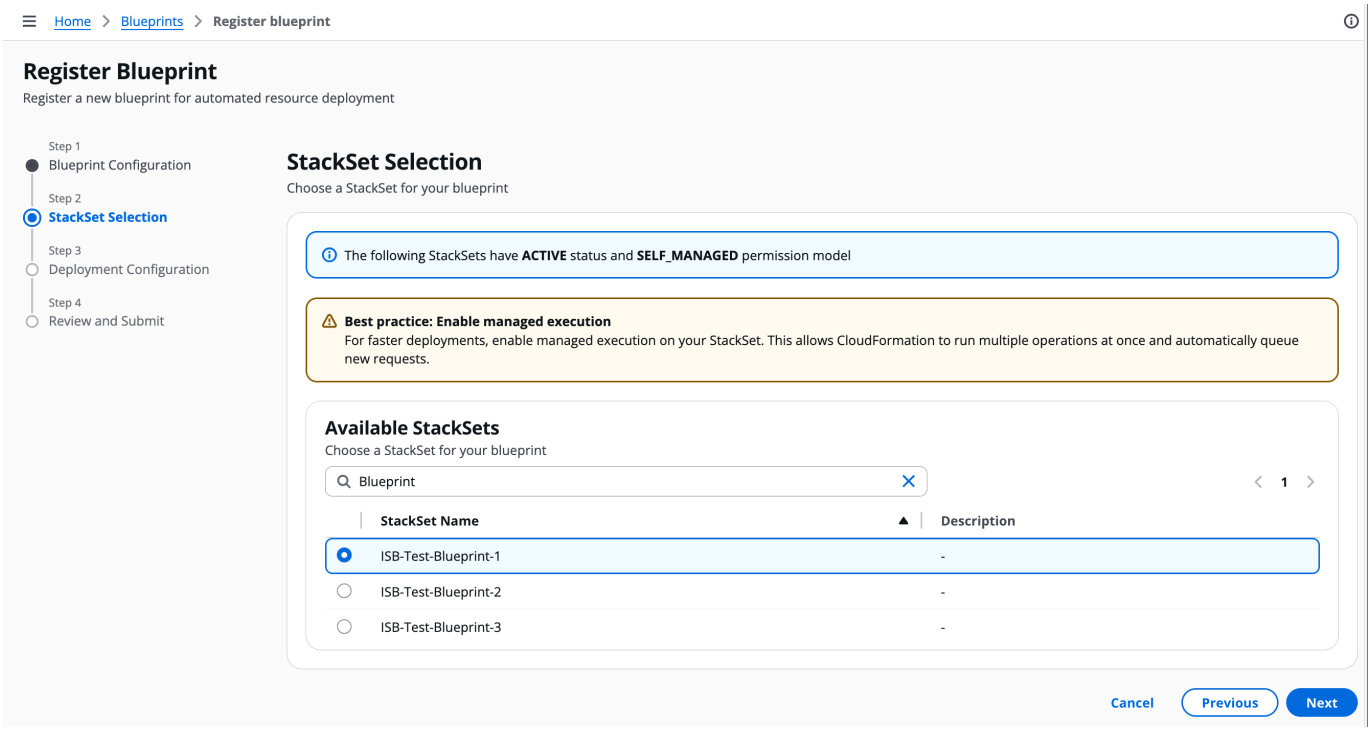
1. From the **Administration** dropdown, choose **Blueprints**.
2. Choose **Register blueprint**.
3. On the **Blueprint Configuration** page, complete the required fields:
   a. For **Name**, enter a descriptive name (1-50 characters).
   b. *(Optional)* Add tags to provide metadata such as estimated cost, description, or support contact.
   c. Choose **Next**.



4. On the **StackSet Selection** page:

   a. Select the StackSet you want to register as a blueprint from the list.

   b. Choose **Next**.




5. On the **Deployment Configuration** page, configure deployment settings:

   a. For **Regions**, select the AWS Regions where the blueprint will be deployed. You can select multiple regions.

> **Note**
>
> The available regions are determined by the **ISB Managed Regions** parameter configured during AccountPool stack deployment. To add or remove regions, update the AccountPool stack parameter. For more information, refer to Deploy the AccountPool stack.

   b. For **Deployment Timeout**, enter the maximum time (in minutes) to wait for deployment completion (default: 30 minutes).

   c. For **Deployment Strategy**, choose a strategy:

| Strategy | Configuration |
| --- | --- |
| Default | Deploys one region at a time with 0% failure tolerance. Safest approach. |
| Custom | Configure each deployment parameter individually. Use when you need specific control over deployment behavior. |

d. If you selected **Custom**, configure these parameters:

- **Region concurrency type**: Sequential (one region at a time) or Parallel (all regions simultaneously)
- **Max concurrent percentage**: Percentage of regions to deploy concurrently
- **Failure tolerance percentage**: Percentage of regions that can fail before the overall deployment is marked as failed. Set to 0% to stop on the first failure.
- **Concurrency mode**: Strict (reduces concurrency on failures) or Soft (maintains maximum concurrency)

  **What success and failure mean in Innovation Sandbox**

  When a lease is approved and the template has a blueprint, Innovation Sandbox deploys the blueprint's StackSet instances to the sandbox account. The deployment outcome determines what happens next:

  - **Deployment succeeds**: The user is granted access to the sandbox account with the pre-deployed resources. The lease becomes **Active**.
  - **Deployment fails**: The lease is terminated, the account is cleaned up and returned to the available pool, and the user does not receive access. If the lease required manual approval, it is reset to **PendingApproval** so the manager can investigate and re-approve. Administrators and managers are notified by email.

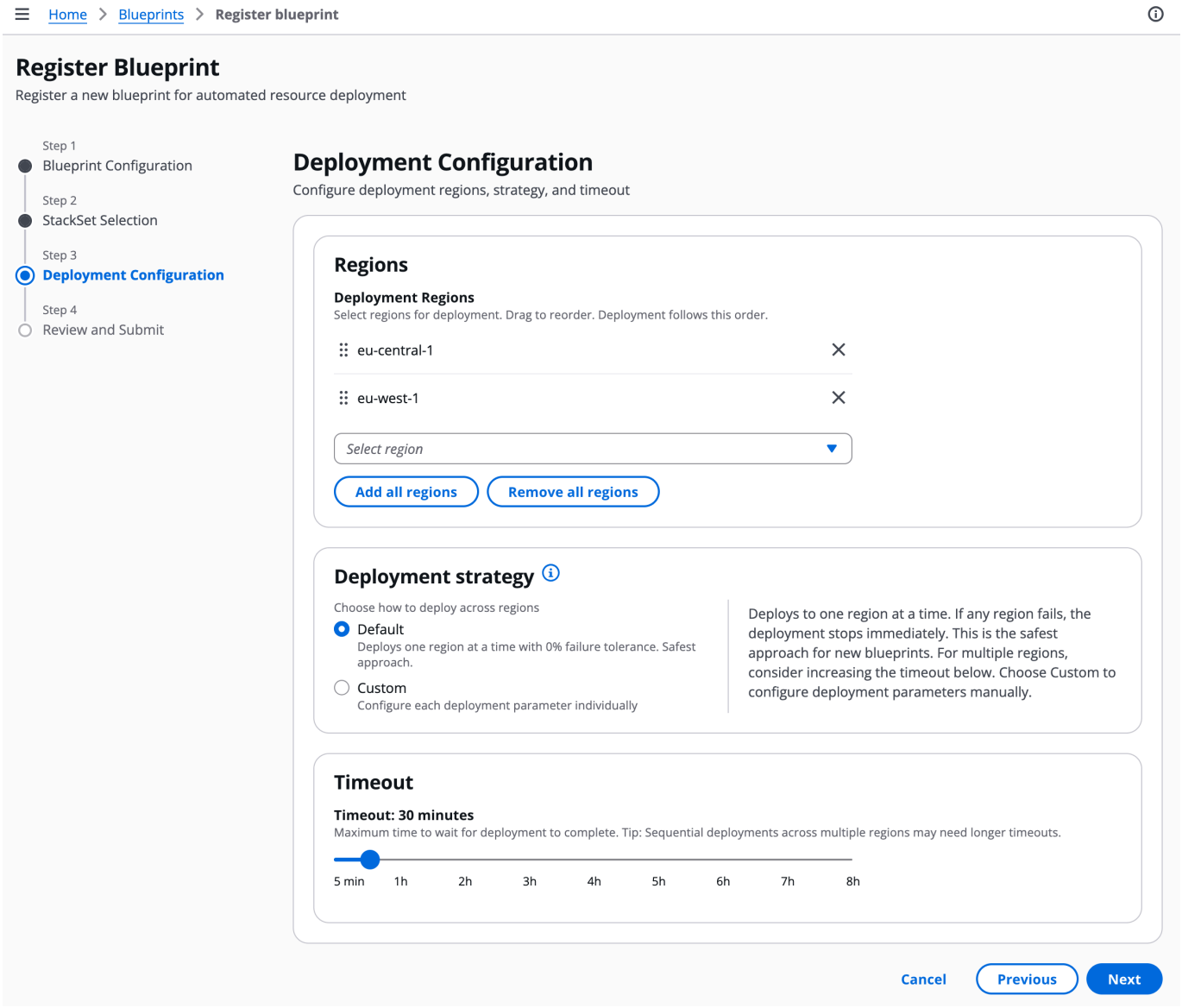
  The failure tolerance percentage controls when CloudFormation considers the overall deployment as failed. If the number of region failures stays within the tolerance, CloudFormation marks the deployment as succeeded, even though some regions have no resources. The user receives the account, but resources may be missing in those failed regions. You can review per-region results in the deployment history on the blueprint details page.

**Example: Blueprint deploys to 3 regions (us-east-1, us-west-2, eu-west-1) with 100% concurrency and 30% failure tolerance**

- **1 of 3 regions fails** (33% failure rate, within 30% tolerance rounded up): CloudFormation marks the deployment as **succeeded**. The user receives the sandbox account and the lease becomes Active. However, the failed region has no blueprint resources. The deployment history on the blueprint details page shows the per-region results.
- **2 of 3 regions fail** (66% failure rate, exceeds 30% tolerance): CloudFormation marks the deployment as **failed**. The lease is terminated, the account is cleaned up, and the user does not receive access. Administrators and managers are notified by email to investigate the blueprint configuration.

> **Tip**
>
> For new blueprints, use the **Default** strategy (0% failure tolerance) until you have validated that the StackSet deploys reliably. Switch to **Custom** with higher failure tolerance only when you understand the trade-off: users may receive accounts with missing resources in failed regions.

e. Choose **Next**.




6. On the **Review and Submit** page, review your configuration and choose **Register blueprint**.

After registration completes, a success message confirms the blueprint is available. Managers can now associate it with lease templates.

**To verify successful registration:**

1. On the **Blueprints** page, confirm the new blueprint appears in the list.
2. Select the blueprint name to view details.
3. Verify all configuration matches your input:

- Blueprint name is correct
- StackSet ID matches the selected StackSet
- Deployment configuration shows your settings
- Tags are present (if configured)

4. Check the health metrics section shows "0 / 0" deployments (no deployments yet).
5. Verify the StackSet details section shows correct IAM roles and regions.

> **Note**
>
> Newly registered blueprints have no deployment history until associated with a lease template and used in a lease. For troubleshooting deployment issues, refer to Blueprint deployment issues in the Troubleshooting chapter.

## Monitoring blueprint health

To view blueprint health and deployment history:

1. From the **Administration** dropdown, choose **Blueprints**.
2. Select a blueprint name to view details.

Home > Blueprints > ExampleBlueprint-Healthy

# ExampleBlueprint-Healthy

## Basic details

Edit

**Name**
ExampleBlueprint-Healthy

**Blueprint ID**
bdf69652-34d6-4fe7-8e9d-61b30803c108

**Created by**
jane.doe@example.com

**Created**
2026-02-15T03:26:06.592Z

**Last updated**
2026-02-24T17:26:29.892Z

## Tags

Edit

| Key | Value |
|---|---|
| test | value |

## Deployment configuration

Edit

**Deployment timeout**
60 minutes

**Region concurrency type**
Sequential (one region at a time)

**Concurrent deployments**
100%

**Failure tolerance**
0%

**Concurrency mode**
Strict: Reduces concurrency as failures occur

## Health metrics

**Total deployments**
35

**Successful deployments**
29 / 35

**Last deployment**
2026-02-24T03:57:27.347Z

## Recent deployments

| Lease ID | Account ID | Status | Started | Duration |
|---|---|---|---|---|
| 00dc212a-95de-44f2-ba04-6731c1b2c909 | 111111111111 | SUCCEEDED | 2/23/2026, 10:56 PM | 1 min |
| e0768ece-7ad4-4d36-8cab-ba92ca069e3d | 222222222222 | SUCCEEDED | 2/23/2026, 10:55 PM | 1 min |
| f3236002-76a4-471b-ab1b-cdc628bc7edd | 333333333333 | SUCCEEDED | 2/23/2026, 10:55 PM | 1 min |
| e8f037df-bd53-4bd9-bbf8-6dfd9a71f31c | 444444444444 | SUCCEEDED | 2/23/2026, 5:00 PM | 1 min |
| a35dd232-d992-43f2-ae94-8ccc6e860744 | 555555555555 | SUCCEEDED | 2/23/2026, 2:48 PM | 1 min |
| 932f073e-68d3-49af-b2a4-916b051ae3e3 | 666666666666 | SUCCEEDED | 2/23/2026, 2:48 PM | 1 min |
| a35dd232-d992-43f2-ae94-8ccc6e860744 | 777777777777 | FAILED | 2/23/2026, 10:51 AM | - |
| 932f073e-68d3-49af-b2a4-916b051ae3e3 | 888888888888 | FAILED | 2/23/2026, 10:51 AM | - |
| 14d17e46-288e-4c75-8453-73fb7d8c58fa | 999999999999 | SUCCEEDED | 2/23/2026, 10:51 AM | 1 min |
| a35dd232-d992-43f2-ae94-8ccc6e860744 | 111111111111 | FAILED | 2/23/2026, 10:31 AM | - |

▶ Deployment details

## StackSet details

**StackSet name**
ISB-Test-Blueprint-2

**StackSet ID**
ISB-Test-Blueprint-2:ebdf07cd-a29b-4f01-a1b8-05d160d6e9e4

**Administration role**
arn:aws:iam::123456789012:role/InnovationSandbox-myisb-IntermediateRole

**Execution role**
InnovationSandbox-myisb-SandboxAccountRole

**Regions**
eu-central-1

**Deployment count**
35

**Successful deployments**
29

**Consecutive failures**
0

### Blueprint details page

The blueprint details page shows:

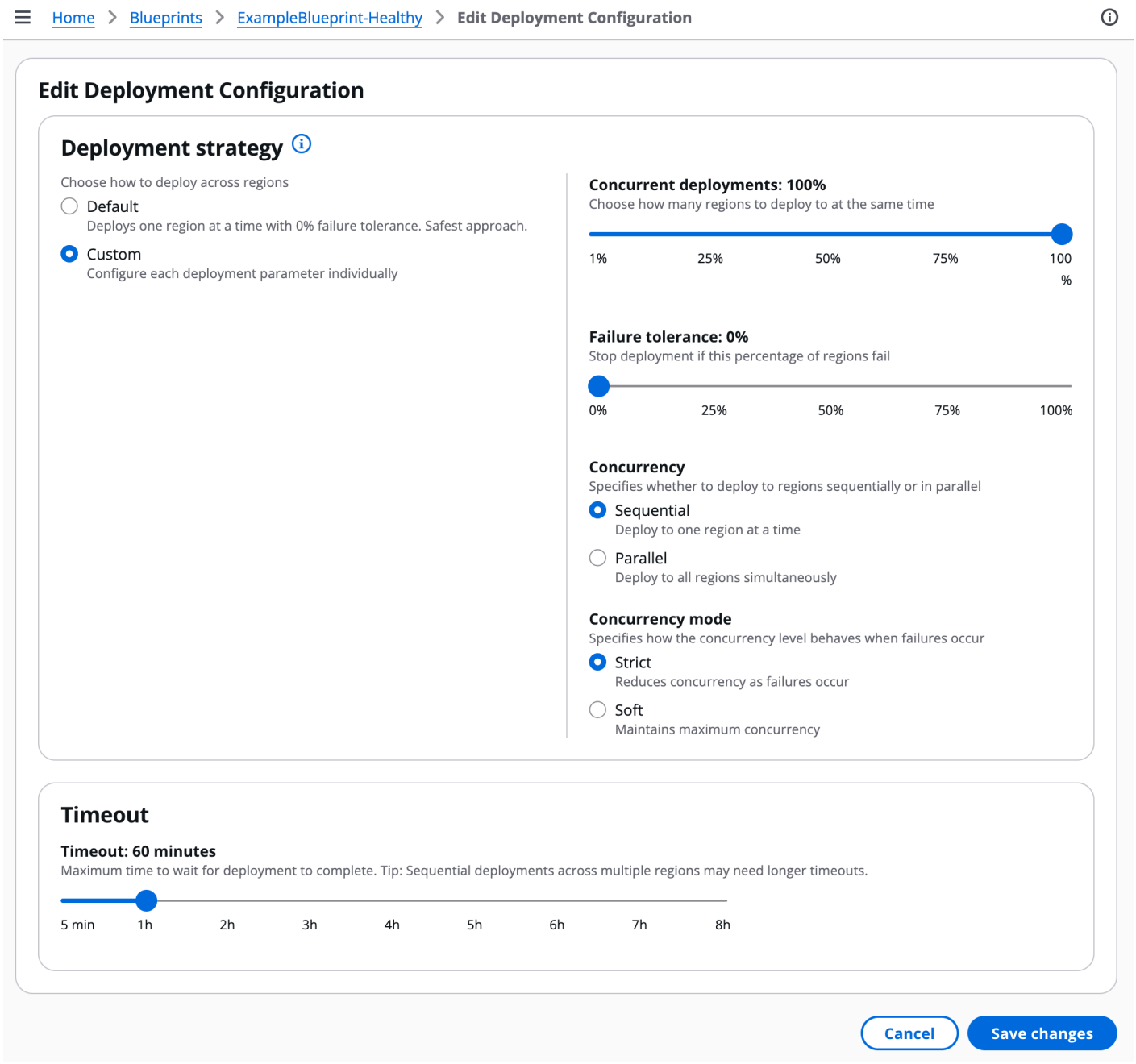
- **Basic details**: Name, blueprint ID, created by, created date, last updated date
- **Tags**: Key-value pairs (if configured)
- **Deployment configuration**: Timeout, region concurrency, concurrent deployments, failure tolerance, concurrency mode
- **Health metrics**: Total deployments, successful deployments (shown as "X / Y"), last deployment time
- **Recent deployments**: Table of recent deployments showing Lease ID, Account ID, Status (RUNNING, SUCCEEDED, FAILED, or QUEUED), Started time, and Duration. Select a deployment row to expand the Deployment details panel showing full details including Operation ID, error type, and error details for failed deployments.
- **StackSet details**: StackSet name, StackSet ID, IAM roles, regions, per-StackSet health metrics

Use this information to assess blueprint reliability and troubleshoot deployment issues.

## Updating blueprint metadata

To update a blueprint:

1. On the **Blueprints** page, select the blueprint name to view details.
2. Choose **Edit** from the section you want to modify:
   - **Basic details**: Update name or tags
   - **Deployment configuration**: Update timeout, deployment strategy, or concurrency settings
3. Make your changes and choose **Save**.



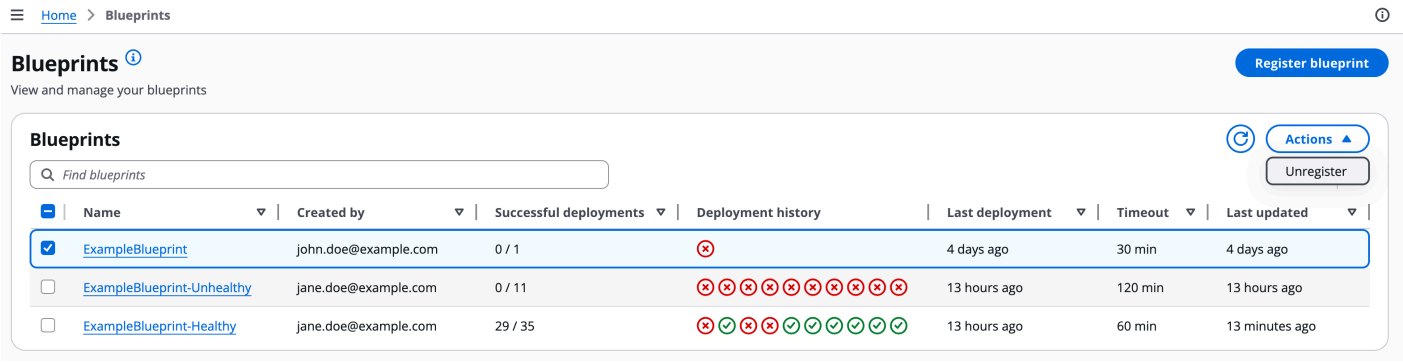

**Note**

Updating blueprint metadata does not affect existing leases. New leases will use the updated configuration.

## Unregistering a blueprint

To unregister a blueprint:

1. On the **Blueprints** page, select one or more blueprints you want to unregister.
2. Under **Actions**, choose **Unregister**.



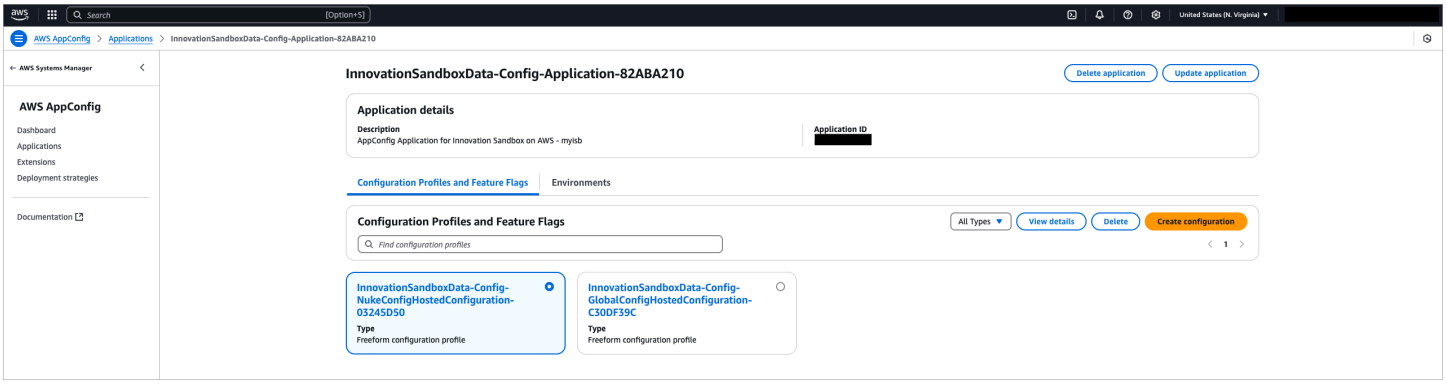

3. Review the blueprints to be unregistered and choose **Submit**.

> **Important**
>
> You cannot unregister blueprints that are associated with active lease templates. Edit the blueprint on all templates and turn off **Enable Blueprint Selection** before unregistering. Unregistering a blueprint removes it from Innovation Sandbox but does not delete the underlying StackSet.

# Viewing or modifying Innovation Sandbox settings

You can view your Innovation Sandbox settings in the **Settings** section of the Administrator dropdown.

To view the current settings, access the AWS AppConfig console in the Hub account, or use the **Settings** section in the web UI.

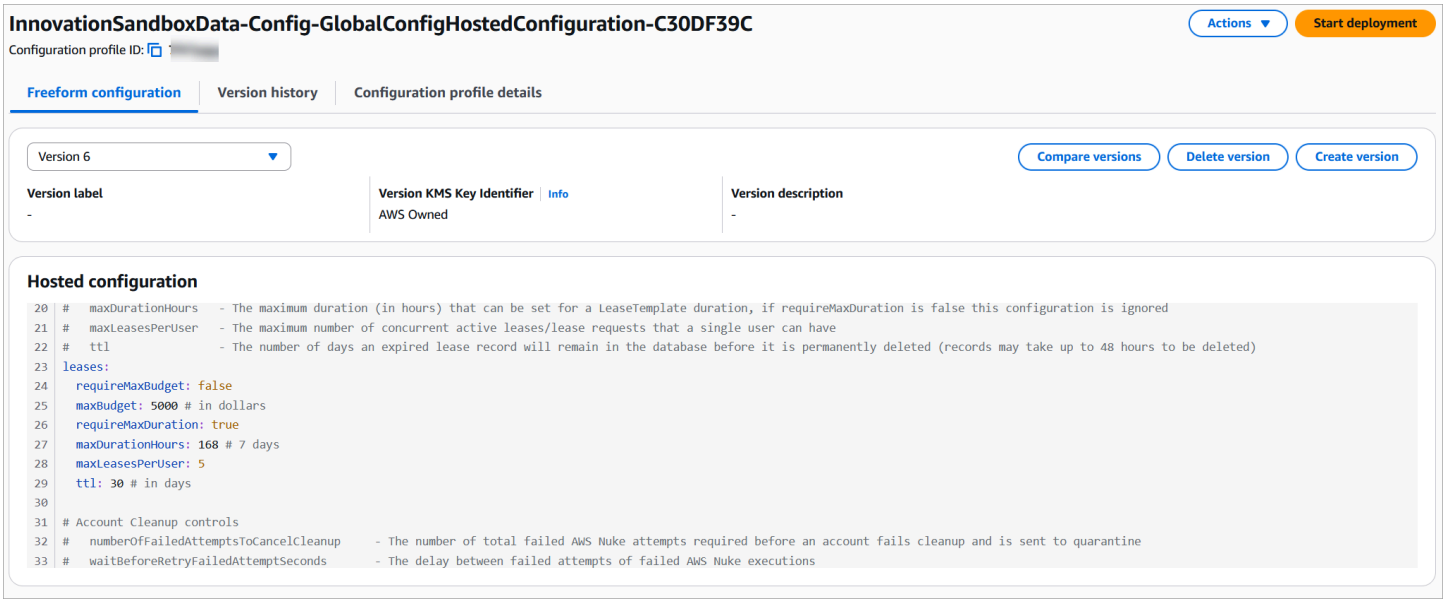



### Innovation Sandbox AppConfig application overview

You **cannot** modify any settings directly using the web UI. To modify these settings, this solution uses AWS AppConfig accessible from within the Hub account.

You can manage these three configuration profiles from the AWS AppConfig console in the Hub account:

- **Nuke configuration**: This configuration determines how AWS Nuke behaves when cleaning your accounts. For more information on AWS Nuke, refer to the AWS Nuke documentation.
- **Global configuration**: This is where you set general settings for your Innovation Sandbox solution. This includes setting the maximum budget and maximum duration for a lease, writing the terms of service and other settings. For more information on these settings, see Global configuration settings.
- **Reporting Configuration**: This configuration defines the allows you to enable cost report groups that can be assigned to lease templates for cost attribution and reporting purposes.




### Configuration profile overview

### Modify configuration

To modify any configuration:

1. Choose the configuration you want to modify, and under the Hosted configuration versions section, choose **Create**. This will open a page where you can modify the configuration file.
2. To update your setting, make your changes and choose **Create hosted configuration version**.
3. To deploy your changes to Innovation Sandbox, choose **Start deployment**. The Deployment details page displays.
4. Under the Deployment details section, keep the **Environment** and **Deployment strategy** parameters set to their default values.
5. Select the version you want to deploy and choose **Start deployment**.

This will create and deploy a new version of your configuration. Note that all hosted configurations are versioned. You can roll back to a previous version by starting a new deployment and selecting a previous version.

> **Note**
>
> After the deployment is successful, you may notice a brief delay as the new settings are deployed to the Innovation Sandbox environment.

## Global configuration settings

The following table includes all of the global configuration settings you can set or modify in Innovation Sandbox.

| Setting | Type | Description |
| --- | --- | --- |
| termsOfService | String | Terms of service that are presented to the user. You can customize this with your own words on how users should responsibly use their sandbox account and what they are responsible for. |
| maintenanceMode | Boolean | If set to true, restricts access of all personas except Admins. This allows Admins to perform sensitive maintenance work like setup, troubleshooting, upgrading, or teardown. |
| leases.maxBudget | Number | The maximum budget that a lease template can be created with. Use this setting to globally enforce that a lease never has a budget over x amount. |
| leases.requiremaxBudget | Boolean | Flag that determines whether or not LeaseTemplates must be created with a maximum budget. |
| leases.maxDurationHours | Number | The maximum duration that a lease template can be created with. This is a way to globally |

| Setting | Type | Description |
| --- | --- | --- |
| | | enforce that a lease never has a duration over x amount. This is measured in hours. |
| leases.maxDurationThresholds | Number | The maximum duration thresholds (in hours). |
| leases.requiremaxDuration | Boolean | Flag that determines whether or not LeaseTemplates must be created with a maximum duration. |
| leases.maxLeasesPerUser | Number | The maximum number of leases one user can hold concurrently. This includes leases pending approval. |
| cleanup.numberOfFailedAttemptsToCancelCleanup | Number | The number of times AWS Nuke will fail before the clean-up process is deemed to have failed. |
| cleanup.waitBeforeRetryFailedAttemptSeconds | Number | The number of seconds to wait between retrying clean-up after a failed attempt |
| cleanup.numberOfSuccessfulAttemptsToFinishCleanup | Number | The number of times AWS Nuke will need to succeed before the clean-up is deemed to be a success. |
| cleanup.waitBeforeRerunSuccessfulAttemptSeconds | Number | The number of seconds to wait between retrying clean-up after a successful attempt. |

| Setting | Type | Description |
| --- | --- | --- |
| notification.emailFrom | String | Email that Amazon SES uses to send email notifications from. |

## Cost report configuration settings

The following table includes the cost report configuration settings you can set or modify in Innovation Sandbox for cost attribution and reporting.

| Setting | Type | Description |
| --- | --- | --- |
| costReportGroups | Array | List of valid cost report group identifiers that can be assigned to lease templates. Maximum of 100 cost report groups, with each group name limited to 50 character s. |
| requireCostReportGroup | Boolean | Flag that determines whether cost report groups are required with leases and lease templates. When enabled, all new lease template creation and updates will require a valid cost report group to be assigned. This will not be enforced for preexistng leases/lease templates and they will need to be manually updated. |

# Manager Guide

This section describes the various actions a Manager can perform using the web UI.

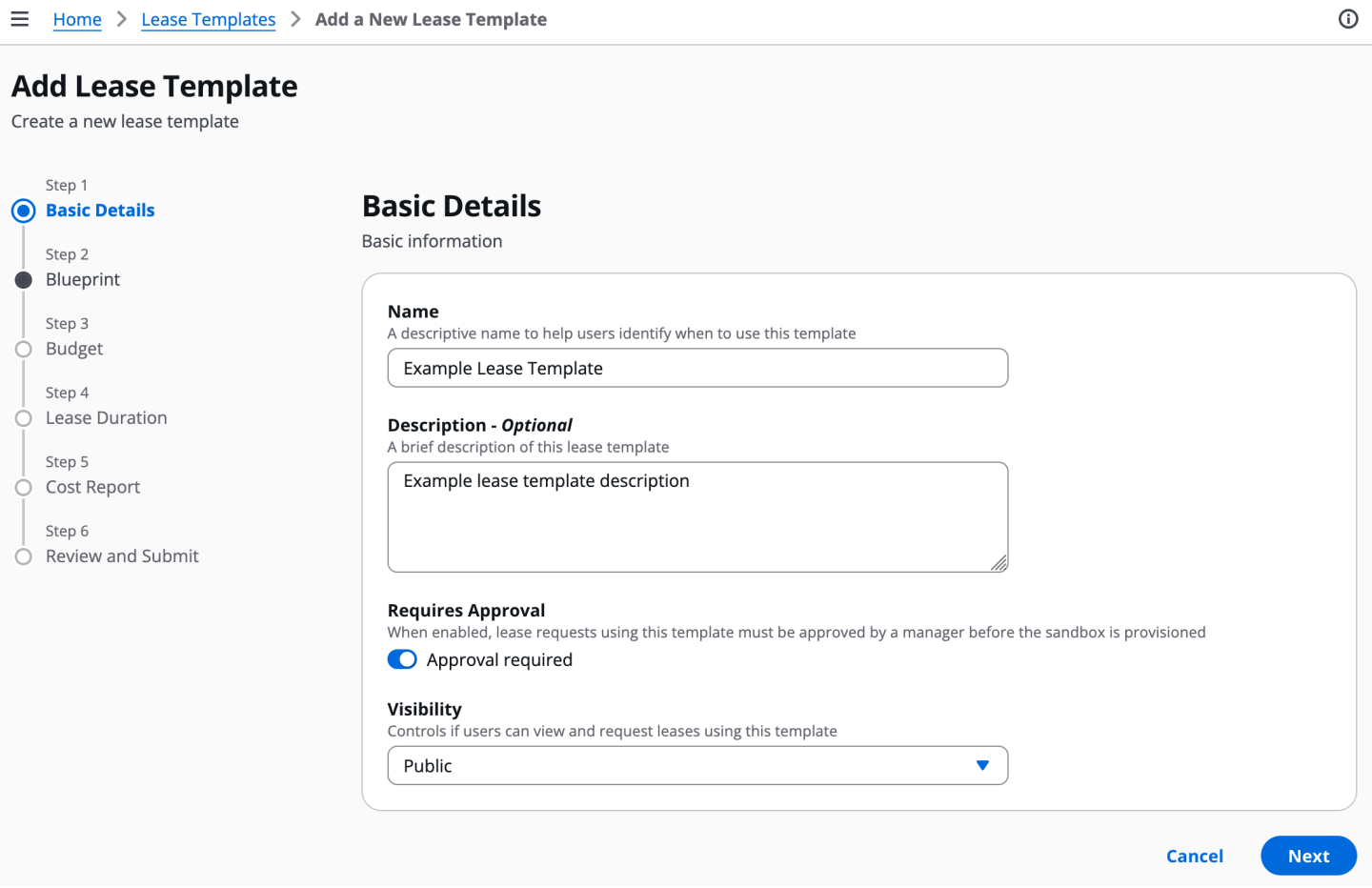
## Creating lease templates

Managers (and Administrators) can create lease templates that define specific configurations users can choose when requesting a lease. A lease template includes settings for blueprints, budget limits, duration, and cost reporting. All of your available lease templates are displayed on the **Lease Templates** page.

To create a lease template, navigate to **Lease Templates** in the web UI and choose **Add new lease template**. This opens a wizard to configure your template.

### Step 1: Basic Details

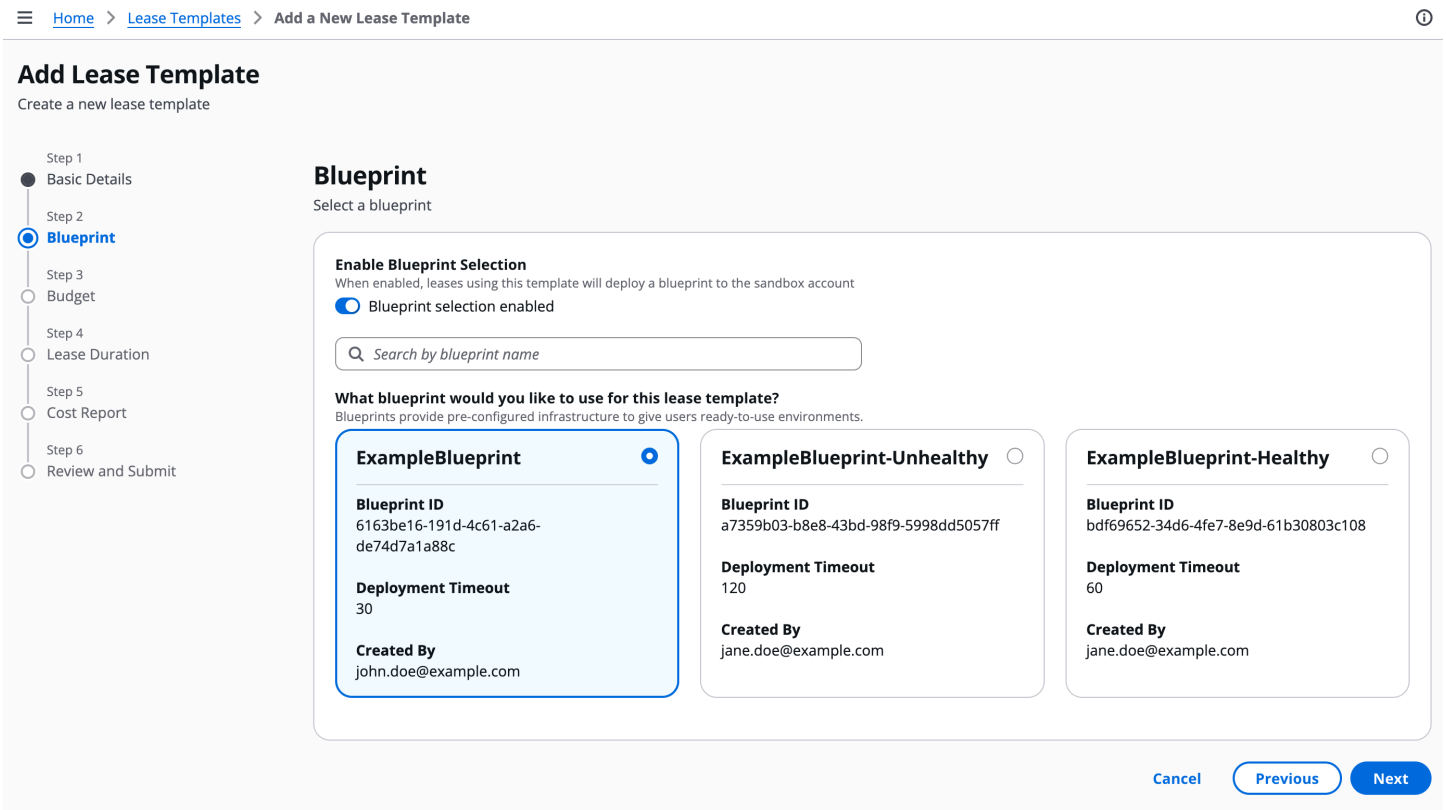
On the **Basic details** page, configure the template's name, description, visibility, and approval requirements.



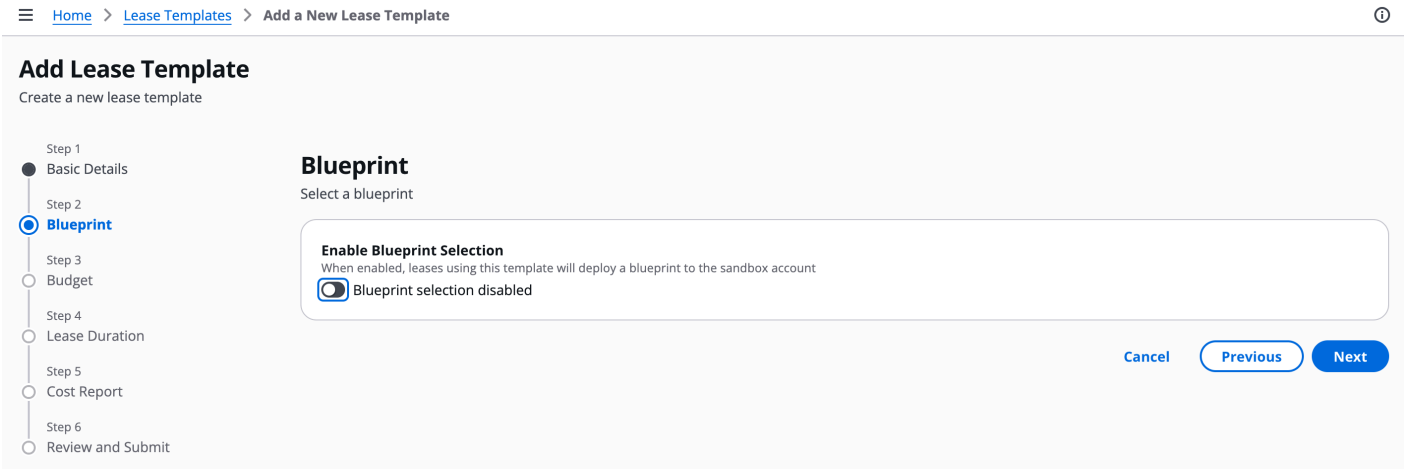
1. For **Name**, enter a descriptive name for your lease template so that you can easily keep track of it.
2. *(Optional)* For **Description**, specify the intended purpose of the account type.
3. For **Requires Approval**, choose whether manager approval is required:
   - **Approval required** (default): Managers must manually approve each lease request. Use this for accounts with high budgets or for experienced users.
   - **No approval required**: Accounts are automatically assigned when requested. Use this for accounts with small budgets, testing, and small workloads.
4. For **Visibility**, choose between **Public** or **Private**:
   - **Public**: The template appears in the general template listing and users can request leases from it through self-service.
   - **Private**: The template is only visible to administrators and managers for direct lease assignment purposes. Users cannot see or request leases from private templates.
5. Choose **Next**.

## Step 2: Blueprint

On the **Blueprint** page, you can associate a blueprint with this lease template to pre-deploy infrastructure when leases are created.




1. Blueprint selection is enabled by default. To skip blueprint selection, turn off **Enable Blueprint Selection**.

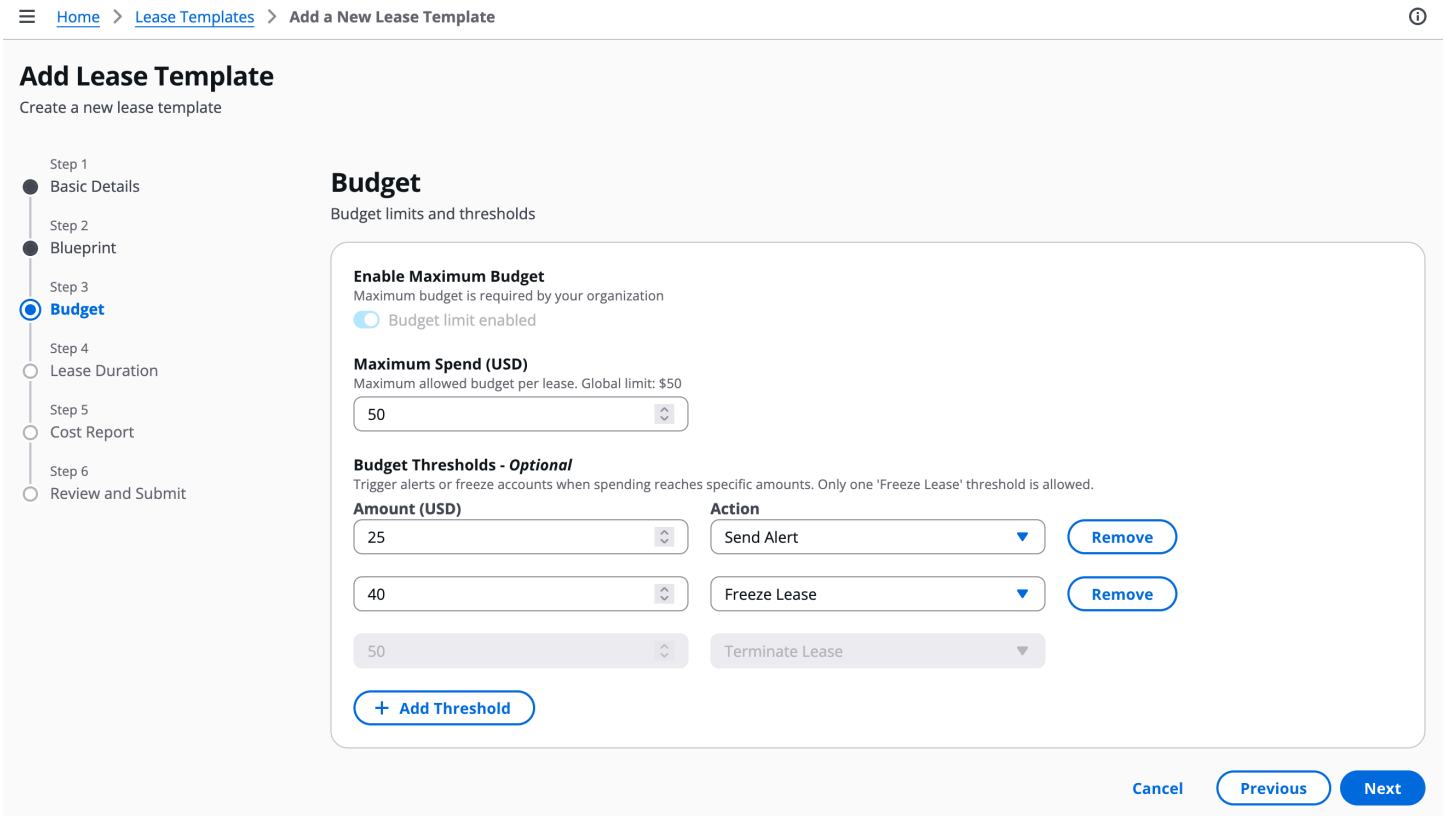



2. Choose a blueprint from the available options.
3. Choose **Next** to continue.

> **Note**
>
> When you approve a lease, the blueprint deploys to the sandbox account. If deployment fails, the lease terminates automatically and the account returns to the pool.

## Step 3: Budget Settings

On the **Budget** page, configure spending limits and budget thresholds. See Budget thresholds for detailed guidance.



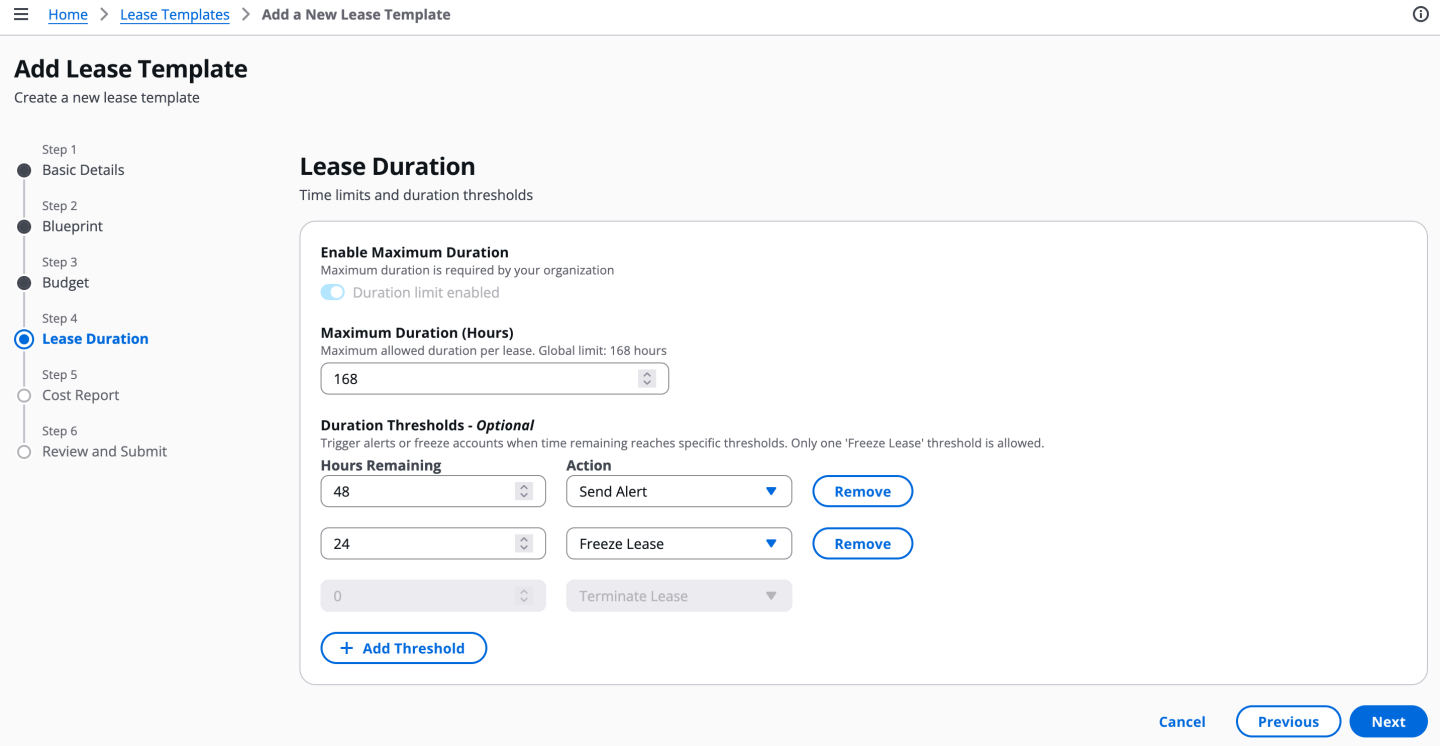

1. Choose whether to set a maximum budget:
   - **Budget limit enabled**: Enter a value in **Maximum Spend** (measured in $USD).
   - **Budget limit disabled**: No spending limit is enforced (not recommended for production use).
2. *(Optional)* Add additional thresholds to send alerts or freeze the account at different spending levels:
   a. Choose **Add Threshold**.
   b. Enter a threshold value in $USD.

c. Select an action: **Send Alert** (email notification) or **Freeze Lease** (prevents new resource creation).

3. Choose **Next**.

## Step 4: Lease Duration

On the **Lease Duration** page, configure time limits and duration thresholds. See Duration thresholds for detailed guidance.

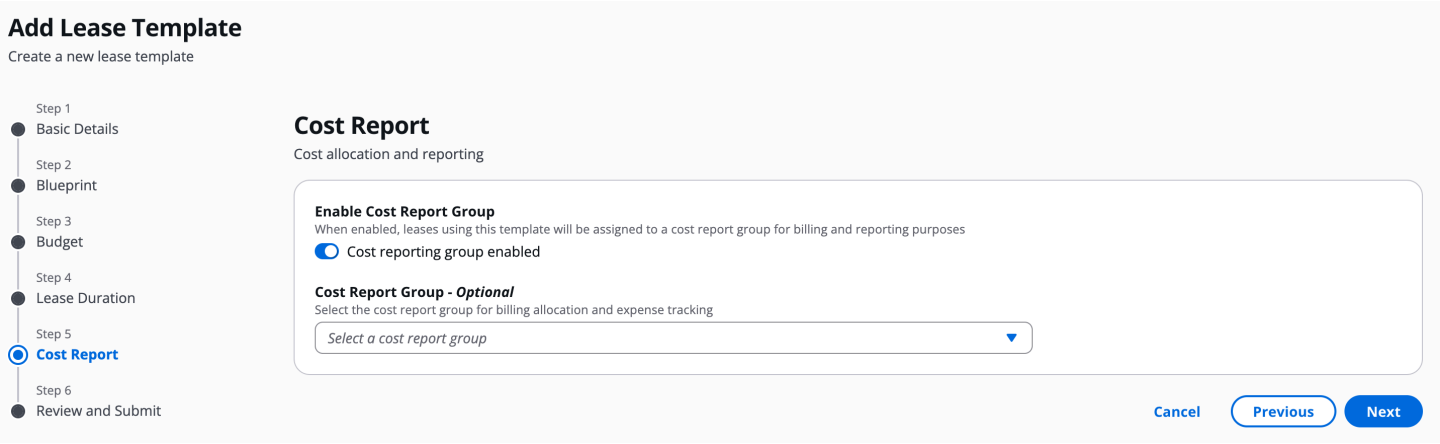



1. Choose whether to set a maximum duration:
   - **Duration limit enabled**: Enter a value in **Maximum Duration (in hours)**. This determines how long the lease remains active.
   - **Duration limit disabled**: Leases do not automatically expire (not recommended for production use).
2. *(Optional)* Add thresholds to send alerts or freeze the account as time remaining decreases:
   a. Choose **Add a threshold**.
   b. Enter a threshold value in hours.
   c. Select an action: **Send Alert** (email notification) or **Freeze Lease** (prevents new resource creation).

3. Choose **Next**.

## Step 5: Cost Report Group

On the **Cost Report Group** page, optionally assign a cost report group to the lease template for cost attribution and reporting purposes.




1. Choose whether to set a cost report group:
   - **Cost reporting group enabled**: You must select a cost reporting group.
   - **Cost reporting group disabled**: No cost reporting group selection required.
2. If cost report groups have been configured by your administrator, select from the available options in the dropdown.
3. Choose **Next**.

> **Note**
>
> Cost report groups are used to generate custom cost reports that are delivered to an S3 bucket for detailed cost tracking and chargeback by department, project, or team. If the administrator has enabled the `requireCostReportGroup` setting, selecting a cost report group will be mandatory.

## Step 6: Review and Submit

On the **Review and Submit** page, review all your settings before creating the template.

1. Review each section of your configuration:

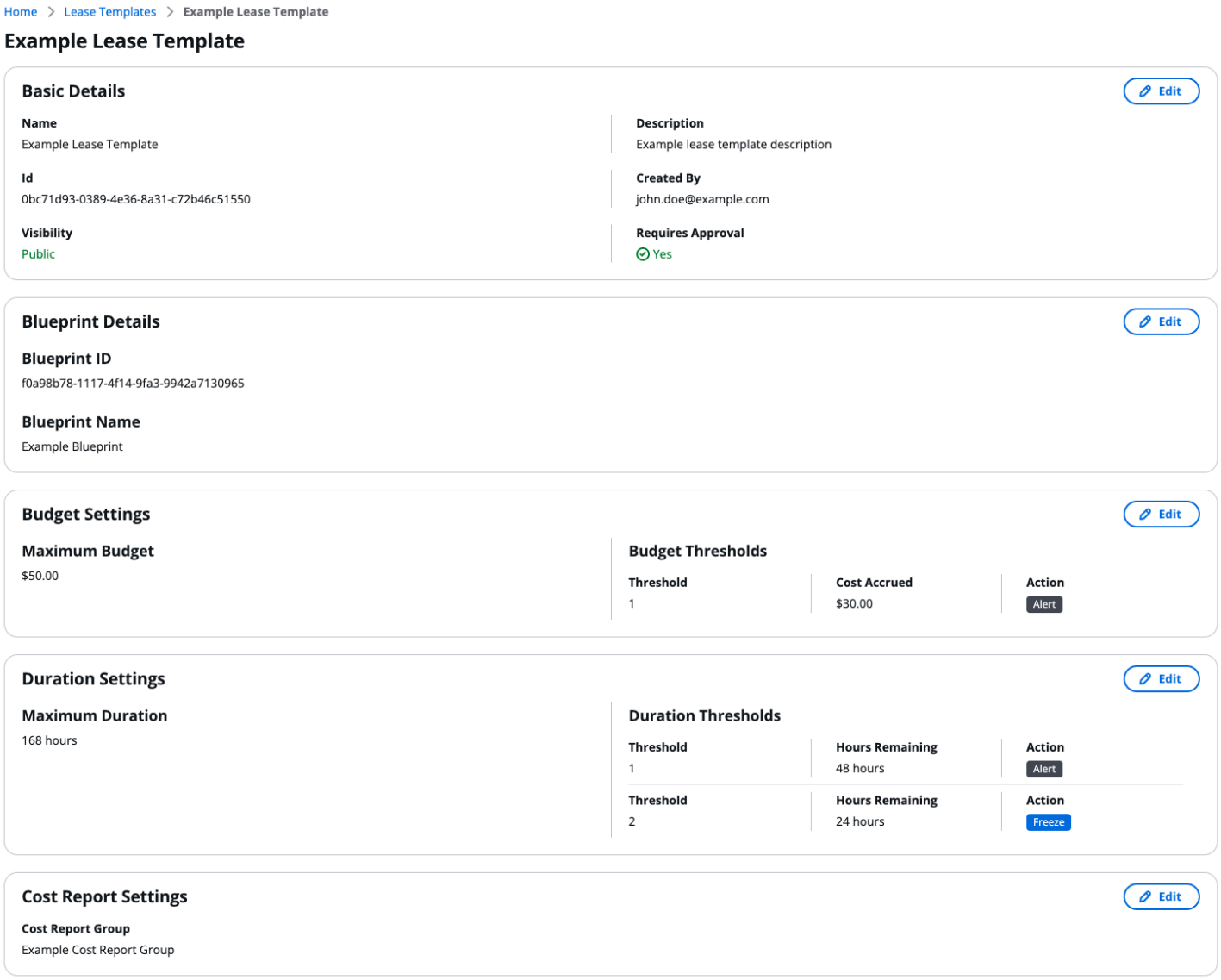
- Basic Details
- Blueprint (if configured)
- Budget Settings
- Lease Duration
- Cost Report Group (if configured)

2. If you need to make changes, use the wizard navigation to return to a previous step.
3. When you're satisfied with the configuration, choose **Create lease template** to create the lease template.

> **Note**
>
> The new lease template will be available for users to request leases (if public) or for managers to assign leases (if private).

## Updating lease templates

After creating a lease template, you can modify its configuration from the lease template details page. Each section of the template can be edited independently.




To update a lease template:

1. On the **Lease Templates** page, select the template name to open the details page.
2. The details page displays all configuration sections with their current settings:
   - **Basic Details**: Name, Description, ID, Created By, Visibility (Public/Private), Requires Approval (Yes/No)
   - **Blueprint Details**: Blueprint ID, Blueprint Name (or "No Blueprint" if not configured)
   - **Budget Settings**: Maximum Budget, Budget Thresholds (with alert and freeze actions)
   - **Duration Settings**: Maximum Duration, Duration Thresholds (with alert and freeze actions)
   - **Cost Report Settings**: Cost Report Group (or "Not assigned" if not configured)

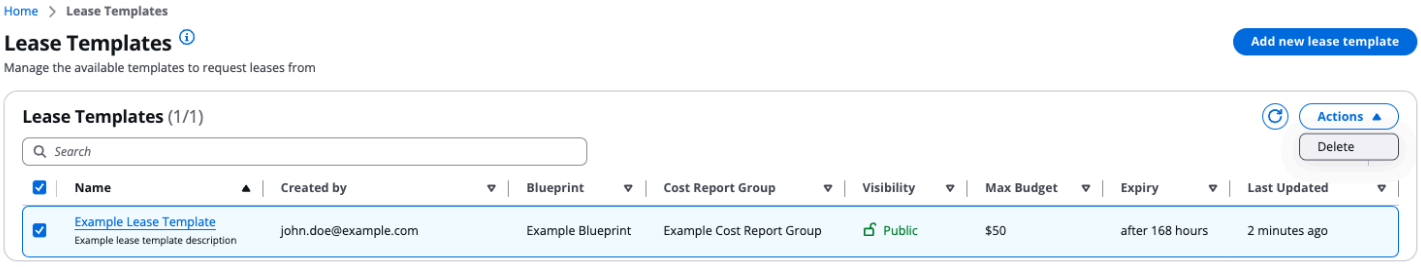
3. Choose the **Edit** button next to the section you want to modify.
4. Make your changes on the edit page for that section.
5. Choose **Save changes** to update the lease template.

To remove a blueprint, choose **Edit** on the Blueprint Details section. Turn off **Enable Blueprint Selection** and choose **Save changes**. New leases created from this template will no longer deploy blueprint infrastructure.

> **Note**
>
> Modifying a lease template will not affect any existing leases with the old configuration. This includes blueprints - existing leases continue using their original blueprint configuration.

## Deleting lease templates

You can delete lease templates that are no longer needed. Deleting a template removes it from the available templates list but does not affect existing leases created from that template.




To delete a lease template:

1. On the **Lease Templates** page, select the lease template you want to delete.
2. This will enable the **Actions** dropdown. Under **Actions**, select **Delete**.
3. Confirm your choice in the confirmation dialog and choose **Delete** to delete the template.

**Note**

Deleting a lease template will not affect any existing leases with the deleted lease template. Existing leases will continue to function normally until they expire or are terminated.

## Assigning leases to users

As a Manager or Administrator, you can create leases directly on behalf of other users without requiring approval. This lease assignment feature is particularly useful for controlled distribution scenarios such as educational workshops, hackathons, or enterprise innovation initiatives where accounts need to be pre-allocated to specific users.

**Important**

The target user must exist in AWS IAM Identity Center before you can assign a lease to them. Users will receive an email notification when a lease is created on their behalf.

To assign a lease to a user:

1. In the web UI, from the left, select **Leases**.
2. Choose **Assign lease**.
3. On the **Assign lease** page, complete the wizard forms:
   a. For **Select lease template**, select from any available lease template (both public and private templates are available for assignment).
   b. For **Select User**, enter the email address of the user you want to assign the lease to. This must match their email address in AWS IAM Identity Center.
   c. For **Terms of Service**, check the box confirming that you accept the terms of service on behalf of the assigned user.
   d. *(Optional)* For **Review & Assign**, add any relevant notes about the lease assignment for audit purposes.
4. Review your settings and choose **Submit** to create the lease assignment.

The lease will be created immediately without requiring approval, and the target user will receive an email notification with details about their new sandbox account access.

### Monitoring assigned leases

Leases you create on behalf of others will show your email address in the "Created by" field, making it easy to track which leases you've assigned. You can view all leases you've created in the **Leases** page, where they will be clearly identified with assignment details.

## Approving and rejecting leases

Certain accounts require approval to be requested for a lease. When a user requests such an account, Managers or Admins need to approve the request for the user to be granted a lease.

1. From the left, select **Approvals** to view your approval requests.
2. Select the request that you would like to approve/reject. You can select multiple requests at the same time.
3. Using the **Actions** dropdown, select either **Approve request(s)** or **Deny request(s)** depending on your use case.
4. On the dialog box asking you to confirm, select **Approve** or **Deny**.

## Choosing the right budget and duration configuration

When creating lease templates, you will be prompted to set budget and duration for the lease as well as thresholds. These thresholds determine the behavior of the lease once a budget or duration is reached. In this section, we will explore in more detail how to set these thresholds and why they are important to your Innovation Sandbox environment by looking at different use cases.

Here are the different actions that can be triggered when a threshold is reached.

| Action | Description |
| --- | --- |
| Send Alert | An alert is sent to the user notifying them that the budget or duration threshold has been reached. |
| Freeze account | The account is set to the Frozen state. The account is being used for a lease but the user |

| Action | Description |
| --- | --- |
| | no longer has access to the account. Administrators and Managers can still access the account for evaluation and review purposes. |
| Terminate account | The clean-up process will start on the account. Note that this action is only available when a maximum budget or duration is set. |

To get started with this guide, follow the instructions in Creating and managing lease templates until you reach the budget section.

## Budget thresholds

The budget configuration determines the spending limit for the account once leased. The thresholds are measured in $USD and actions are triggered when the account spending reaches the threshold value.

**Use case 1: Not setting a budget**

If you select **Do not set a budget**, the lease will not automatically terminate, even if spending exceeds a certain limit. We recommend using this option for experienced users. It is also recommended for these leases to require approval, so you can limit their use. Bear in mind that the lease will terminate if a maximum duration is set.

You can still set thresholds on a lease with no budget. It is encouraged that you do so users can keep track of the lease usage and take action if necessary. The following figure shows an example of a lease with no budget but with thresholds set.

**Setting thresholds with no budget**

In this example, an alert is sent when the budget reaches $100, $500 and $750, and the account is frozen when the budget reaches $1000. Freezing the account prevents further user activity on the account, as any active resources will continue to incur costs. It gives managers time to investigate the spending, if needed. The user can also keep track on the spending using alerts.

**Use case 2: Setting a budget with thresholds**

Choosing to add a budget creates an extra layer of protection around the account once it is leased. Accounts with a budget are wiped automatically when the budget is reached. The right budget for your lease can depend on multiple factors including (but not limited to):

- The type of workloads that will be run on the accounts: For instance, you might want to set a higher budget for accounts that will be used for machine learning workloads.
- The experience of the user: A user with little or no experience with AWS might incur more costs than an experienced user.
- The purpose of the account: Accounts used for testing might have a lower budget than other accounts.

> **Note**
>
> The maximum budget you can set is limited by the maximum budget set in the Global configuration set by the administrator of your Innovation Sandbox environment. See Viewing or modifying Innovation Sandbox settings for more information.

When you set a maximum budget a threshold is automatically created for you. This threshold will wipe the account once that budget is reached.

#### Default threshold when a budget is set

You can also set additional thresholds to send alerts or freeze the account at different budget levels. They can be used to keep track of the spending and take action if necessary.

### Duration thresholds

#### Use case 3: Not setting a duration

Leases with no duration will only terminate if a maximum budget is set, or if manually terminated by a manager or administrator. Hence, it is important to keep this in mind when choosing **Do not set a maximum duration**. In addition, choosing this option will not allow you to set any thresholds. We recommend using leases with no durations, for workloads that are expected to run for an unknown amount of time.

#### Use case 4: Setting a duration with thresholds

The duration configuration determines how long the account is available once leased to a user. The thresholds are measured in hours. It is important to note that the threshold's actions are only triggered when a certain amount of hours is left.

#### Standard duration threshold

In this example, an alert is sent when 5 hours are left on the lease. It gives the user time to save their work if they want. Once the lease terminates, the account goes through the clean-up process.

## Managing leases

As a Manager or Administrator, you can view and manage the status of leases. Leases give users access to a temporary AWS account. Their budget and duration configuration are defined by its corresponding lease template. A lease is assigned to a user and cannot be shared.

You can view all leases on the **Leases** page. Under **Filter options**, you can filter your leases, either by **lease status** (Active, Pending Approval) or **Lease Template** assigned to the lease.

To change lease status:

1. On the Lease page, select a lease from the list of leases.
2. Under **Actions**, choose the appropriate option to **Freeze**, **Terminate**, **Unfreeze** or **Update** a lease.
   - When a lease is frozen, the user can view leases under their accounts, but cannot access the account through the AWS console.
   - When a lease is terminated, the user loses all access to the AWS account and will need to request a new lease.
   - When a lease is unfrozen, the user regains full access to their AWS account and can continue working with their resources. Only frozen leases can be unfrozen.
   - Updating a lease allows you to increase the budget, extend the duration, update thresholds or change the cost report group of the lease.

> **Note**
>
> When updating a lease, you can extend or reduce the budget of the lease. If you reduce the budget and the user has already spent more than the new budget, the account will go

through the clean-up process once Innovation Sandbox detects that the new budget has been reached. The detection process runs once every hour.

**Important**

You cannot reactivate terminated leases.

## Leases states in Innovation Sandbox

This table explains the various states the leases can be in at any given time.

| State | Description |
| --- | --- |
| Active | The lease is actively being used by a sandbox user. |
| Frozen | The lease has been frozen either by reaching a predefined freeze threshold (based on spend or lease duration) or through manual action by an Admin or Manager. Sandbox users will no longer have access to the lease but the account could still have active AWS Resources running in it, that you will be billed for. If you want to preserve the resources in the account, we recommend an Admin review and eject the account out of the account pool. |
| Pending Approval | The lease request is pending approval from an Admin or a Manager. |
| Approval Denied | The lease request has been denied by an Admin or a a Manager. |
| Lease Duration Expired | The lease has reached its predefined maximum lease duration and the resources in the account are being cleaned up. |

| State | Description |
| --- | --- |
| Lease Manually Terminated | The lease has been manually terminated by an admin or a sandbox manager and the resources in the account are being cleaned up. |
| Account Quarantined | The clean up process failed to terminate some of the resources in the account and manual intervention is required by the Admin to complete clean up. We recommend the Admin manually clean up the remaining resources in the account and initiate Retry Cleanup to complete the clean up process. |
| Account Manually Ejected | An Admin has manually ejected the account out of account pool. |

## Viewing your lease costs

As a Manager or Administrator, you can view the costs incurred by the leases. This allows you to keep track of the costs of your leased accounts.

You can view all leases on the **Leases** page. Each lease will display the amount spent on the lease so far under the **Budget** column. If the lease has a fixed budget, you will be shown a progress bar, showing how close the lease is to reaching the budget. All leases will also display the current spent inside the lease.

By default, the **Leases** page will only display the **Active** and **Frozen** leases. If you'd like to see the costs incurred by terminated leases, you can use the **Status** filter.

Administrators with access to the organization's management account can access the AWS Cost Explorer console for full data on spending in their organization.

> **Note**
>
> Cost Explorer refreshes your cost data at least once every 24 hours. For more information, refer to the Analyzing your costs and usage with AWS Cost Explorer page.

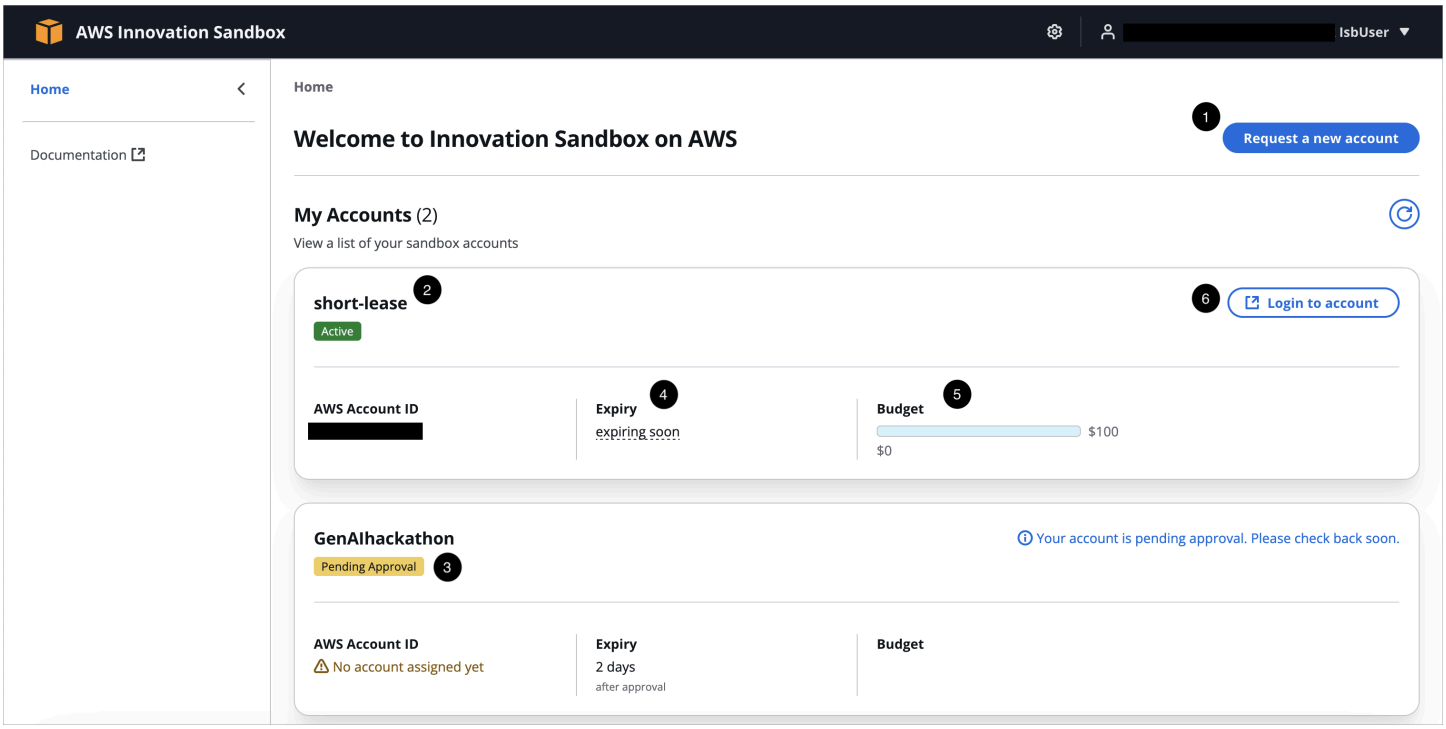
## Accessing user accounts for troubleshooting

Managers or Administrators may need to access a user's AWS account for troubleshooting.

To access a user's account, from the **Leases** page, find the lease corresponding to the account. If the lease is active, the **Login to account** option will be visible under the **Access** column. This will allow you to access the AWS Access portal, where you can log in using one of the available IAM roles.

# User Guide

This section contains all the information regarding actions available to an Innovation Sandbox user. After logging in to the web UI, the following page displays.




### Innovation Sandbox Home Page (User view)

From the home page, you can:

1. Request a new account lease. For more information see, Requesting a new account lease.
2. View all of your current leases.
3. View the current state of your requested leases. In Figure 1, the user has one active lease and one lease pending approval from a manager or administrator.

4. See when your lease expires. You can hover on the status to see the exact date and time.
5. See how much of the allocated budget you have spent.
6. Log in to an account. This is only available if your lease in the **Active** state. For more information see on logging in, Logging in to an account.

## Requesting a new account lease

You can request account leases to gain access to sandbox AWS accounts.

To request an account:

1. After logging in to the web UI, the home page will display all of your current active leases.
2. Choose **Request a new account**.
3. Under **Select lease template**, choose the type of lease you'd like to request. The lease templates are created by your management and administration team.
4. Choose **Next**.
5. In the **Terms of Service** section, read the terms of service and check the box that says *I accept the terms of service*. Ensure that you understand the risks associated with owning a sandbox account lease.
6. Choose **Next** to proceed.
7. Review your choices and choose **Submit**. Optionally, you can add comments describing why you are requesting this account. Note that these comments are visible to the reviewer of the lease (managers or administrators).

If the account type does not require approval, your request is automatically approved and you can access the console by choosing **Login to account**. If it requires approval, your request will be in the **Pending Approval** state until an administrator or manager approves the request.

> **Note**
>
> If you do not see the account under **My Accounts**, you may need to reload the page. Refresh the page to view your account leases.

## Logging in to an account

Once you've requested an account and the lease is in an **Active** state, you can access the AWS account associated with that lease.

1. On the home page, select **Login to account** for the account you want to access. This directs you to the AWS Access portal.
2. In the **Account access details** box, you will find all the available roles that can be used for logging in.
3. Select the role name you want to use. This opens a new page, redirecting you to the AWS console.
4. Alternatively, to retrieve your AWS CLI credentials, choose **Access keys** next to your desired role. This will open a pop-up with instructions for Mac, Linux, Windows and PowerShell environments.

## Requesting a lease extension

If you would like to extend your **Active** lease, contact your Admin or Manager to update your lease to extend lease duration or increase the budget. They will receive a notification and update the lease (subject to availability).

> **Note**
>
> If the lease has already expired, you cannot extend the lease and will need to request a new lease.

# Monitoring the solution

## Overview

The Innovation Sandbox solution includes observability tools for monitoring the solution resources.

## Amazon CloudWatch Application Insights

Innovation Sandbox on AWS includes access to Amazon CloudWatch Application Insights to provide automatic detection and alerting for any errors raised by the solution. When a recurring error is detected within the solution, Application Insights will raise an alarm indicating the potential problem.

Currently, active alarms are displayed in the AWS Cloudwatch Console Dashboard. You can also view an overview of all current and previously detected issues for the solution using the CloudFormation Application Insights console.

CloudWatch Application Insights helps you monitor your applications by identifying and setting up key metrics, logs, and alarms across your application resources and your technology stack. It continuously monitors metrics and logs to detect and correlate anomalies and errors. To assist with troubleshooting, it creates automated dashboards for detected problems, which include correlated metric anomalies and log errors, along with additional insights to identify a potential root cause.

To view the CloudWatch AppInsights dashboard for Innovation Sandbox:

1. Sign in to the CloudWatch console.
2. From the left sidebar, under **Insights**, choose **Application Insights**.
3. Select the **Applications** tab.
4. In the *Find applications* search box, type the solution name to find the dashboard.
5. Select the dashboard, and the application.

The dashboard displays various metrics and logs for your solution.

# Cloudwatch log queries

**Note**

Innovation Sandbox implements a multi-tier log retention strategy optimized for both cost efficiency and audit compliance:

**Retention Tiers**

CloudWatch Logs

90 days (default)

S3 Standard Storage

1 year (default)

S3 Glacier Storage

6 additional years (default)

Total Retention

7 years

**Automatic Archiving**

Logs are automatically exported from CloudWatch to S3 every 7 days, ensuring long-term audit trail availability while minimizing CloudWatch storage costs.

**Operational Benefits**

The 90-day CloudWatch retention provides sufficient time for operational troubleshooting, while the S3/Glacier archiving ensures compliance with typical audit and regulatory requirements.

Innovation Sandbox provides several pre-populated AWS CloudWatch log insights queries that allow you to troubleshoot issues.

To access log insights queries:

1. Sign in to the CloudWatch console.
2. From the left sidebar, under **Logs**, choose **Logs Insights**.

3. On the Logs Insights tab, select **Saved and sample queries**.
4. From the Sample queries, run one of these queries:
   - **LogQuery** — search for all logs related to a specific account, lease, leaseTemplate, or user.
   - **ErrorLogs** — view all recent errors.
   - **AccountCleanupLogs** — view the logs from a specific cleanup execution.

The logs section will display the compute logs for the solution.

## AWS X-Ray

Innovation Sandbox includes access to AWS X-Ray for all critical execution paths. This allows you to troubleshoot any failing workflows and identify where the errors are occurring.

# Update the solution

> **Important**
>
> Always review the solution release notes before updating to understand any changes, new features, or configuration requirements that may affect your deployment.

## Overview

The Innovation Sandbox on AWS solution can be updated to newer versions via two methods:

- Via **CloudFormation templates**, update your existing stacks through the AWS CloudFormation console
- Via **source code (Git)**, pull the latest changes from the Git repository and redeploy

> **Note**
>
> During the update process, you should enable maintenance mode through AWS AppConfig to prevent users and managers from making api requests while the update is in progress.

## Update via CloudFormation templates

If you originally deployed the solution using CloudFormation templates, follow these steps to update to a newer version:

1. Download the latest CloudFormation templates:
   - AccountPool template
   - IDC template
   - Data template
   - Compute template
2. Navigate to the AWS CloudFormation console
3. The stacks should be updated in the following order: AccountPool, IDC, Data, then Compute

4. Choose **Update stack** from the stack actions
5. Select **Replace current template** and specify the new template URL or upload the downloaded template
6. On the **Specify stack detail** page, under **Parameters** review the parameters and modify them as necessary
7. On the **Configure stack options** page, under **Stack failure options** section, ensure **Rollback all stack resources** is selected to automatically revert changes if the update fails
8. Complete the stack update process
9. Repeat for each stack that needs to be updated

> **Note**
>
> If you enabled maintenance mode during the update process, remember to disable it once all updates are complete to restore normal user access.

## Update via source code (Git)

If you originally deployed the solution from the Git repository source code, follow these steps to update to a newer version:

> **Note**
>
> Before updating from source code, ensure your development environment is properly configured by referring to the README for setup instructions.

1. Navigate to your local copy of the Innovation Sandbox on AWS repository
2. Pull the latest changes from the remote repository:

```
git pull origin main
```

3. Review the updated code and configuration files for any changes that may affect your deployment
4. If you made custom modifications, resolve any merge conflicts that may arise

5. Redeploy the solution using npm commands:
   - To deploy all stacks:

```
npm run deploy:all
```

   - Alternatively, stacks can be deployed individually in the following order:

```
npm run deploy:account-pool
npm run deploy:idc
npm run deploy:data
npm run deploy:compute
```

6. Verify that the product's user workflow works as expected after the update

# Troubleshooting

This section provides information about known issues and instructions to mitigate known errors. If these instructions do not resolve your issue, see the Contact AWS Support section to open an AWS Support case for this solution.

## Application sign in error

If you receive an error titled **"Application sign in error"** when accessing the Innovation Sandbox on AWS Web UI, it is likely that the IAM Identity Center application has been misconfigured. Possible root causes include:

- Incorrect **Attribute mappings** in the IAM Identity Center SAML 2.0 application
- Incorrect **Application ACS URL** in the IAM Identity Center SAML 2.0 application
- Incorrect **webAppUrl** in the AWS AppConfig global configuration
- Incorrect **idpSignInUrl** in the AWS AppConfig global configuration
- Missing primary email address in IAM Identity Center for the user attempting to login

Refer to the Post deployment configuration tasks section, and validate that the configurations in both IAM Identity Center and AWS AppConfig are correct.

## Incorrect global configuration

If you receive an error with the message **"Incorrect global configuration"** when accessing the Innovation Sandbox on AWS Web UI, your global configuration in AWS AppConfig has not been updated properly and likely contains invalid fields or values. Review any validation errors included in the error message or logs in the **Compute-ISBLogGroup log group** and make the appropriate corrections to the global configuration. Refer to the Updating the global configuration section for information on how to update the global configuration.

## Authentication failed, Invalid document signature

If you receive an error with the message **"Invalid document signature"** when accessing the Innovation Sandbox on AWS Web UI, the IDP Cert secret was either not updated or an incorrect value was provided. Refer back to the Update values in AWS Secrets Manager section to ensure that the value is correct.

# Authentication failed, SAML assertion audience mismatch

If you receive an error with the message **"SAML assertion audience mismatch"** when accessing the Innovation Sandbox on AWS Web UI, the audience value provided in your IAM Identity Center SAML application configuration does not match the one configured in AWS AppConfig. These values must match for the solution authentication to work properly. Refer back to the Create a SAML 2.0 application and Updating the global config sections for information on how to update these values.

# Investigating accounts in Quarantine state

> **Note**
>
> If the account clean-up mechanism fails to automatically delete resources at the end of an active lease, you might have accounts in a `Quarantine` state. We highly recommend investigating quarantined accounts as quickly as possible, as these accounts can incur costs for resources running inside these accounts.

When the Innovation Sandbox solution is unable to cleanup resources in a sandbox account, the account is moved to a `Quarantine` state and an email is sent to the solution administrators indicating that action should be taken to resolve the account's quarantine status.

To resolve the quarantined status:

1. Log in to the web UI as an Admin, and from the left, under **Administration**, choose **Accounts**.
2. Verify the accounts in `Quarantine` status, and decide whether to clean up the account and return to the account pool, or to eject the account from the solution.
   - To clean up the account and return it to the account pool, choose the account, and under **Actions**, choose **Retry cleanup**.
   - To eject the account, choose the account, and under **Actions**, choose **Eject account**. This moves the account to the **Exit** OU, from which you can manually move it to your desired OU. For more information, refer to the Uninstall the solution section.

If the account is in quarantine because the **retry cleanup failed**, refer to the Resolving cleanup failures section.

# Resolving cleanup failures

If the cleanup process fails to completely clean an account at the end of a lease, Innovation Sandbox will move the account into a Quarantine state, and email the Administrators notifying them of the issue.

To resolve an account that has failed cleanup:

1. Log in to the web UI as an Admin, and from the left, under **Administration**, choose **Accounts**.
2. Confirm the account that has failed the cleanup process. You will need this to view log information in the AWS Console.
3. Log in to the AWS Console using the Hub account, and navigate to the **CloudWatch > Logs Insights** page.
4. From the right pane, under Sample queries, choose the ISB group, and from the dropdown, choose the `AccountCleanupLogs` saved query, and choose **Apply**.
5. In the query window, choose a time frame that includes when the account was last cleaned up (for example: last 3 days) and paste the 'Last Cleanup ReferenceID' into the indicated section.
6. Choose **Run query** to see related events. The log information is displayed under the *Logs* tab.
7. To manually handle any deletion failures in the affected account, navigate back to the **Accounts** page, and log in to the account using the **Login to account** option.
8. After you have manually handled all errors, to restart the cleanup process in the web UI, choose the account and under **Actions**, choose **Retry cleanup**.

# Viewing a specific Lease history

1. Log in to the web UI as an Admin, and from the left, under **Administration**, choose **Leases**.
2. Choose the lease name to view lease details.
3. Copy the LeaseID from the Lease Summary page. You will need this to view lease history in the AWS Console.
4. Log in to the AWS Console, and navigate to the **CloudWatch > Logs Insights** page.
5. From the right pane, under Sample queries, choose the ISB group, and from the dropdown, choose `LogQuery` saved query and choose **Apply**.
6. In the query window, choose the time frame to view logs for and paste the LeaseID into the indicated section.

7. Choose **Run query** to view logs related to the LeaseID provided for the selected time frame. The log information is displayed under the *Logs* tab.

# Viewing a specific User history

1. Log in to the web UI as an Admin, and from the left, under **Administration**, choose **Accounts**.
2. From the Accounts page, confirm the user email address you want to view history for. You will need this to view user/account history in the AWS Console.
3. Log in to the AWS Console, and navigate to the **CloudWatch > Logs Insights** page.
4. From the right pane, under Sample queries, choose the ISB group, and from the dropdown, choose `LogQuery` saved query and choose **Apply**.
5. In the query window, choose the time frame to view logs for and paste the email address into the indicated section.
6. Choose **Run query** to view logs related to the email address provided for the selected time frame. The log information is displayed under the *Logs* tab.

## 403 Permissions error

If you find an issue within the Identity Center:

- Your session might have timed out. Refresh your browser to resolve this.
- Maintenance mode is enabled and you are signed in using a Manager or User role. You will need to contact your Admin to disable maintenance mode in AWS AppConfig. For more information, refer to the Maintenance mode section.

## Unexpected server errors

If you find unexpected server errors while using the web UI, you can trace the issue by using AWS X-Ray.

1. Copy the X-Ray trace ID from the error:
   - When an unexpected error occurs in the web UI, a trace ID will be provided.
   - Or, for any error logs found in Amazon CloudWatch, expand the log to find the X-Ray trace ID for the operation.

2. In the AWS Console, navigate to the AWS X-Ray page and paste the trace ID into the search box.

For more information, refer to the AWS X-Ray Traces, and AWS X-Ray Common Errors pages.

# Lease assignment issues

This section provides troubleshooting guidance for common issues related to the lease assignment feature.

## User not found error

If you receive an error stating that the target user cannot be found when attempting to assign a lease:

**Root causes:**

- The user does not exist in AWS IAM Identity Center
- The email address was entered incorrectly
- The user exists but their email attribute is not properly configured

**Resolution steps:**

1. Verify the user exists in AWS IAM Identity Center:
   a. Navigate to the AWS IAM Identity Center console in your hub account
   b. Go to **Users** and search for the target user
   c. Confirm the user's email address matches exactly what you entered
2. If the user doesn't exist, create the user in IAM Identity Center before attempting lease assignment
3. If the user exists but the email doesn't match, update either the user's email in IAM Identity Center or use the correct email address in the lease assignment

# Unfreezing a lease

When a lease is frozen due to reaching a budget or duration threshold, simply unfreezing the lease without addressing the underlying threshold condition will result in the lease being automatically refrozen when the monitoring system runs its next check.

To successfully unfreeze a lease that was frozen due to threshold conditions:

1. **Update the lease parameters first**: Before unfreezing the lease, you must update either the budget limit, duration limit, or both, depending on which threshold caused the freeze.
   - If frozen due to budget threshold: Increase the maximum budget amount for the lease
   - If frozen due to duration threshold: Extend the lease duration
   - You can update both if needed
2. **Unfreeze the lease**: After updating the appropriate thresholds, you can then unfreeze the lease through the web UI.
3. **Verify the changes**: Confirm that the updated budget or duration limits are sufficient to prevent immediate refreezing.

# Group Cost reporting issues

If the group cost reporting function fails to generate reports, you can manually re-run the function to resolve the issue.

## Re-running the group cost reporting function

To manually invoke the group cost reporting function:

1. Log in to the AWS Console using the Hub account.
2. Navigate to the **AWS Lambda** service.
3. Locate and select the group cost reporting Lambda function.
4. Choose **Test** to create a new test event
5. To run the cost reporting for the previous month, use an empty test event `{}`.
6. Choose **Test** to execute the function.

## Running cost reporting for a previous month

To generate cost reports for a specific previous month:

1. Follow steps 1-4 from the previous section.
2. In the test event configuration, use the following JSON format:

```
{
  "detail": {
    "reportMonth": "yyyy-MM"
  }
}
```

Replace `yyyy-MM` with the desired year and month (for example, `2024-03` for March 2024).

3. Choose **Test** to execute the function for the specified month.

> **Note**
>
> The cost reporting function will process billing data for the specified month and generate reports accordingly. Ensure that billing data is available for the requested month before running the function.

# Blueprint deployment issues

This section covers common issues related to blueprint deployment and resolution steps.

## StackSet not found error

If a lease fails with status **ProvisioningFailed** and logs show `StackSetNotFoundException`, the blueprint's associated StackSet no longer exists or you cannot access it.

The solution stores the composite StackSet ID (format: `stacksetname:uuid`) at registration time. This ID is unique and immutable, so if the original StackSet is deleted and a new one is created with the same name, the blueprint will not resolve to the new StackSet.

**Possible root causes:**

- StackSet was deleted after blueprint registration
- StackSet permissions changed

**Impact on existing leases:**

Only new lease deployments are affected. Active leases that already have deployed stack instances continue to function normally. When those leases terminate, the solution attempts stack instance cleanup on a best-effort basis and proceeds with normal account cleanup via AWS Nuke regardless of the cleanup result.

**To resolve this issue:**

1. Sign in to the CloudFormation console in your hub account.
2. Verify that the StackSet exists and is active.
3. If the StackSet was deleted:
   a. Navigate to the Innovation Sandbox web UI.
   b. From **Administration**, choose **Blueprints**.
   c. Unregister the affected blueprint.
   d. If needed, create a new StackSet and register it as a new blueprint. For instructions, refer to Creating self-managed StackSets and Registering a new blueprint.
4. If the StackSet exists but is inaccessible, verify IAM permissions for the IntermediateRole.

## Blueprint deployment timeout

A deployment with error type `DeploymentTimeout` means the StackSet deployment took longer than the configured timeout (default: 30 minutes). The lease transitions to **ProvisioningFailed** and the account is automatically cleaned up, including any resources deployed by the blueprint.

**To resolve this issue:**

1. Review the failed deployment in the blueprint's **Recent deployments** table to confirm the timeout duration.
2. To prevent future timeouts, increase the deployment timeout in the blueprint's **Deployment configuration** section. For details, refer to the section called "Updating blueprint metadata".

## Concurrent deployment failure (OperationInProgressException)

If a blueprint deployment fails with error type `OperationInProgressException` and message "Another operation is in progress on this StackSet", this means multiple leases using the same blueprint were approved at the same time and managed execution is not enabled on the StackSet. Without managed execution, CloudFormation rejects concurrent operations on the same StackSet.

**Impact:**

- Innovation Sandbox terminates the affected lease (auto-approved) or resets it to PendingApproval (manual approval) so the manager can re-approve.
- The failed deployment appears in the blueprint's deployment history with error type `OperationInProgressException`.

**To resolve:**

Enable managed execution on your StackSet to allow CloudFormation to queue concurrent operations automatically:

```
aws cloudformation update-stack-set \
  --stack-set-name <STACKSET_NAME> \
  --managed-execution Active=true \
  --administration-role-arn <ADMIN_ROLE_ARN> \
  --execution-role-name <EXECUTION_ROLE_NAME> \
  --use-previous-template
```

For more information, refer to Managed execution in the AWS CloudFormation API Reference.

> **Tip**
>
> To avoid this issue when creating new StackSets, include the `--managed-execution Active=true` flag. For instructions, refer to Creating self-managed StackSets in the Administrator Guide.

## Blueprint permission errors

If blueprint operations fail with permission errors, verify IAM role configuration.

**To resolve:**

1. Verify the StackSet uses the self-managed permission model. Service-managed StackSets are not supported.
2. Verify the IAM roles have the required CloudFormation permissions:
   - `cloudformation:CreateStackInstances`

- `cloudformation:DescribeStackSetOperation`
- `cloudformation:DeleteStackInstances`

3. If you are using custom IAM roles (not the recommended ISB roles), ensure they have equivalent permissions. The recommended roles are:
   - Administration Role: `arn:aws:iam::{ACCOUNT-ID}:role/InnovationSandbox-{NAMESPACE}-IntermediateRole`
   - Execution Role: `InnovationSandbox-{NAMESPACE}-SandboxAccountRole`

> **Note**
>
> Custom IAM roles are supported. The solution logs a warning if non-recommended roles are detected but does not block registration or deployment.

## StackSet not appearing in the Innovation Sandbox application

If your StackSet does not appear in the StackSet selection list during blueprint registration, verify the following:

**Possible root causes:**

- The StackSet was created in a different AWS account than the Innovation Sandbox hub account
- The StackSet uses the SERVICE_MANAGED permission model instead of SELF_MANAGED
- The StackSet is in a non-ACTIVE status

**To resolve:**

1. Verify the StackSet exists in the same AWS account where Innovation Sandbox is deployed (the hub account).
2. In the CloudFormation console, navigate to **StackSets** and confirm:
   a. The StackSet status is **ACTIVE**.
   b. The permission model is **SELF_MANAGED**. Service-managed StackSets are not supported.
3. If the StackSet was created in a different account, recreate it in the hub account. For instructions, refer to Creating self-managed StackSets.

# Contact AWS Support

If you have AWS Business Support+, AWS Enterprise Support, or AWS Unified Operations, you can use AWS Support Center to get expert assistance with this solution. The following sections provide instructions.

## Create a case

1. Sign in to Support Center.
2. Choose **Create case**.

## How can we help?

1. Choose **Technical**.
2. For **Service**, choose **Solutions**.
3. For **Category**, choose **Other Solutions**.
4. For **Severity**, choose the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions.

If you cannot resolve your question with these links, choose **Next step: Additional information**.

## Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail, including the name of this product and the version you are using, such as this example: Innovation Sandbox on AWS vX.Y.Z.
3. Choose **Attach files**.
4. Attach the information that AWS Support needs to process the request.

## Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

# Solve now or contact us

Review the **Solve now** solutions.

If you cannot resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

# Uninstall the solution

> **Important**
>
> We recommented removing accounts from the account pool before you delete the stacks to prevent accounts from incurring costs.

To uninstall the solution, follow these steps:

- End leases and eject accounts
- Uninstall stacks
- Delete the custom application from the IDC

# End leases and eject accounts

## Enable maintenance mode

Maintenance mode allows Admins to perform sensitive maintenance work like setup, troubleshooting, upgrading, or teardown of the solution.

When you enable maintenance mode, it will stop users and managers from making API requests to the solution, and any new API requests will not interfere with maintenance tasks being performed by the Admin.

To enable maintenance mode:

1. Log in to the AWS account where the Innovation Sandbox Hub and data stacks are deployed, and select the correct home Region.
2. Navigate to AWS AppConfig, and from the left pane, select **Applications**.
3. On the Applications page, select **InnovationSandboxData-Config-Application-XXXXXXX**. The Application details display.
4. Under **Configuration Profiles and Feature Flags**, select **InnovationSandboxData-Config-GlobalConfigHostedConfiguration-XXXXX** configuration profile, and choose **Create**.
5. Update the **maintenanceMode** value to `true`.

```
...
# Put the solution into maintenance mode
maintenanceMode: true
...
```

6. Select **Create hosted configuration version**.
7. Select **Start Deployment**, and choose the latest hosted configuration version you just created.
8. Choose **Start Deployment**.

This will set the account to Maintenance mode.

## End all Active and Frozen leases

In this step, you will terminate all active and frozen leases to stop incurring costs for these accounts.

1. Log in to the web UI as an **Administrator**.
2. From the left pane, select **Leases**.
3. On the Leases page, under Filter options, for status, filter for all *Active* and *Frozen* leases if not already selected by default.
4. Under the **Leases** section, select all the leases matching the filter criteria.
5. From the **Actions** dropdown, select **Terminate**.

**Note**: If there are multiple pages of leases, repeat this for all leases that match the *Active* and *Frozen* filters.

This will terminate the leases and submit the accounts for clean-up. Depending on the number of accounts, clean-up may take a few minutes.

## Eject accounts

In this step, you will manually eject accounts that have been cleaned up, and are available for reuse.

1. Log in to the web UI as an **Administrator**.
2. From the left pane, select **Administration** > **Accounts**. The Accounts page displays all the accounts currently in the account pool.

3. Search for, and select all the accounts you want to eject from the account pool. You can eject any accounts from the account pool, except those in the **Clean up** state.

> **Note**
>
> If a clean-up failed on an account, that account will be moved to a Quarantine state. After you troubleshoot these accounts, you can manually clean-up accounts in Quarantine. Accounts in Quarantine will continue to incur cost, so make sure you manually troubleshoot these accounts before attempting to clean-up these accounts.

4. From the **Actions** dropdown, select **Eject account**.
5. On the confirmation dialog, select **Submit** to confirm.

**Note**: If there are multiple pages for accounts, repeat this for all accounts you want to eject.

This will eject the accounts from the Account pool.

## Move accounts out of the Organizational Unit

In this step, you will move accounts out of the Organization Unit so that the StackSet can delete all the stack instances from the sandbox account.

1. Log in to the Organization Management account, and navigate to AWS Organizations.
2. From the left pane, select **AWS Accounts**.
3. From the organization structure tree, select the Innovation Sandbox OU, named *<NAMESPACE>_InnovationSandboxAccountPool*. For example, *myisb_InnovationSandboxAccountPool*.
4. Confirm that there are no other accounts in the OUs other than the *Exit* or *Entry* OUs. If there are accounts in other Account Pool OUs, eject these accounts using steps described in the Eject accounts section.
5. Move the accounts in *Exit* to outside the Innovation Sandbox OU, or the root OU.

This will ensure that there are no accounts in the OU before you uninstall the stacks for the solution.

# Uninstall solution stacks

You can uninstall the stacks, use the AWS Management Console or the AWS Command Line Interface (AWS CLI).

Make sure you uninstall the stacks in this order:

1. Compute stack
2. Data stack
3. IDC stack
4. AccountPool stack

## Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.
2. Select the stack you want to delete.
3. Choose **Delete stack**.

> **Note**
>
> Make sure you uninstall the stacks in this order: Compute, Data, IDC, and AccountPool.

### Using AWS Command Line Interface

Verify that AWS CLI is available in your environment. For installation instructions, refer to What Is the AWS Command Line Interface in the *AWS CLI User Guide*.

Once you have access to AWS CLI, run the following command:

```
$ aws cloudformation delete-stack --stack-name <STACK_NAME>
```

> **Note**
>
> Make sure you uninstall the stacks in this order: Compute, Data, IDC, and AccountPool.

# Resources retained after deletion

Some resources, which contain customer data, are not deleted automatically when you uninstall the stacks. The cost of these resources is minimal, and you can manually delete these resources.

### Compute stack

- Customer Managed Key
  - `AwsSolutions/InnovationSandbox/InnovationSandbox-Compute`
- CloudWatch log groups
  - `InnovationSandbox-Compute-ISBLogGroupXXXXX`
  - `InnovationSandbox-Compute-ISBLogGroupCustomResourcesXXXXX`
- S3 buckets
  - CloudFront distribution host (`innovationsandbox-compute-cloudfrontuiapiisbfronte-XXXXX`)
  - CloudFront distribution access log (`innovationsandbox-compute-cloudfrontuiapiisbfronte-XXXXX`)
  - Application logs archive (`innovationsandbox-compute-logarchivingisblogsarchi-XXXXX`)

### Data stack

- Customer Managed Key
  - `AwsSolutions/InnovationSandbox/InnovationSandbox-Data`
- DynamoDB tables
  - `InnovationSandbox-Data-LeaseTableXXXXX`
  - `InnovationSandbox-Data-LeaseTemplateTableXXXXX`
  - `InnovationSandbox-Data-AccountTableXXXXX`

### IDC stack

- Customer Managed Key
  - `AwsSolutions/InnovationSandbox/InnovationSandbox-IDC`
- CloudWatch log group

- `InnovationSandbox-IDC-ISBLogGroupCustomResourcesXXXXX`
- Innovation Sandbox groups
  - <NAMESPACE>_IsbUsersGroup
  - <NAMESPACE>_IsbManagersGroup
  - <NAMESPACE>_IsbAdminsGroup

**Account Pool stack**

- Customer Managed Key
  - `AwsSolutions/InnovationSandbox/InnovationSandbox-AccountPool`
- CloudWatch log group
  - `InnovationSandbox-AccountPool-ISBLogGroupCustomResourcesXXXXX`

# Delete the custom application in IAM Identity Center

In this step, delete the SAML2.0 application you created using the instructions in the Create SAML application section.

To delete the application:

1. Log in to the account where the IAM Identity Center is enabled (usually the Organization Management account), and the IDC stack is deployed.
2. Navigate to the AWS IAM Identity Center console, and select the Innovation Sandbox home region.
3. From the left pane, select **Groups**.
4. To remove users from the three Innovation Sandbox groups:
   a. Select a group.
   b. Select the **Users** tab.
   c. Select all the users.
   d. Choose **Remove users from group**.
   e. If there are more than one page of users, repeat this for all users.
5. Under **Application assignments**, select **Applications**.
6. Choose the **Customer managed** tab, and select the name of your application to view details.

7. Under **Assigned users and groups**, select all the groups and users associated with the application, and choose **Remove access**.
8. Navigate back to the list of **Customer managed** applications.
9. Select the application name, and under **Actions**, select **Remove**.

This will remove users from all groups, and delete the SAML2.0 application from your IAM Identity Center.

# Developer guide

This section provides the source code, and list of API endpoints for the solution.

## Source code

To download the templates and scripts for this solution, and to share your customizations with others, refer to the Innovation Sandbox on AWS GitHub repository.

## List of solution API endpoints

All features from the Innovation Sandbox on AWS solution are available as API endpoints.

To view the list of current API endpoints in an OpenAPI specification format, refer to the Innovation Sandbox API specification.

# Reference

This section includes information about an optional feature for collecting unique metrics for this solution and a list of Amazon staff who contributed to this solution.

## Data collection

This solution sends operational metrics to AWS (the "Data") about the use of this solution. We use this Data to better understand how customers use this solution and related services and products. AWS's collection of this Data is subject to the AWS Privacy Notice.

The following information is collected and sent to AWS:

- Hub Account ID
- Lease Approved Events
  - Maximum budget amount configured for the lease
  - Lease duration in hours
  - Whether the lease was automatically approved or required manual approval
  - Creation method (indicates whether lease was created via user request or manager assignment)
- Account Cleanup Events
  - Number of accounts successfully cleaned (based on Step Function success metrics)
  - Duration of failed account cleanup attempts
  - Duration of successful account cleanup attempts
- Lease Terminated Events
  - Maximum budget amount that was configured for the lease
  - Actual amount spent during the lease period
  - Maximum duration that was configured for the lease
  - Actual duration the lease was active
  - Reason for lease termination (expired, manually terminated, budget exceeded, etc.)
- LeaseUnfrozen
  - Total number of leases unfrozen
  - Frequency of unfreeze events per individual lease

- Spend Monitoring (monthly heartbeat — 4th of every month)
  - Total cost of all sandbox accounts
  - Total operational cost of running the solution infrastructure
- Deployment Summary (daily heartbeat)
  - Total number of lease templates configured
  - Number of public lease templates (visible to all users)
  - Number of private lease templates (visible only to administrators and managers)
  - Number of leases created by managers on behalf of users
  - Number of leases created by users through self-service requests
  - Number of accounts available in the account pool
  - Number of accounts currently active with leases
  - Number of accounts in frozen state
  - Number of accounts undergoing cleanup process
  - Number of accounts in quarantine state requiring manual intervention

# Contributors

- Wayne Soutter
- Chris Ellis
- Rakshana Balakrishnan
- Nils de Vries
- Emma Arrigo
- Claudia Woods
- Joan Morgan
- Todd Gruet
- Shu Jackson
- Celia Ng
- Rainer Moeller
- Lalit Grover
- Kevin Hargita
- Caleb Pearson

- Abe Wubshet
- Adrian Tadros
- Sanjay Reddy Kandi
- Vincent Rioux
- Swapnil Ogale
- Elie Elmalem
- Patrick Quinlan

# Revisions

Refer to the CHANGELOG.md file in the GitHub repository.

# Notices

The solution is licensed under the terms of the Apache License, Version 2.0.

## Terms of Use for Admins

The **Innovation Sandbox on AWS ("ISB")** allows you to experiment with AWS resources in non-production accounts. Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services including ISB are provided "as is" and AWS does not make any warranties, representations, or conditions of any kind, whether express or implied about ISB. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

You assume all responsibility for your use of ISB and making your own independent assessments of ISB, as well as any compliance with any additional terms, licenses, laws, rules, regulations, policies, or standards that apply to you, and, your use of ISB is subject to the AWS Shared Responsibility Model.

These Terms of Use supplement, and do not modify, any other existing agreements between you and AWS.

- **Non-Production Environments Only**: The ISB solution is for use only for experiments in a non-production environment and may make irreversible changes to your environment, such as terminating AWS resources. It is not for use in production accounts.
- **Limitations of Cost Control Mechanisms**: AWS makes no guarantees that usage cost will never exceed the budget limit set in ISB, and ISB may not prevent spend from going over the budget in all cases.
- **May Not Terminate All AWS Resources**: In certain limited situations, ISB may not delete all AWS Resources and AWS will attempt to notify you that manual intervention is necessary for these accounts. These accounts could continue to incur costs until manually resolved by you. AWS makes no guarantees regarding the automatic termination of these resources by ISB and you are responsible for all fees in cases where ISB does not automatically terminate resources.
- **No Third-Party Use**: ISB allows you to provide your own AWS accounts to internal end-users for learning and experimentation. You may not provide your AWS accounts to third-party users

(such as other companies or public users) as this may grant third-party users access to your AWS resources.

- **Manually Adding New Users**: If you have manually added additional users to an AWS account which has already been granted to a sandbox user by ISB, it is your responsibility to ensure deletion of the user's access after their sandbox use.
- **Fraud and Abuse Detection**: You are responsible for monitoring your sandbox account to detect any cases of potential fraud, abuse, or misuse.