Automatisierte Installationsanleitung

Wickr Enterprise

Wickr Enterprise: Automatisierte Installationsanleitung

Table of Contents

Was ist Wickr Enterprise?

Wickr Enterprise ist ein end-to-end verschlüsselter, selbst gehosteter Dienst, der Organisationen und Regierungsbehörden dabei hilft, sicher über Gruppennachrichten, Sprach- one-to-one und Videoanrufe, Dateifreigabe und Bildschirmübertragung zu kommunizieren. Kunden können Wickr Enterprise nutzen, um Datenaufbewahrungspflichten im Zusammenhang mit Messaging-Apps für Privatanwender zu umgehen und die Zusammenarbeit auf sichere Weise zu erleichtern. Fortschrittliche Sicherheits- und Verwaltungskontrollen helfen Unternehmen dabei, gesetzliche und regulatorische Anforderungen zu erfüllen und maßgeschneiderte Lösungen für Herausforderungen im Bereich der Datensicherheit zu entwickeln.

Informationen können zu Aufbewahrungs- und Prüfzwecken in einem privaten, vom Kunden kontrollierten Datenspeicher protokolliert werden. Kunden haben umfassende administrative Kontrolle über Daten. Dazu gehören die Festlegung von Berechtigungen, die Konfiguration kurzlebiger Nachrichtenoptionen und die Definition von Sicherheitsgruppen. Administratoren können Workflows mit Wickr-Bots auch sicher automatisieren. Wickr Enterprise lässt sich in zusätzliche Dienste wie Active Directory und Single Sign-On (SSO) mit OpenID Connect (OIDC) integrieren. Informationen zur Konfiguration von Wickr Enterprise finden Sie unter Erste Schritte mit Wickr Enterprise.

> **Note**
>
> Wenn Sie das Wickr Enterprise-Bereitstellungspaket noch nicht haben, finden Sie weitere Informationen unter Kontaktieren Sie uns für Geschäftsanfragen.

Erste Schritte mit Wickr Enterprise

Topics

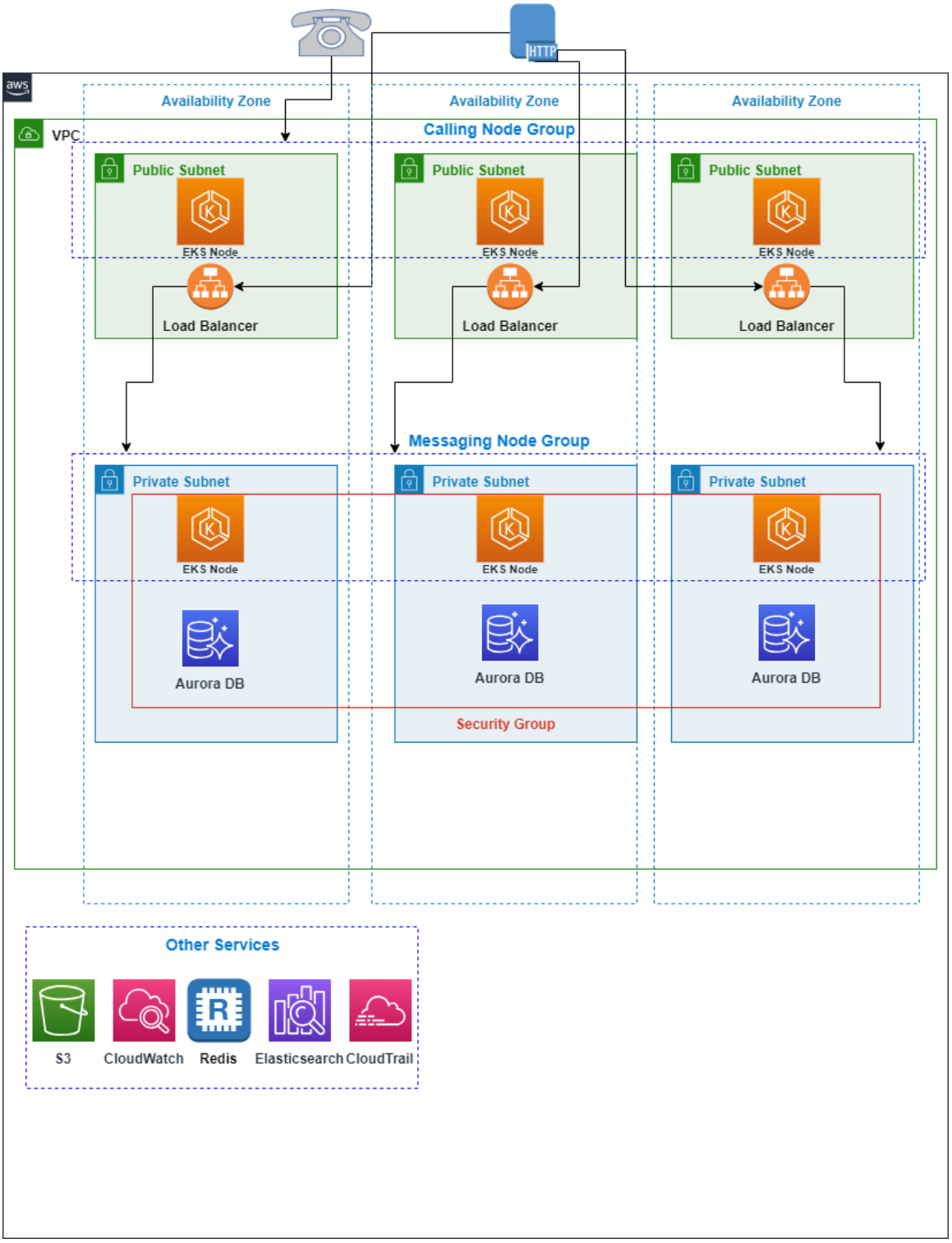
Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind:

- Laden Sie Node.js 16+ herunter
- AWS CLI mit Anmeldeinformationen für Ihr Konto konfiguriert.

  Diese werden entweder aus Ihrer Konfigurationsdatei unter `~/.aws/config` oder mithilfe der `AWS_` Umgebungsvariablen abgerufen.
- Installieren Sie kubectl. Weitere Informationen finden Sie unter Installation oder Aktualisierung von kubectl im EKSUser Amazon-Leitfaden.
- Installieren Sie Kots CLI. Weitere Informationen finden Sie unter Installation der Kots-CLI.
- Ports, die zugelassen werden sollen: 443/TCP für HTTPS- und TCP-Anrufverkehr; 16384-19999/UDP für UDP-Anrufverkehr; TCP/8443

Architektur



Abhängigkeiten installieren

Mit dem folgenden Befehl können Sie dem Standardpaket alle Abhängigkeiten hinzufügen:

```
npm install
```

Konfiguration

AWS Cloud Development Kit (AWS CDK) verwendet Kontextwerte, um die Konfiguration der Anwendung zu steuern. Wickr Enterprise verwendet CDK-Kontextwerte, um die Kontrolle über Einstellungen wie den Domainnamen Ihrer Wickr Enterprise-Installation oder die Anzahl der Tage für die Aufbewahrung von RDS-Backups zu ermöglichen. Weitere Informationen finden Sie unter Runtime-Kontext im Entwicklerhandbuch.AWS Cloud Development Kit (AWS CDK)

Es gibt mehrere Möglichkeiten, Kontextwerte festzulegen. Wir empfehlen jedoch, die Werte so `cdk.context.json` zu bearbeiten, dass sie Ihrem speziellen Anwendungsfall entsprechen. Nur Kontextwerte, die mit beginnen, `wickr/` beziehen sich auf die Wickr Enterprise-Bereitstellung. Der Rest sind CDK-spezifische Kontextwerte. Um dieselben Einstellungen beizubehalten, wenn Sie das nächste Mal ein Update über das CDK vornehmen, speichern Sie diese Datei.

Sie müssen mindestens `wickr/licensePathwickr/domainName`, und entweder `wickr/acm:certificateArn` oder `wickr/route53:hostedZoneId` und `wickr/route53:hostedZoneName` festlegen.

Mit einer öffentlich gehosteten Zone

Wenn Sie eine öffentlich gehostete Route 53 53-Zone in Ihrer haben AWS-Konto, empfehlen wir, die folgenden Einstellungen zu verwenden, um Ihren CDK-Kontext zu konfigurieren:

- `wickr/domainName`- Der Domainname, der für diese Wickr Enterprise-Bereitstellung verwendet werden soll. Wenn Sie eine öffentlich gehostete Route 53 53-Zone verwenden, werden DNS-Einträge und ACM-Zertifikate für diesen Domainnamen automatisch erstellt.
- `wickr/route53:hostedZoneName`- Name der gehosteten Route 53 53-Zone, in der DNS-Einträge erstellt werden sollen.
- `wickr/route53:hostedZoneId`- Route 53 53-Hosting-Zonen-ID, in der DNS-Einträge erstellt werden sollen.

Diese Methode erstellt in Ihrem Namen ein ACM-Zertifikat zusammen mit den DNS-Einträgen, die Ihren Domainnamen auf den Load Balancer vor Ihrer Wickr Enterprise-Bereitstellung verweisen.

Ohne eine öffentlich gehostete Zone

Wenn Sie in Ihrem Konto keine öffentlich gehostete Route 53 53-Zone haben, muss ein ACM-Zertifikat manuell erstellt und mithilfe des `wickr/acm:certificateArn` Kontextwerts in das CDK importiert werden.

- `wickr/domainName`— Der Domainname, der für diese Wickr Enterprise-Bereitstellung verwendet werden soll. Wenn Sie eine öffentlich gehostete Route 53 53-Zone verwenden, werden DNS-Einträge und ACM-Zertifikate für diesen Domainnamen automatisch erstellt.
- `wickr/acm:certificateArn`— Der ARN eines ACM-Zertifikats, das auf dem Load Balancer verwendet werden soll. Dieser Wert muss angegeben werden, wenn eine öffentlich gehostete Route 53 53-Zone in Ihrem Konto nicht verfügbar ist.

Ein Zertifikat in ACM importieren

Sie können ein extern erworbenes Zertifikat mit dem folgenden Befehl importieren:

```
aws acm import-certificate \
   --certificate fileb://path/to/cert.pem \
   --private-key fileb://path/to/key.pem \
   --certificate-chain fileb://path/to/chain.pem
```

Die Ausgabe ist der Zertifikat-ARN, der für den Wert der `wickr/acm:certificateArn` Kontexteinstellung verwendet werden sollte. Es ist wichtig, dass das hochgeladene Zertifikat für die gültig ist`wickr/domainName`, andernfalls können HTTPS-Verbindungen nicht validiert werden. Weitere Informationen finden Sie im AWS Certificate Manager Benutzerhandbuch unter Importieren eines Zertifikats.

Erstellen Sie DNS-Einträge

Da keine öffentlich gehostete Zone verfügbar ist, müssen DNS-Einträge nach Abschluss der Bereitstellung manuell erstellt werden, damit sie auf den Load Balancer vor Ihrer Wickr Enterprise-Bereitstellung verweisen.

Bereitstellung in einer vorhandenen VPC

Wenn Sie eine bestehende VPC verwenden möchten, können Sie eine verwenden. Die VPC muss jedoch so konfiguriert werden, dass sie die für EKS erforderlichen Spezifikationen erfüllt. Weitere Informationen finden Sie unter Amazon EKS-Netzwerkanforderungen für VPC und Subnetze anzeigen im Amazon EKS-Benutzerhandbuch. Stellen Sie sicher, dass die zu verwendende VPC diese Anforderungen erfüllt.

Darüber hinaus wird dringend empfohlen, sicherzustellen, dass Sie über VPC-Endpunkte für die folgenden Dienste verfügen:

- CLOUDWATCH
- CLOUDWATCH_LOGS
- EC2
- EC2_NACHRICHTEN
- ECR
- ECR_DOCKER
- ELASTISCHER LASTENAUSGLEICH
- KMS
- SECRETS_MANAGER
- SSM
- SSM_NACHRICHTEN

Um Ressourcen in einer vorhandenen VPC bereitzustellen, legen Sie die folgenden Kontextwerte fest:

- `wickr/vpc:id`- Die VPC-ID, in der Ressourcen bereitgestellt werden sollen (z. B.`vpc-412beef`).
- `wickr/vpc:cidr`- Die IPv4 CIDR der VPC (z. B.`172.16.0.0/16`).
- `wickr/vpc:publicSubnetIds`- Eine durch Kommas getrennte Liste öffentlicher Subnetze in der VPC. Der Application Load Balancer und die aufrufenden EKS-Worker-Knoten werden in diesen Subnetzen bereitgestellt (z. B.`subnet-6ce9941,subnet-1785141,subnet-2e7dc10`).
- `wickr/vpc:privateSubnetIds`- Eine durch Kommas getrennte Liste von privaten Subnetzen in der VPC. Die EKS-Worker-Knoten und der Bastion-Server werden in diesen Subnetzen bereitgestellt (z. B.). `subnet-f448ea8,subnet-3eb0da4,subnet-ad800b5`
- `wickr/vpc:isolatedSubnetIds`- Eine durch Kommas getrennte Liste isolierter Subnetze in der VPC. Die RDS-Datenbank wird in diesen Subnetzen bereitgestellt (z. B.). `subnet-d1273a2,subnet-33504ae,subnet-0bc83ac`
- `wickr/vpc:availabilityZones`- Eine durch Kommas getrennte Liste von Verfügbarkeitszonen für die Subnetze in der VPC (z. B.). `us-east-1a,us-east-1b,us-east-1c`

Weitere Informationen zu VPC-Endpunkten mit Schnittstellen finden Sie unter Zugreifen auf einen AWS Dienst mithilfe eines Schnittstellen-VPC-Endpunkts.

Andere Einstellungen

Weitere Informationen finden Sie unter Kontextwerte.

Bootstrap

Wenn Sie CDK zum ersten Mal in dieser Region verwenden, müssen Sie zuerst das Konto booten, um CDK verwenden zu können. AWS-Konto

```
npx cdk bootstrap
```

Bereitstellen

Dieser Vorgang dauert etwa 45 Minuten.

```
npx cdk deploy --all --require-approval=never
```

Nach Abschluss des Vorgangs wurde die Infrastruktur erstellt und Sie können mit der Installation von Wickr Enterprise beginnen.

Erstellen Sie DNS-Einträge

Dieser Schritt ist nicht erforderlich, wenn Sie bei der Konfiguration des CDK eine öffentlich gehostete Zone verwendet haben.

Die Ausgabe des Bereitstellungsprozesses wird einen Wert enthalten`WickrAlb.AlbDnsName`, bei dem es sich um den DNS-Namen des Load Balancers handelt. Die Ausgabe wird wie folgt aussehen:

```
WickrAlb.AlbDnsName = Wickr-Alb-1Q5IBPJR4ZVZR-409483305.us-west-2.elb.amazonaws.com
```

In diesem Fall lautet der DNS-Name`Wickr-Alb-1Q5IBPJR4ZVZR-409483305.us-west-2.elb.amazonaws.com`. Dies ist der Wert, der bei der Erstellung eines CNAME- oder A/AAAA (ALIAS-) Eintrags für Ihren Domainnamen verwendet werden sollte.

Wenn Sie nicht über die Ausgabe der Bereitstellung verfügen, führen Sie den folgenden Befehl aus, um den DNS-Namen des Load Balancers anzuzeigen:

```
aws cloudformation describe-stacks --stack-name WickrAlb \
   --query 'Stacks[0].Outputs[?OutputKey==`AlbDnsName`].OutputValue' \
   --output text
```

KOTS-Konfiguration generieren

> **Warning**
>
> Diese Datei enthält vertrauliche Informationen über Ihre Installation. Teilen oder speichern Sie sie nicht öffentlich.

Das Wickr Enterprise-Installationsprogramm benötigt eine Reihe von Konfigurationswerten für die Infrastruktur, um erfolgreich installiert zu werden. Sie können ein Hilfsskript verwenden, um die Konfigurationswerte zu generieren.

```
./bin/generate-kots-config.ts > wickr-config.json
```

Wenn Sie im ersten Schritt ein externes Zertifikat in ACM importiert haben, übergeben Sie das `--ca-file` Flag an dieses Skript, zum Beispiel:

```
./bin/generate-kots-config.ts --ca-file path/to/chain.pem > wickr-config.json
```

Wenn Sie eine Fehlermeldung erhalten, dass der Stack nicht existiert, setzen Sie die `AWS_REGION` Umgebungsvariable (`export AWS_REGION=us-west-2`) auf Ihre ausgewählte Region und versuchen Sie es erneut. Oder, wenn Sie den Kontextwert festlegen`wickr/stackSuffix`, übergeben Sie das Suffix mit dem `--stack-suffix` Flag.

Verbindung zum Kubernetes-Cluster herstellen

Auf die Amazon EKS-API kann nur über einen Bastion-Host zugegriffen werden, der im Rahmen der Bereitstellung erstellt wird. Daher müssen alle `kubectl` Befehle entweder auf dem Bastion-Host selbst ausgeführt oder über den Bastion-Host als Proxy weitergeleitet werden.

Proxyverbindungen über die Bastion

Wenn Sie zum ersten Mal eine Verbindung zum Cluster herstellen, müssen Sie Ihre lokale kubeconfig-Datei mit dem `aws eks update-kubeconfig` Befehl aktualisieren und dann in Ihrer Konfiguration festlegen. `proxy-url` Dann starten Sie jedes Mal, wenn Sie eine Verbindung zum Cluster herstellen möchten, eine SSM-Sitzung mit dem Bastion-Host, um eine Portweiterleitung zum Proxy für den API-Zugriff durchzuführen.

Einmalige Einrichtung

Auf dem `WickrEks` CloudFormation Stack befindet sich ein Ausgabewert mit einem Namen, der mit beginnt`WickrEnterpriseConfigCommand`. Der Wert enthält den vollständigen Befehl, der zum Generieren der Kubectl-Konfiguration für Ihren Cluster erforderlich ist. Diese Ausgabe kann mit dem folgenden Befehl angezeigt werden:

```
aws cloudformation describe-stacks --stack-name WickrEks \
--query 'Stacks[0].Outputs[?starts_with(OutputKey,
 `WickrEnterpriseConfigCommand`)].OutputValue' \
--output text
```

Dies sollte einen Befehl ausgeben, der mit beginnt`aws eks update-kubeconfig`. Führen Sie diesen Befehl aus.

Als Nächstes muss die Kubernetes-Konfiguration so geändert werden, dass sie Proxy-Anfragen über den Bastion-Host sendet. Dies kann mit den folgenden Befehlen erfolgen:

```
CLUSTER_ARN=$(aws cloudformation describe-stacks --stack-name WickrEks --query
 'Stacks[0].Outputs[?OutputKey==`WickrEnterpriseEksClusterArn`].OutputValue' --output
 text)
kubectl config set "clusters.${CLUSTER_ARN}.proxy-url" http://localhost:8888
```

Wenn es richtig funktioniert hat, sehen Sie eine Ausgabe wie `'Property "clusters.arn:aws:eks:us-west-2:012345678912:cluster/WickrEnterprise5B8BF472-1234a41c4ec48b7b615c6789d93dcce.proxy-url" set.'`

Port weiter zur Bastion

Um eine Verbindung zum Amazon EKS-Cluster herzustellen, müssen Sie eine SSM-Sitzung starten, um Anfragen an den Proxy weiterzuleiten, der auf Ihrem Bastion-Host ausgeführt wird. Der Befehl dazu wird als Ausgabe `BastionSSMProxyEKSCommand` auf dem `WickrEks` Stack bereitgestellt. Führen Sie den folgenden Befehl aus, um den Ausgabewert anzuzeigen:

```
aws cloudformation describe-stacks --stack-name WickrEks \
--query 'Stacks[0].Outputs[?OutputKey==`BastionSSMProxyEKSCommand`].OutputValue' \
--output text
```

Der Befehl, den es ausgibt, beginnt mit`aws ssm start-session`. Führen Sie diesen Befehl aus, um einen lokalen Proxy zu starten, der auf Port 8888 läuft und über den Sie eine Verbindung zum Amazon EKS-Cluster herstellen können. Wenn die Portweiterleitung korrekt funktioniert hat, sollte in der Ausgabe „Warten auf Verbindungen...“ stehen. Lassen Sie diesen Prozess so lange laufen, wie Sie für den Zugriff auf den Amazon EKS-Cluster benötigen.

Wenn alles korrekt eingerichtet ist, können Sie es `kubectl get nodes` in einem anderen Terminal ausführen, um die Worker-Knoten im Amazon EKS-Cluster aufzulisten:

```
kubectl get nodes
   NAME                            STATUS   ROLES    AGE     VERSION
   ip-10-0-111-216.ec2.internal   Ready     none   3d      v1.26.4-eks-0a21954
   ip-10-0-180-1.ec2.internal     Ready     none   2d23h   v1.26.4-eks-0a21954
   ip-10-0-200-102.ec2.internal   Ready     none   3d      v1.26.4-eks-0a21954
```

Wickr Enterprise installieren

Nachdem Ihre Verbindung zum Kubernetes-Cluster hergestellt wurde, können Sie mit der Installation von Wickr Enterprise mithilfe des Plugins beginnen. `kubectl kots` Sie benötigen Ihre KOTS-Lizenzdatei (eine von Wickr bereitgestellte `.yaml` Datei) und Ihre Datei mit den Konfigurationswerten, die in der Datei `wickr-config.json` im Abschnitt KOTS-Konfiguration generieren gespeichert wurden. Weitere Informationen zu Generate KOTS Config finden Sie unter Generate KOTS Config.

Wickr Enterprise manuell installieren

Der folgende Befehl startet die Installation von Wickr Enterprise:

```
kubectl kots install wickr-enterprise-ha \
    --license-file ./license.yaml \
    --config-values ./wickr-config.json  \
    --namespace wickr \
    --skip-preflights
```

Sie werden aufgefordert, ein Passwort für die KOTS Admin Console einzugeben. Speichern Sie dieses Passwort, da Sie es in future benötigen, um die Konfiguration Ihrer Wickr Enterprise-Installation zu aktualisieren oder zu ändern.

Wenn die Installation abgeschlossen ist, `kubectl kots` wird (normalerweise`http://localhost:8080`) ein lokaler Port geöffnet, über den Sie auf die KOTS Admin-Konsole zugreifen können. Sie können den Status Ihrer Wickr Enterprise-Installation auf dieser Website ändern oder überwachen oder mit der Einrichtung von Wickr beginnen, indem Sie den Domainnamen aufrufen, den Sie für Ihre Installation in Ihrem Browser konfiguriert haben.

Wickr Enterprise mit Lambda installieren

Während der CDK-Bereitstellung wird ein Lambda erstellt und aufgerufen, um die Wickr Enterprise-Installation in Ihrem Namen automatisch abzuschließen. Um es manuell aufzurufen, öffnen Sie die AWS Konsole und suchen Sie nach `WickrLambda-func*` Lambda-Funktion. Wählen `test` Sie auf der Registerkarte Test die Option, dass die Eingabe irrelevant ist.

Nach der Installation

Für die Verwaltung Ihrer Wickr Enterprise-Installation stehen zwei Webkonsolen zur Verfügung: die KOTS Admin Console und die Wickr Admin Console.

> **Note**
>
> Nehmen Sie alle erforderlichen Änderungen vor, um die Sicherungs- und Protokollierungsrichtlinien Ihres Unternehmens widerzuspiegeln (Amazon S3 S3-Einstellungen, Elastic Load Balancing Balancing-Zugriffsprotokolle, Amazon Virtual Private Cloud Flow-Logs).

KOTS-Admin-Konsole

Diese Schnittstelle wird für die Verwaltung der bereitgestellten Version von Wickr Enterprise verwendet. Sie können den Status der Installation einsehen, Konfigurationen ändern oder Upgrades durchführen. Auf die KOTS Admin Console kann nur über einen Kubernetes-Port-Forward zugegriffen werden, der mit dem folgenden Befehl geöffnet werden kann:

```
kubectl kots --namespace wickr admin-console
```

> **Note**
>
> Sie müssen zuerst Ihre Bastion-Verbindung einrichten, wie im Abschnitt Port Forward to the Bastion beschrieben. Weitere Informationen zur Portweiterleitung zur Bastion finden Sie unter Proxyverbindungen über die Bastion.

Wenn die Portweiterleitung erfolgreich konfiguriert wurde, gibt der vorherige Befehl Folgendes aus:

```
• Press Ctrl+C to exit
• Go to http://localhost:8800 to access the Admin Console
```

Verwenden Sie die angegebene URL, um auf die KOTS Admin Console zuzugreifen. Das Passwort für die Anmeldung ist das, das Sie bei der Ausführung `kubectl kots install` während der

Installation gewählt haben. Wenn Sie Ihr Passwort zurücksetzen müssen, finden Sie weitere Informationen unter Passwort für die KOTS Admin-Konsole zurücksetzen.

Wickr Admin-Konsole

Diese Schnittstelle wird für die Konfiguration Ihrer Wickr Enterprise-Installation verwendet, um Netzwerke, Benutzer und Verbände einzurichten. Es ist über HTTPS unter dem DNS-Namen zugänglich, den Sie so konfiguriert haben, dass er auf Ihren Load Balancer verweist. Wenn DNS automatisch mit einer öffentlich gehosteten Zone konfiguriert wurde, entspricht der Domainname dem Wert des `wickr/domainName` Kontextwerts.

Der Standardbenutzername ist`admin`, mit dem Passwort`Password123`. Sie müssen dieses Passwort bei der ersten Anmeldung ändern.

Werte im Kontext

Kontextwerte sind Schlüssel-Wert-Paare, die einer App, einem Stack oder einem Konstrukt zugeordnet werden können. Sie können Ihrer App aus einer Datei (normalerweise entweder `cdk.json` oder `cdk.context.json` in Ihrem Projektverzeichnis) oder über die Befehlszeile bereitgestellt werden. CDK verwendet Kontextwerte, um die Konfiguration der Anwendung zu steuern. Wickr Enterprise verwendet CDK-Kontextwerte, um die Kontrolle über Einstellungen wie den Domainnamen Ihrer Wickr Enterprise-Installation oder die Anzahl der Tage für die Aufbewahrung von RDS-Backups zu ermöglichen.

Es gibt mehrere Möglichkeiten, Kontextwerte festzulegen. Wir empfehlen jedoch, die Werte so `cdk.context.json` zu bearbeiten, dass sie Ihrem speziellen Anwendungsfall entsprechen. Nur Kontextwerte, die mit beginnen, `wickr/` beziehen sich auf die Wickr Enterprise-Bereitstellung.

Name	Beschreibung	Standard
`wickr/licensePath`	Der Pfad zu Ihrer KOTS-Lizenz (eine von `.yaml` Wickr bereitgestellte Datei).	Null
`wickr/domainName`	Der Domainname, der für diese Wickr Enterprise-Bereitstellung verwendet werden soll. Wenn Sie eine öffentlich gehostete Route 53 53-Zone verwenden, werden DNS-Einträge und ACM-Zertifikate für diesen Domainnamen automatisch erstellt.	Null
`wickr/route53:hostedZoneId`	ID der gehosteten Route 53 53-Zone, in der DNS-Einträge erstellt werden sollen.	Null
`wickr/route53:hostedZoneName`	Name der gehosteten Route 53 53-Zone, in der DNS-Einträge erstellt werden sollen.	Null

Name	Beschreibung	Standard
`wickr/acm:certificateArn`	ARN eines ACM-Zertifikats, das auf dem Load Balancer verwendet werden soll. Dieser Wert muss angegeben werden, wenn eine öffentlich gehostete Route 53 53-Zone in Ihrem Konto nicht verfügbar ist.	Null
`wickr/caPath`	Zertifikatspfad, nur erforderlich, wenn selbstsignierte Zertifikate verwendet werden.	Null
`wickr/vpc:id`	Die ID der VPC, in der Ressourcen bereitgestellt werden sollen. Nur erforderlich bei der Bereitstellung in einer vorhandenen VPC. Wenn diese Option nicht gesetzt ist, wird eine neue VPC erstellt.	Null
`wickr/vpc:cidr`	IPv4 CIDR, das mit der erstellten VPC verknüpft werden soll. Wenn Sie die Bereitstellung in einer vorhandenen VPC vornehmen, legen Sie hier den CIDR der vorhandenen VPC fest.	172.16.0.0/16
`wickr/vpc:availabilityZones`	Durch Kommas getrennte Liste der Availability Zones. Nur erforderlich bei der Bereitstellung in einer vorhandenen VPC.	Null

Name	Beschreibung	Standard
`wickr/vpc:publicSubnetIds`	Durch Kommas getrennte Liste der öffentlichen Subnetze. IDs Nur erforderlich bei der Bereitstellung in einer vorhandenen VPC.	Null
`wickr/vpc:privateSubnetIds`	Durch Kommas getrennte Liste der privaten Subnetze. IDs Nur erforderlich bei der Bereitstellung in einer vorhandenen VPC.	Null
`wickr/vpc:isolatedSubnetIds`	Durch Kommas getrennte Liste isolierter Subnetze IDs für die RDS-Datenbank. Nur erforderlich bei der Bereitstellung in einer vorhandenen VPC.	Null
`wickr/rds:deletionProtection`	Aktivieren Sie den Löschschutz für RDS-Instances.	true
`wickr/rds:removalPolicy`	Richtlinie zum Entfernen der RDS-Instances „Snapshot“, „Destroy“ oder „Retain“.	Snapshot
`wickr/rds:readerCount`	Anzahl der Reader-Instances, die im RDS-Cluster erstellt werden sollen.	1
`wickr/rds:instanceType`	Instanztyp, der für RDS-Instances verwendet werden soll.	r6g.xlarge
`wickr/rds:backupRetentionDays`	Anzahl der Tage, an denen Backups aufbewahrt werden sollen.	7

Name	Beschreibung	Standard
`wickr/eks:namespace`	Standard-Namespace für Wickr-Dienste in EKS.	Wickr
`wickr/eks:defaultCapacity`	Anzahl der EKS-Worker-Knoten für die Messaging-Infrastruktur.	3
`wickr/eks:defaultCapacityCalling`	Anzahl der EKS-Worker-Knoten für die Anrufinfrastruktur.	2
`wickr/eks:instanceTypes`	Durch Kommas getrennte Liste von Instanztypen, die für Messaging EKS-Worker-Knoten verwendet werden sollen.	m5.xlarge
`wickr/eks:instanceTypesCalling`	Durch Kommas getrennte Liste von Instanztypen, die für das Aufrufen von EKS-Worker-Knoten verwendet werden sollen.	c5n.large
`wickr/eks:enableAutoscaler`	Schaltet die Aktivierung der Cluster-Autoscaler-Funktionalität für EKS um.	true
`wickr/s3:expireAfterDays`	Anzahl der Tage, nach denen Datei-Uploads aus dem S3-Bucket entfernt werden.	1095
`wickr/eks:clusterVersion`	Cluster-Versionen, einschließlich Kubernetes-Version, KubecTLLayer-Version, AlbController-Version, Version und mehr. nodeGroupRelease	1.27

Name	Beschreibung	Standard
`wickr/stackSuffix`	Ein Suffix, das auf CloudFormation Stack-Namen angewendet wird.	"
`wickr/autoDeployWickr`	Stellen Sie die Wickr-Anwendung automatisch mit Lambda bereit.	true

Ressourcen vernichten

Um alles zu löschen, was von dieser AWS CDK Anwendung erstellt wurde, müssen Sie den `WickrRds` Stapel vor allen anderen Stacks löschen.

Damit die Amazon RDS-Ressourcen ordnungsgemäß gelöscht werden können, muss der Löschschutz deaktiviert und die Entfernungsrichtlinie muss entweder auf `snapshot` oder eingestellt sein`destroy`. Wenn dies nicht die aktuellen Einstellungen sind, ändern Sie die `wickr/rds:removalPolicy` Werte `wickr/rds:deletionProtection` und in Ihrem AWS CDK Kontext und stellen Sie den Amazon RDS-Stack erneut bereit, indem Sie Folgendes ausführen`npx cdk deploy -e WickrRds`.

Sobald der Löschschutz und die Entfernungsrichtlinie ordnungsgemäß eingerichtet sind, führen Sie `cdk destroy` für den `WickrRds` Stack Folgendes aus:

```
npx cdk destroy WickrRds
```

Wenn der `WickrRds` Stapel mit der Zerstörung fertig ist, können die verbleibenden CloudFormation Stapel mit dem folgenden Befehl zerstört werden:

```
npx cdk destroy --all
```

Fehlerbehebung

Löschen des Wickr-Namespace

Wenn Sie den `wickr` Namespace löschen müssen, um von vorne zu beginnen, ist es wichtig, dass Sie zuerst alle Dienstkonten sichern, die von CDK in diesem Namespace erstellt wurden. Mit diesen Dienstkonten können Wickr-Dienste über IAM-Rollen mit AWS APIs ihnen kommunizieren. Ohne sie funktionieren Aufgaben wie das Hochladen von Dateien über Amazon Simple Storage Service (Amazon S3) nicht mehr.

Verwenden Sie den folgenden Befehl, um die Service Accounts zu sichern und den `wickr` Namespace und die entsprechenden Service Accounts zu löschen und neu zu erstellen:

```
kubectl -n wickr get sa fileproxy -o yaml > fileproxy-sa.yaml && \
  kubectl delete ns wickr && \
  kubectl create ns wickr && \
  kubectl apply -f fileproxy-sa.yaml
```

Das Passwort für die KOTS Admin-Konsole zurücksetzen

Sie können Ihr Passwort für die KOTS Admin-Konsole mit dem folgenden Befehl zurücksetzen:

```
kubectl kots -n wickr reset-password
```

Wenn Sie dieses Passwort ändern, möchten Sie möglicherweise auch das `wickr/kots` Secrets Manager Manager-Geheimnis aktualisieren, obwohl es in der Regel von keiner Automatisierung erneut verwendet wird.

Probleme beim Herstellen einer Verbindung zum EKS-Cluster mit Bastion

Wenn Ihre Verbindung zum EKS-Cluster über die Bastion langsam zu sein scheint oder gelegentlich ein Timeout auftritt, wird beim Ausführen `kubectl` von Befehlen möglicherweise der folgende Fehler angezeigt:

net/http: Die Anfrage wurde beim Warten auf die Verbindung abgebrochen (Client.Timeout wurde beim Warten auf Header überschritten)

Dieses Problem kann häufig behoben werden, indem Sie sich per SSM beim Bastion-Host anmelden (siehe auf dem Stack) und den `BastionSSMCommand` Dienst neu starten: WickrEks `tinyproxy`

```
sudo systemctl restart tinyproxy
```

Maßgeschneiderte Installation

Im Abschnitt Benutzerdefinierte Installation erfahren Sie, wie Sie Wickr Enterprise installieren.

Topics

Voraussetzungen

Bevor Sie mit der Installation von Wickr Enterprise beginnen, stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind.

Hardwareanforderungen

Wickr Enterprise benötigt für den Betrieb einen Kubernetes-Cluster. Es ist möglich, auf einem einzelnen Knoten zu arbeiten, wenn der Low Resource Mode aktiviert ist. Dies wird jedoch nicht für den allgemeinen Produktionsgebrauch empfohlen. In einer Produktionsbereitstellung empfehlen wir mindestens drei Messaging-Worker-Knoten sowie mindestens zwei Calling-Worker-Knoten.

Ein Worker-Knoten sollte die folgenden Mindestanforderungen haben.

- 2 bis 4 CPU-Kerne
- 8 GB RAM
- 200 GB Festplattenspeicher

Mindestanforderungen an die Hardware

Ein einzelner Worker-Knoten-Cluster, der im Niedrigressourcenmodus ausgeführt wird, benötigt mindestens 3000 Mio. CPU und 5846 MB RAM. Dies beinhaltet nicht die Kube-System-Pods.

Ressourcenanforderungen nach Pod

Pod-Name	Eigentümer	CPU	Arbeitsspeicher
Admin-API	Korb	100 m	256 Meilen
directory	Korbgeflecht	100 m	128 Minuten
abläuft	Korbgeflecht	100 m	128 Minuten
Datei-Proxy	Wickr	100 m	256 Meilen
oidc	Korbgeflecht	100 m	128 Minuten
OpenSearch	Korbgeflecht	500 m	100 Minuten
Orville	Korb	50 m	128 Minuten
Orville-Redis	Korb	50 m	128 Minuten
Push-Gerät	Korb	100 m	128 Minuten
Rabbitmq	Korbwaren	50 m	256 Minuten
reagieren	Korb	100 m	64 Minuten
Quittungen	Wickr	250 m	128 Minuten
redis	Korbgeflecht	50 m	128 Minuten

Pod-Name	Eigentümer	CPU	Arbeitsspeicher
Server-API	Korb	250 m	256 Meilen
Telefonzentrale	Korb	250 m	512 Mi
Kotsadm	KOTS	50 m	50 Minuten
Kotsadm-Minio	KOTS	100 m	512 Mi
kotsadm-rqlite	KOTS	200m	1 Gi
Mini-Operator	Interner S3	200m	256 Mi
Mini-Mandant	Interner S3	100 m	256 Meilen
mysql-primär	Internes MySQL	100 m	512 Mi
mysql-sekundär	Internes MySQL	100 m	512 Mi

Anforderungen an den Speicherplatz

Wickr Enterprise benötigt einen Standard, der bei der Erstellung von Persistent Volume Claims verwendet werden StorageClass soll. Bei der Bereitstellung in einer Air-Gap-Umgebung oder vor Ort müssen Sie möglicherweise eine für Ihren Cluster konfigurieren. Eine verfügbare Option ist Longhorn. Die empfohlenen Speicherplatzanforderungen variieren je nach Verwendung der Optionen Internal S3 und Internal Mysql sowie der Menge an Speicherplatz, den Sie für Datei-Uploads zur Verfügung haben möchten.

- Internes Zwischenspeichern von Bildern: ~60 Gi
- RabbitMQ: 24 Gi Standard//8 Gi im Modus mit niedrigen Ressourcen
- Redis: 24 Gi Standard/8 Gi im Modus mit niedrigen Ressourcen
- OpenSearch: 24 Gi Standard/8 Gi im Modus mit niedrigen Ressourcen
- Internes Mysql: 80 Gi Standard/20 Gi im Modus mit niedrigen Ressourcen
- Interner S3:160 Gi Standard//2 Gi im Modus mit niedrigen Ressourcen
- KOTS Mini: 4 Gi
- KOTS Größe: 1 GB

Minimale Speichergröße

- 377 Gi Standard mit internem S3 und internem Mysql
- 111 Gi im Modus mit niedrigen Ressourcen

Anforderungen an die Kubernetes-Version

Wickr Enterprise setzt auf repliziertes KOTS. Replicated, eine kommerzielle Softwarevertriebsplattform, bietet eine Liste der derzeit unterstützten Versionen von Kubernetes. Weitere Informationen finden Sie unter Kubernetes-Versionskompatibilität.

Anforderungen an die Software

Wickr Enterprise benötigt für den Betrieb einen Kubernetes-Cluster und KOTS. Informationen zu den unterstützten Betriebssystem- und Kubernetes-Versionen finden Sie in der KOTS-Dokumentation. Weitere Informationen finden Sie unter Mindestsystemanforderungen.

Hostsystem für Entwickler

Betriebssystem — Die Befehle in dieser Dokumentation sind so konzipiert, dass sie unter Linux, macOS oder Windows funktionieren, auf denen WSL (Windows Subsystem for Linux) installiert ist.

Interne Stateful Services

Wickr Enterprise kann interne Dienste sowohl für MySQL-Datenbanken als auch für S3-kompatiblen Speicher bereitstellen. Für allgemeine Produktionszwecke wird jedoch empfohlen, diese Dienste außerhalb des Kubernetes-Clusters bereitzustellen.

- MySQL 5.7-Datenbank
  - Amazon RDS MySQL 5.7- oder MySQL 5.7-Datenbank (extern)
  - Mysql-Bitnami-Helm-Diagramm (intern)
  - Dateispeicherung
    - Amazon S3- oder S3-kompatibler Speicheranbieter (extern)
    - Steuerkarte für Minio-Bediener (intern)

Netzwerkanforderungen

Wickr Enterprise benötigt einen FQDN, SSL-Zertifikate und bestimmte offene TCP- und UDP-Ports.

- FQDN: Eine Domain oder Subdomain, die von der Wickr Enterprise-Bereitstellung verwendet werden soll.
- SSL-Zertifikat: Ein SSL-Zertifikatsschlüsselpaar, das von einer öffentlichen Zertifizierungsstelle signiert wurde, oder ein selbstsigniertes Zertifikatsschlüsselpaar. Das Zertifikat muss den FQDN im Common Name und auch als SAN-DNS-Eintrag auflisten. Das Zertifikat muss auch die extendedKeyUsage ServerAuth-Erweiterung aktivieren.
- Für Online-Installationen ist ausgehender Zugriff auf replizierte Ressourcen und Ressourcen von Drittanbietern erforderlich. Replicated verwaltet eine Liste ihrer IP-Adressen. Weitere Informationen finden Sie unter Replizierte IP-Adressen. Replicated verwaltet auch eine Liste der benötigten Ressourcen von Drittanbietern. Weitere Informationen finden Sie unter Firewall-Öffnungen für Online-Installationen.
- Air-Gap-Installationen erfordern Zugriff auf eine private Container-Registry.

Messaging-Knoten

Messaging-Knoten benötigen keine öffentliche IPV4 Adresse und sollten sich in einem privaten Subnetz befinden. Der Nachrichtenverkehr wird über den LoadBalancer oder Ingress in den Cluster geleitet.

Knoten aufrufen

Aufrufende Knoten benötigen eine öffentliche IPV4 Adresse und müssen sich daher in einem öffentlichen Subnetz befinden. Gesprächsmedien werden standardmäßig über UDP übertragen. Wenn TCP-Anrufe aktiviert sind, akzeptiert der TCP-Proxy Verbindungen über TCP 443 und leitet sie an den Orville-Dienst weiter.

- TCP: 443 TCP-Proxy aufrufen
- UDP: 16384-16484 Streams Audio/Video

Zugriff auf Installation und Konfiguration

Der Zugriff auf die KOTS Admin Console für die Installation und Konfiguration erfolgt über einen Kubernetes-Port-Forward.

```
kubectl kots admin-console -n wickr
```

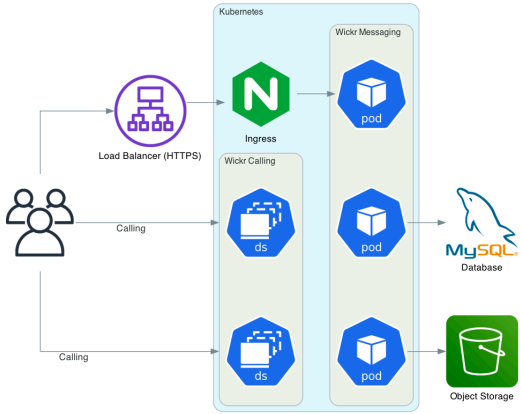
Anforderungen an die Lizenz

Für die Installation ist eine Lizenzdatei im Yaml-Format erforderlich. Diese wird Ihnen vom Wickr-Support zur Verfügung gestellt.

Architektur

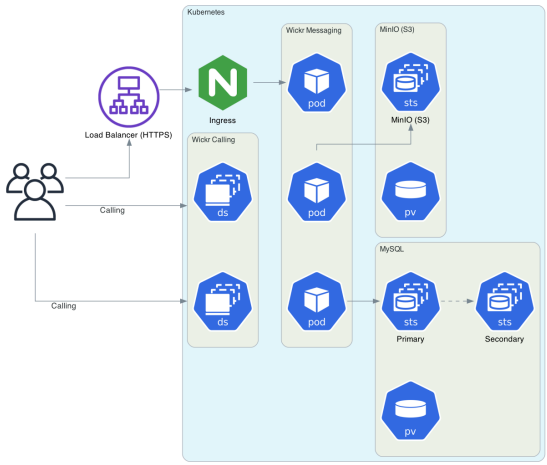
Empfohlene Produktionsarchitektur

Das folgende Diagramm zeigt Wickr Enterprise, wie für die Produktion empfohlen, konfiguriert, wobei sich sowohl MySQL- als auch Object Storage-Dienste außerhalb des Kubernetes-Clusters befinden.



Interne Architektur oder Testarchitektur

Das folgende Diagramm zeigt die Konfiguration von Wickr Enterprise unter Verwendung der internen MYSQL- und Object Storage-Dienste. Obwohl es die spezifischen Anforderungen bestimmter Bereitstellungen erfüllen kann, wird es nicht für den allgemeinen Produktionseinsatz empfohlen.



Installation

1. Installieren Sie kubectl und kots CLI.
2. Connect zum Kubernetes-Cluster her.
3. Besorgen Sie sich die Wickr Enterprise-Lizenzdatei vom Wickr Support.
4. Installieren Sie Wickr Enterprise mit dem folgenden Befehl.

```
kubectl kots install wickr-enterprise-ha \
   --license-file ./license.yaml \
   --namespace wickr
```

> **Note**
>
> license.yaml steht für Ihre bereitgestellte Lizenzdatei.

Nach der Erstinstallation bietet die KOTS Admin Console Verwaltungs- und Konfigurationsoptionen auf Clusterebene.

KOTS Admin-Konsole

Diese Schnittstelle wird für die Verwaltung der bereitgestellten Version von Wickr Enterprise verwendet. Sie können den Status der Installation einsehen, Konfigurationen ändern oder Upgrades von Wickr Enterprise durchführen. Auf die KOTS Admin Console kann nur über einen Kubernetes-Port-Forward zugegriffen werden, der mit dem folgenden Befehl geöffnet werden kann:

```
kubectl kots admin-console -n wickr
```

Einstellungen für den Zugriff

Ingress-Controller

Wickr Enterprise unterstützt vier Arten von Ingress-Controllern:

- LoadBalancer (Standard)
  - Das Loadbalancer-Objekt erfordert in vollständig lokalen Installationen möglicherweise eine explizite Konfiguration, obwohl es häufig von Cloud-Anbietern bereitgestellt wird.
  - Stellt den Ingress-Controller-Dienst (Ingress-Nginx) mit dem Diensttyp bereit. LoadBalancer Dies setzt voraus, dass der Kubernetes-Cluster auf einer Plattform läuft, die externe Load Balancer unterstützt.
- Bestehendes ALB
  - Verbindet den Ingress-Controller mit einem vorhandenen ALB.
  - Sie müssen den vorhandenen Application Load Balancer Balancer-Zielgruppen-ARN ARN.
- Bestehender NLB
  - Verbindet den Ingress-Controller mit einem vorhandenen NLB.
  - Sie müssen den vorhandenen Network Load Balancer Target Group ARN angeben.
- NodePort
  - Der Ingress-Controller (ingress-nginx) wird so konfiguriert, dass er den NodePort Diensttyp verwendet, der einen Port auf allen Knoten im Kubernetes-Cluster öffnet und den Datenverkehr an den Ingress weiterleitet. Der Client-Verkehr kann dann entweder über DNS oder einen externen Load Balancer an diese Knoten weitergeleitet werden.
  - Sie können einen Portbereich zwischen 1 und 65535 wählen, oder es wird ein zufälliger Port zwischen 30000 und 32767 verwendet.

- Ingress
  - Bringen Sie Ihren eigenen Ingress-Controller mit. Diese Konfiguration akzeptiert einen Ingress-Klassennamen, den die Dienste dann in ihren Ingress-Manifesten verwenden. Dies bedeutet, dass der Ingress-Controller über eine gewisse externe Konnektivität verfügt, die bereits über einen anderen Lastausgleichsmechanismus konfiguriert wurde.
  - Derzeit wird nur der Ingress-Nginx-Controller unterstützt.

Hostname mit Platzhalter

Standardmäßig werden Ingress-Routen mit dem Hostwert `*` definiert. Deaktivieren Sie diese Einstellung, um den definierten Hostnamen für den Wickr Enterprise Server zu verwenden. Für IP-basierte Hostnamen ist ein Wildcard-Hostname erforderlich.

Datenbank-Einstellungen

Wickr Enterprise benötigt eine MySQL 8.0-Datenbank. Wenn Sie MySQL 5.7 verwenden, finden Sie weitere Informationen unter Upgrade auf MySQL 8.0 Upgrade. Wir empfehlen die Verwendung einer externen Datenbank zu Ihrem Kubernetes-Cluster, z. B. Amazon RDS. Sie haben jedoch auch die Möglichkeit, als Teil der Installation eine interne MySQL-Datenbank innerhalb des Kubernetes-Clusters bereitzustellen.

Einstellungen für externe Datenbanken

- Hostname: Hostname oder IP-Adresse des Datenbankservers.
- Hostname des Lesers: Hostname oder IP-Adresse eines schreibgeschützten Endpunkts für den Datenbankserver (falls verfügbar).
- Port: Der Port, über den auf MySQL zugegriffen wird.
- Datenbankname: Der Name der Datenbank, die auf dem Server erstellt wurde.
- Benutzername: Der Benutzer, der berechtigt ist, auf die Datenbank zuzugreifen.
- Passwort: Das Passwort für diesen Benutzer.
- CA-Zertifikat: Ein PEM-Zertifikat für die Verbindung zur Datenbank über TLS.

> **Note**
>
> Stellen Sie sicher, dass Ihre MySQL-Installation den Standardzeichensatz latin1 mit der Sortierung latin1_swedish_ci verwendet. Dies kann erreicht werden, indem Sie überprüfen, ob Ihr MySQL-Server mit den folgenden Flags gestartet wurde: `"--character-set-server latin1", "--collation-server latin1_swedish_ci"`

Interne Datenbankeinstellungen

Der interne Datenbanktyp stellt zwei StatefulSets in Ihrem Cluster für eine primäre und eine sekundäre MySQL-Datenbank mit binärer Replikation bereit. Die sekundäre Datenbank empfängt keinen Datenverkehr und ist nur für Disaster Recovery und Backups verfügbar.

Speichergröße: Größe (in Gibibyte) der persistenten Volumes für die Datenbank-Pods.

Erhöhung der MySQL-Speichergröße

> **Note**
>
> Ihr Volumetyp StorageClass muss die Volumenerweiterung unterstützen, um die Speichergröße zu erhöhen. Weitere Informationen finden Sie unter Volumenerweiterung.

Die in Wickr Enterprise verwendeten MySQL-Dienste werden als StatefulSet Ressourcen in Kubernetes bereitgestellt. StatefulSets machen Sie viele Eigenschaften der Ressource unveränderlich, einschließlich der Persistent Volume Claim-Vorlagen. Um die Unveränderlichkeit von zu umgehen, müssen die folgenden Aktionen ausgeführt werden StatefulSets, um die Größe der von MySQL verwendeten Volumes zu erhöhen.

1. Bearbeiten Sie die persistenten Volumenansprüche für und. `data-mysql-primary-0` `data-mysql-secondary-0`

    1. `kubectl -n wickr edit pvc data-mysql-primary-0.` Set `spec.resources.requests.storage`auf die gewünschte Speichergröße.
    2. `kubectl -n wickr edit pvc data-mysql-secondary-0.` Set `spec.resources.requests.storage`auf die gewünschte Speichergröße.

2. Löschen Sie die vorhandenen StatefulSets, aber lassen Sie die Pods bestehen, indem Sie die `--cascade=orphan` Markierung weitergeben.

   `kubectl -n wickr delete statefulset --cascade=orphan mysql-primary mysql-secondary`.

3. Aktualisieren Sie in der KOTS-Benutzeroberfläche die Einstellung Speichergröße so, dass sie dem Wert entspricht, den Sie in Schritt 1 festgelegt haben. Speichern Sie diese Konfiguration und stellen Sie sie bereit.
4. Starten Sie den neu StatefulSets , um die Volumes zu erweitern und die MySQL-Dienste wieder online zu bringen.

   `kubectl -n wickr rollout restart statefulset mysql-primary mysql-secondary`.

Upgrade auf MySQL 8.0

Externe Datenbank (RDS)

Gehen Sie wie folgt vor, um Wickr Backend offline zu schalten.

1. Suchen Sie den Namespace von Ingress `kubectl get deployments --all-namespaces`

   Im folgenden Beispiel ist der Namespace Wickr und replicas ist 3.

```
NAMESPACE     NAME                       READY   UP-TO-DATE   AVAILABLE   AGE
...
wickr         ingress-nginx-controller   3/3     3            3           43h
...
```

2. Reduzieren Sie den Ingress `kubectl scale deployment/ingress-nginx-controller --replicas=0 -n wickr`
3. Erstellen Sie einen Snapshot, um die Datenbank zu sichern. Weitere Informationen finden Sie unter Manuelle Backups verwalten im Amazon Relational Database Service Service-Benutzerhandbuch.
4. Aktualisieren Sie die Engine-Version auf MySQL 8.0.x (MySQL 8.4 wird nicht unterstützt). Weitere Informationen finden Sie unter Upgraden einer DB-Instance-Engine-Version im Amazon Relational Database Service Service-Benutzerhandbuch.

Um Wickr Backend online zu bringen, reduzieren Sie den Ingress `kubectl scale deployment/ingress-nginx-controller --replicas=3 -n wickr`

Interne Datenbank

Weitere Informationen finden Sie unter MySQL Backup und wiederherstellen.

S3-Dateispeicher

Wickr Enterprise benötigt einen S3-kompatiblen Speicherdienst. Wir empfehlen, einen S3-Service zu verwenden, der sich außerhalb Ihres Kubernetes-Clusters befindet, z. B. Amazon S3. Sie haben jedoch auch die Möglichkeit, als Teil der Installation einen internen S3-Service innerhalb des Kubernetes-Clusters bereitzustellen.

Externe S3-Einstellungen

- Bucket-Name: Der Name des S3-Buckets, in dem Datei-Uploads gespeichert werden.
- Region: Die AWS Region des S3-Buckets.
- Endpunkt: Legen Sie den Endpunkt fest, den Wickr für die Interaktion mit der S3-API verwenden wird. Standardmäßig wird der S3-Serviceendpunkt der Region verwendet.
- Fileproxy-Service-Kontoname: Nur Amazon S3. Der Name eines vorhandenen Kubernetes-Dienstkontos, das für die Authentifizierung bei S3 mithilfe von IAM-Rollen für Dienstkonten verwendet werden soll.
- Externer S3-Zugriffsschlüssel: Dies ist Ihr vorhandener S3-Zugriffsschlüssel.
- Externer geheimer S3-Schlüssel: Dies ist Ihr vorhandener geheimer S3-Schlüssel.

Interne S3-Einstellungen

Der interne S3-Typ stellt standardmäßig 4 MiniO-Server-Pods bereit, die jeweils 4 Persistent Volume Claims enthalten. Die Standardkonfiguration verwendet MinIO Erasure Coding, um die Fehlertoleranz zu erhöhen.

- Interne S3-Serveranzahl: Die Anzahl der zu erstellenden MiniO-Server-Pods. Die Standardeinstellung ist 4 für eine fehlertolerante Bereitstellung. Dieser Wert kann für eine development/test Bereitstellung auf einen niedrigen Wert von 1 festgelegt werden.
- Anzahl interner S3-Volumes: Die Anzahl der MiniO-Volumes, die in jedem MiniO-Server-Pod erstellt werden sollen. Die Standardeinstellung ist 4 für eine fehlertolerante Bereitstellung. Dieser

Wert kann für eine Bereitstellung auf einen development/test niedrigen Wert von 1 festgelegt werden.

- Größe des internen S3-Volumes: Die Größe der in den MiniO-Server-Pods erstellten MiniO-Volumes in GB. Die Standardgröße ist 10 GB.
- Bei einer standardmäßigen internen S3-Bereitstellung werden 4 Server mit 4 verwendet. PVCs Jedes PVC hat 10 Gi, was 160 Gi Raw-Speicher ergibt, wobei Benutzern 120 Gi Erasure Coded Speicher zur Verfügung stehen.
- Minio Erasure Coding Calculator ist verfügbar. Weitere Informationen finden Sie unter Erasure Code Calculator.

Einstellungen für persistente Volumenansprüche

Wickr Enterprise benötigt Persistent Volume Claims, um statusbehaftete Daten zu speichern. Mit dieser Einstellung können Sie den Namen oder den Namen der Speicherklasse angeben, die Sie verwenden möchten. Wenn das Feld leer gelassen wird, versucht Wickr, die Standard-Speicherklasse zu verwenden. Das Ändern der Speicherklasse nach der Bereitstellung von Wickr wird nicht unterstützt.

Eine Standardeinstellung StorageClass für Persistent Volume Claims wird häufig von Cloud-Anbietern bereitgestellt. Bei vollständig lokalen Installationen kann jedoch eine explizite Konfiguration mithilfe eines Drittanbieterdienstes wie Longhorn erforderlich sein.

Einstellungen für das TLS-Zertifikat

Laden Sie ein PEM-Zertifikat und einen privaten Schlüssel zum Beenden von TLS hoch. Der alternative Name des Antragstellers auf dem Zertifikat muss mit dem Hostnamen übereinstimmen, der in den Einstellungen Ihrer Wickr Enterprise-Bereitstellung konfiguriert wurde.

Verketten Sie für das Feld „Zertifikatskette“ vor dem Hochladen alle Zwischenzertifikate (falls erforderlich) mit dem Root-CA-Zertifikat.

Let's Encrypt

Wählen Sie diese Option, um mithilfe von Let's Encrypt automatisch ein Zertifikat zu generieren. Zertifikate werden mithilfe der HTTP-01-Challenge über den cert-manager-Operator ausgestellt.

Die HTTP-01-Herausforderung erfordert, dass der gewünschte DNS-Name zum Eingangspunkt für Ihren Cluster (normalerweise ein Load Balancer) aufgelöst wird und der Datenverkehr zum TCP-Port

80 öffentlich zugänglich ist. Diese Zertifikate sind kurzlebig und werden regelmäßig erneuert. Es ist notwendig, Port 80 offen zu halten, damit die Zertifikate automatisch erneuert werden können.

> **Note**
>
> Dieser Abschnitt bezieht sich ausdrücklich auf das Zertifikat, das von der Wickr Enterprise-Anwendung selbst verwendet wird.

Angeheftetes Zertifikat

Wickr Enterprise erfordert das Pinning von Zertifikaten, wenn selbstsignierte Zertifikate oder Zertifikate verwendet werden, denen Client-Geräte nicht vertrauen. Wenn das von Ihrem Load Balancer vorgelegte Zertifikat selbst signiert ist oder von einer anderen Zertifizierungsstelle als der Wickr Enterprise-Installation signiert wurde, laden Sie das CA-Zertifikat hier hoch, damit sich die Clients stattdessen daran anheften können.

In den meisten Situationen ist diese Einstellung nicht erforderlich.

Anbieter von Zertifikaten

Wenn Sie vorhaben, ein Zertifikat zur Verwendung mit Wickr Enterprise zu erwerben, finden Sie unten eine Liste von Anbietern, deren Zertifikate bekanntermaßen standardmäßig ordnungsgemäß funktionieren. Wenn ein Anbieter unten aufgeführt ist, wurden seine Zertifikate explizit mit der Software validiert.

- Digicert
- Schnelles SSL

Generieren eines selbstsignierten Zertifikats

Wenn Sie Ihr eigenes selbstsigniertes Zertifikat für die Verwendung mit Wickr Enterprise erstellen möchten, enthält der folgende Beispielbefehl alle erforderlichen Flags für die Generierung.

```
openssl req -x509 -newkey rsa:4096 -sha256 -days 365 -nodes -keyout $YOUR_DOMAIN.key -
out $YOUR_DOMAIN.crt -subj "/CN=$YOUR_DOMAIN" -addext "subjectAltName=DNS:$YOUR_DOMAIN"
 -addext "extendedKeyUsage = serverAuth"
```

Wenn Sie ein IP-basiertes, selbstsigniertes Zertifikat erstellen möchten, verwenden Sie stattdessen den folgenden Befehl. Um das IP-basierte Zertifikat zu verwenden, stellen Sie sicher, dass das Feld Wildcard Hostname in den Ingress-Einstellungen aktiviert ist. Weitere Informationen finden Sie unter Ingress-Einstellungen.

```
openssl req -x509 -newkey rsa:4096 -sha256 -days 365 -nodes -keyout $YOUR_DOMAIN.key -
out $YOUR_DOMAIN.crt -subj "/CN=$YOUR_DOMAIN" -addext "subjectAltName=IP:$YOUR_DOMAIN"
 -addext "extendedKeyUsage = serverAuth"
```

> **Note**
>
> Ersetzen Sie $YOUR_DOMAIN im Beispiel durch den Domainnamen oder die IP-Adresse, die Sie verwenden möchten.

Einstellungen aufrufen

- Anrufende Knoten erforderlich: Wenn diese Einstellung aktiviert ist, werden die Anrufdienste von Wickr nur auf Kubernetes-Knoten mit dem Label bereitgestellt. `role=calling` Deaktivieren Sie diese Einstellung, um Anruf- und Messaging-Dienste auf denselben Knoten oder für Bereitstellungen mit nur einem Knoten bereitzustellen.

  In der Regel sollten Sie auch den aufrufenden TCP-Proxy deaktivieren, wenn diese Einstellung deaktiviert ist, da der TCP-Proxydienst auf Port 443 ausgeführt wird.
- TCP-Proxy aktivieren: Diese Einstellung steuert, ob der Dienst für den TCP-Fallback-Modus bei Anrufen bereitgestellt wird oder nicht. Deaktivieren Sie diese Einstellung, wenn andere Dienste auf 443/tcp laufen oder wenn Sie keinen TCP-Fallback-Modus für Anrufe benötigen. Dies muss für Bereitstellungen aktiviert sein, die Wickr Open Access verwenden möchten.
- Öffentliche Server-IP-Adressen automatisch erkennen: Wenn diese Einstellung aktiviert ist, ermitteln die anrufenden Dienste ihre öffentliche IP-Adresse, indem sie HTTPS-Anfragen an https://ipv4.icanhazip.com/ und stellen. https://ipv6.icanhazip.com/ Wenn diese Option deaktiviert ist, müssen Sie die Einstellung „Primäre Host-IP-Adresse für den Anrufverkehr verwenden“ oder „Hostname überschreiben“ aktivieren. Andernfalls können die Anrufdienste nicht gestartet werden.
- Primäre Host-IP-Adresse für den Anrufverkehr verwenden: Verwenden Sie die primäre IP-Adresse der Kubernetes-Knoten, um Dienste aufzurufen. Dies bedeutet, dass alle Wickr-Clients über die primäre IP-Adresse des Knotens eine Verbindung zu Ihren Kubernetes-Knoten herstellen können, wie unter Aus der Downward API beschrieben. `status.hostIP`

- Hostname Override: Geben Sie einen Hostnamen oder eine IP-Adresse an, die als Verbindungspunkt für Calling-Dienste zurückgegeben werden soll. Diese Einstellung sollte nur verwendet werden, wenn ein einzelner aufrufender Server ausgeführt wird, da für alle Replikate des Dienstes derselbe Wert zurückgegeben wird. Wenn eine Hostnamen-Außerkraftsetzung festgelegt und die Einstellung „Primäre Host-IP-Adresse verwenden“ aktiviert ist, hat die Einstellung für die primäre IP-Adresse des Hosts Vorrang.
- Anrufen des Host-Netzwerks aktiviert: Standardmäßig verwenden aufrufende Pods das Host-Netzwerk der Knoten für die Konnektivität. Deaktivieren Sie dies, um einen NodePort Dienst für den Anrufverkehr verfügbar zu machen. Wenn das Aufrufen von eingehendem Datenverkehr aktiviert ist, stellen Sie sicher, dass ein geeigneter Dienst so konfiguriert ist, dass er eingehenden Datenverkehr zulässt. Dies muss aus Gründen der STIG-Konformität deaktiviert sein.

Ingress-Einstellungen aufrufen

Wickr unterstützt eine Einstellung für eingehende Anrufe, sodass ein Client eine Verbindung zu einem beliebigen aufrufenden Knoten innerhalb des Clusters herstellen und die Anrufroute an den richtigen aufrufenden Server weiterleiten kann. Wickr unterstützt vier Arten von eingehenden Anrufen:

- LoadBalancer (Standard)
  - LoadBalancer Sie werden vom Cloud-Anbieter bereitgestellt (vollständige Installationen vor Ort erfordern eine zusätzliche Konfiguration). Nach der LoadBalancer Bereitstellung muss die KOTS-Konfiguration erneut aktualisiert werden, um den Hostnamen oder die IP-Adressen des Load Balancers bereitzustellen.
- NodePort
  - Macht auf jedem aufrufenden Knoten einen NodePort Dienst verfügbar, der als Einstiegspunkt für den Anrufverkehr dient. Ein Hostname, der in einen oder mehrere Knoten aufgelöst wird, oder die IP-Adresse eines oder mehrerer Knoten muss angegeben werden. Sie können einen Portbereich von 30000-32767 für UDP- und optional TCP-Verkehr wählen.
- Bestehender NLB
  - Hängt den Calling Ingress Service an einen vorhandenen NLB an. Sie müssen den Zielgruppen-ARN für UDP und optional TCP-Verkehr angeben.
- Kein Dienst
  - Wählen Sie diese Option, wenn Sie keinen zusätzlichen Kubernetes-Dienst benötigen, um eingehenden Datenverkehr zuzulassen. Dies wird in der Regel zusammen mit der Host-

Netzwerkeinstellung verwendet, um eingehenden Anrufverkehr direkt an Ihre aufrufenden Knoten weiterzuleiten.

Überlegungen

- Um die Abwärtskompatibilität mit älteren Clients und Verbundnetzwerken ohne Call-Ingress zu gewährleisten, ist der alte Anrufmodus weiterhin verfügbar (direkte Verbindung zu den aufrufenden Servern), wenn Calling Ingress aktiviert ist. Wenn Sie Standardports ändern, stellen Sie sicher, dass es keine Portkollisionen auf den aufrufenden Knoten gibt.
- Der NLBs Dual-Stack-Dienst für UDP-Verkehr muss IPv6 Backend-Ziele haben. Weitere Informationen finden Sie unter Network Load Balancer Balancer-Zielgruppen.
- Wenn Sie STIG-Konformität benötigen, müssen Sie die Host-Netzwerkoption für Anrufe deaktivieren. Wenn die Knoten im Dual-Stack-Modus konfiguriert sind, der Cluster jedoch nicht, können Sie die IPv6 Konnektivität verlieren (vorausgesetzt, es handelt sich um einen IPv4 Cluster).
- Für das Aufrufen von Ingress sind vordefinierte Hostnamen oder IP-Adressen erforderlich. Die Skalierung von Knoten oder die Bereitstellung von benutzerdefiniertem Routing kann eine Änderung der Konfiguration erfordern.
- Die Standardports für eingehende Anrufe sind 8443 für TCP und 16384 für UDP. Stellen Sie sicher, dass Firewalls und Sicherheitsgruppen den Verkehr für diese Ports oder alternative Ports zulassen, falls die Standardwerte außer Kraft gesetzt werden.

Referenzarchitekturen

Eingang mit Load Balancer

Diese Option macht einen einzelnen Load Balancer als Einstiegspunkt für den gesamten Anrufverkehr verfügbar.

1. Wählen Sie für Calling Ingress Type entweder Load Balancer oder Existing NLB aus. Weitere Informationen zu Existing NLB finden Sie im Wickr Enterprise CDK Sample on zum NLB-Stack. GitHub
2. Führen Sie je nach Typ des aufrufenden Eingangs einen der folgenden Schritte aus:
   - Geben Sie für Existing NLB die Zielgruppe ARNs für UDP- und TCP-Verkehr und den Hostnamen des NLB an.

- Geben Sie für Load Balancer den Hostnamen an, nachdem er von Kubernetes bereitgestellt wurde.

Alternativ können Sie für jeden Calling Ingress Type die IP-Adressen des Load Balancers oder einen benutzerdefinierten Hostnamen angeben, der auf den Load Balancer verweist.

3. (Optional) Um Nachrichten- und Anrufverkehr unter einem einzigen NLB zu kombinieren, wählen Sie im Abschnitt Ingress die Option Existing NLB aus und geben Sie eine HTTPS-Zielgruppe an.

Eingang mit NodePort

Diese Option ist nützlich, wenn das Host-Netzwerk deaktiviert ist und Sie keinen zusätzlichen Load Balancer verfügbar machen möchten.

> **Note**
>
> Stellen Sie sicher, dass Ihre Firewalls und Sicherheitsgruppen Datenverkehr für die zulassen. NodePorts

1. Wählen Sie für Calling Ingress Type die Option. NodePort
2. Fügen Sie die Hostnamen oder IP-Adressen des aufrufenden Knotens hinzu.
3. Deaktivieren Sie das Calling Host Network.

Direkter Zugang mit HostNetwork

Diese Option macht keinen zusätzlichen Kubernetes-Dienst verfügbar und ermöglicht es, eingehenden Datenverkehr aufzurufen, um eine direkte Verbindung über das Host-Netzwerk der aufrufenden Knoten herzustellen. Dieser Ansatz wird bevorzugt, wenn Konnektivität erforderlich ist. IPv6

1. Wählen Sie für Calling Ingress Type die Option Kein Dienst aus.
2. Fügen Sie die Hostnamen oder IP-Adressen des aufrufenden Knotens hinzu.
3. Aktivieren Sie das Calling Host Network.

Kubernetes-Cluster-Autoscaler (optional)

Kubernetes Cluster Autoscaler ist ein optionaler Konfigurationswert für die Wickr Enterprise-Installation. Er hilft Ihnen bei der Skalierung Ihrer Kubernetes-Knotengruppen im Falle eines erhöhten Datenverkehrs oder anderer Ressourceneinschränkungen, die zu einer schlechten Leistung führen könnten.

Die Wickr Enterprise-Installation unterstützt drei Cloud-Anbieter-Integrationen: Google Cloud AWS und Azure. Jeder Cloud-Anbieter hat unterschiedliche Anforderungen für diese Integration. Bitte folgen Sie den Anweisungen für Ihren spezifischen Cloud-Anbieter unten, um diese Funktion zu aktivieren.

AWS

Wenn Sie das WickrEnterprise CDK nicht zur Installation Ihrer Wickr-Umgebung verwendet haben AWS, müssen Sie einige zusätzliche Schritte ausführen, um den Cluster Autoscaler zu aktivieren.

1. Fügen Sie die folgenden Tags zu Ihren Knotengruppen hinzu. Dadurch kann der Cluster Autoscaler die entsprechenden Knoten automatisch erkennen.

    1. `k8s.io/cluster-autoscaler/`**`clusterName`** `= owned`wobei ClusterName der Name Ihres Kubernetes-Clusters ist
    2. `k8s.io/cluster-autoscaler-enabled = true`
2. Fügen Sie ein Kubernetes-Dienstkonto im Kube-System-Namespace hinzu und verknüpfen Sie es mit einer IAM-Richtlinie, die Autoscaling- und EC2-Aktionen ermöglicht. Weitere Informationen und detaillierte Anweisungen finden Sie unter Konfiguration eines Kubernetes-Servicekontos für die Übernahme einer IAM-Rolle im Amazon EKS-Benutzerhandbuch.

    1. Bei der Einrichtung des Servicekontos müssen Sie den Namespace „kube-system“ verwenden
    2. Die folgende Richtlinie kann für das Dienstkonto verwendet werden:

        JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances",
```

```
                "autoscaling:DescribeLaunchConfigurations",
                "autoscaling:DescribeTags",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:TerminateInstanceInAutoScalingGroup",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstances",
                "ec2:DescribeLaunchTemplateVersions"
              ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Wählen Sie AWSin der replizierten Benutzeroberfläche bei der Konfiguration des Cluster-Autoscalers Ihren Cloud-Anbieter aus und geben Sie den Namen des Dienstkontos ein, das Sie oben erstellt haben, um den Cluster Autoscaler anzuweisen, dieses Dienstkonto zu verwenden.

Google Cloud

Es wird dringend empfohlen, die integrierten Autoscaling-Funktionen von GKE sowohl für Autopilot- als auch für Standardcluster zu verwenden. Wenn Sie jedoch mit dieser Integration fortfahren möchten, müssen die folgenden Anforderungen erfüllt sein, bevor Sie fortfahren können.

Voraussetzungen:

1. Die Managed Instance Groups (MIG) müssen mit einem Sicherheitsbereich erstellt werden, der mindestens „Lesen/Schreiben“ für Compute Engine-Ressourcen umfasst. Dies kann derzeit nicht zu einem späteren Zeitpunkt zur MIG hinzugefügt werden.
2. Für den Cluster muss Workload Identity Federation aktiviert sein. Sie können dies auf einem vorhandenen Cluster aktivieren, indem Sie Folgendes ausführen: `gcloud container clusters update ${CLUSTER_NAME} --workload-pool= ${PROJECT_ID}.svc.id.goog`
3. Ein Google Cloud Platform (GCP) -Dienstkonto mit Zugriff auf die Rolle `roles/compute.instanceAdmin.v1`. Dieses kann mit den folgenden Anweisungen erstellt werden:

```
# Create GCP Service Account
gcloud iam service-accounts create k8s-cluster-autoscaler
```

```
# Add role to GCP Service Account
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
--member "serviceAccount:k8s-cluster-autoscaler@${PROJECT_ID}.iam.gserviceaccount.com"
 \
--role "roles/compute.instanceAdmin.v1"

# Link GCP Service Account to Kubernetes Service Account
gcloud iam service-accounts add-iam-policy-binding k8s-cluster-autoscaler@
${PROJECT_ID}.iam.gserviceaccount.com \
--role roles/iam.workloadIdentityUser \
--member "serviceAccount:${PROJECT_ID}.svc.id.goog[kube-system/cluster-autoscaler-gce-
cluster-autoscaler]"
```

Azure

Azure Kubernetes Service (AKS) bietet integrierte Cluster-Autoskalierung für die meisten Bereitstellungen. Es wird dringend empfohlen, diese Methoden für die automatische Clusterskalierung zu verwenden. Wenn Ihre Anforderungen jedoch so sind, dass diese Methoden nicht funktionieren, haben wir eine Kubernetes Cluster Autoscaler-Integration für Azure Kubernetes Service bereitgestellt. Um diese Integration nutzen zu können, müssen Sie die folgenden Informationen sammeln und sie in der Konfiguration des KOTS-Admin-Panels unter Cluster Autoscaler eingeben, nachdem Sie Azure als Ihren Cloud-Anbieter ausgewählt haben.

Azure-Authentifizierung

Abonnement-ID: Die Abonnement-ID kann über das Azure-Portal abgerufen werden, indem Sie der offiziellen Dokumentation folgen. Weitere Informationen finden Sie unter Abonnement und Mandant abrufen IDs im Azure-Portal.

Die folgenden Parameter können abgerufen werden, indem ein AD Service Principal mit dem az-Befehlszeilenprogramm erstellt wird.

```
az ad sp create-for-rbac –role="Contributor" –scopes="/subscriptions/subscription-id" –
output json
```

App-ID:

Kunden-Passwort:

Mandanten-ID:

Konfiguration von Azure Cluster Autoscaler

Zusätzlich zu den Authentifizierungsanforderungen sind die folgenden Felder erforderlich, damit der Cluster-Autoscaler ordnungsgemäß funktioniert. Befehle zum Abrufen dieser Informationen wurden der Einfachheit halber bereitgestellt. Sie können jedoch je nach Ihrer spezifischen AKS-Konfiguration einige Änderungen erfordern.

Azure Managed Node Resource Group: Dieser Wert ist die verwaltete Ressourcengruppe, die von Azure erstellt wurde, als Sie den AKS-Cluster eingerichtet haben, und nicht die Ressourcengruppe, die Sie definiert haben. Um diesen Wert zu erhalten, benötigen Sie CLUSTER_NAME und RESOURCE_GROUP, die bei der Erstellung des Clusters angegeben wurden. Sobald Sie diese Werte haben, können Sie sie abrufen, indem Sie Folgendes ausführen:

```
az aks show –resource-group ${RESOURCE_GROUP} –name ${CLUSTER_NAME} –query
 nodeResourceGroup -o tsv
```

VMSS-Name des Anwendungsknotenpools: Dies ist der Name des Virtual Machine Scaling Sets (VMSS), das Ihrem AKS-Knotenpool für die Wickr-Anwendung zugeordnet ist. Dies ist die Ressource, die je nach den Anforderungen Ihres Clusters nach oben oder unten skaliert wird. Um diesen Wert zu erhalten, können Sie den folgenden az-Befehl ausführen:

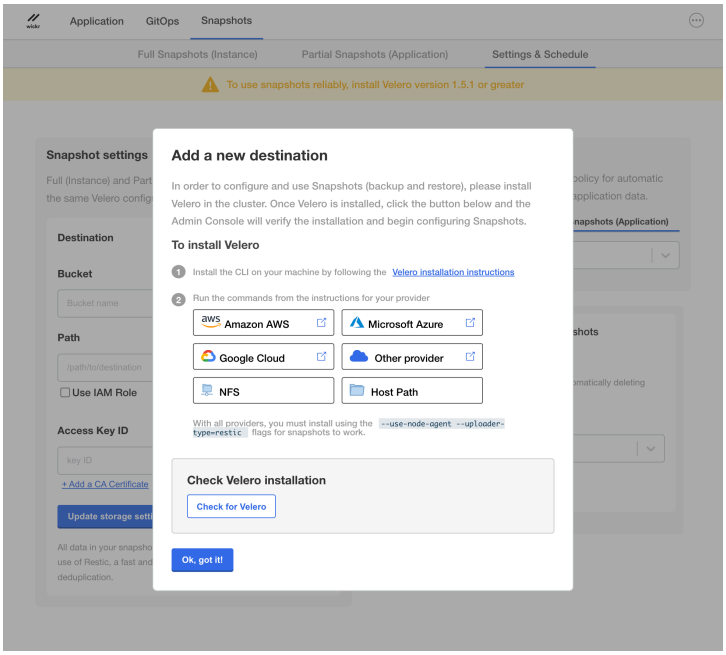
```
CLUSTER_NODEPOOL_NAME="(Your-NodePool-Name)"
CLUSTER_RESOURCE_GROUP="(Your-Managed-Node-Resource-Group-As-Defined-Above>)"
az vmss list -g ${CLUSTER_RESOURCE_GROUP} --query '[?tags."aks-managed-
poolName"==`'''${CLUSTER_NODEPOOL_NAME}'''`].{VMSS_name:name}' -o tsv
```

ACalling VMSS-Name des Knotenpools (optional): Dies ist der Name des VMSS, der Ihrem aufrufenden Nodepool zugeordnet ist, falls Sie einen haben. Um diesen Wert zu erhalten, können Sie eine modifizierte Version des Befehls für den VMSS-Name des Anwendungsknotenpools ausführen und dabei den Wert CLUSTER_NODEPOOL_NAME für den Namen des Nodepools für Ihren aufrufenden Nodepool ausschalten.

Sicherungen

Wickr Enterprise verwendet Velero für Backup-Zwecke. Velero bietet die notwendigen Tools für die Sicherung und Wiederherstellung von Kubernetes-Cluster-Ressourcen und persistenten Volumes, unabhängig davon, ob sie bei einem Cloud-Anbieter oder vor Ort betrieben werden.

Velero-Backups mit Minio: Derzeit sind Velero-Backups nur für Minio im Low Resource Mode aktiviert.



Installation mithilfe der Velero-Dokumentation

- Installieren Sie die Velero CLI. Weitere Informationen finden Sie unter Installation der Velero CLI.
- Installieren Sie Velero auf Ihrem Cluster und konfigurieren Sie den Speicher entsprechend Ihrem Anbieter:
  - AWS.
  - GCP.
  - Azurblau.
  - Andere Anbieter.

Einschränkung

Standardmäßig sind keine Volumes in der Sicherung enthalten. Wenn Pods ein Volume bereitstellen, das gesichert werden soll, müssen Sie das Backup mit einer Anmerkung konfigurieren, in der die spezifischen Volumes aufgeführt sind, die in das Backup aufgenommen werden sollen.

Fügen Sie für jedes Volume, für das ein Backup erforderlich ist, die Datei backup.velero hinzu. io/ backup-volumes annotation. The annotation name is backup.velero.io/backup-volumes und der Wert ist eine durch Kommas getrennte Liste von Volumes, die in das Backup aufgenommen werden sollen. Weitere Informationen finden Sie unter Snapshots konfigurieren.

Installation von Airgap

Wickr Enterprise und KOTS unterstützen beide die Bereitstellung in einem vollständig mit Airgap ausgestatteten Kubernetes-Cluster. Sie müssen Zugriff auf eine private Docker-Image-Registry gewähren, die vom Airgapped Kubernetes-Cluster aus erreichbar ist. Die an KOTS gelieferte Private Docker Image Registry muss mit username/password Authentifizierung gesichert sein, damit sie zu diesem Zweck ordnungsgemäß funktioniert. KOTS wird die Private Docker Image Registry verwenden, um alle Wickr Enterprise-Images zu hosten.

- Wickr Enterprise license.yaml mit aktiviertem Airgap (Wenden Sie sich an das Wickr-Vertriebs- oder Kundendienstteam)
- Wickr Enterprise wickr.airgap Archivpaket (Wenden Sie sich an das Wickr-Vertriebs- oder Kundendienstteam)
- Zugriff auf eine private Docker-Image-Registry.
- Zugriff auf einen Kubernetes-Cluster, der in der Airgap-Umgebung bereitgestellt wird.
- Kubectl ist installiert.
- KOTS CLI installiert.
- kotsadm.tar.gz wurde heruntergeladen.

Führen Sie die folgenden Befehle aus, um KOTS und Wickr Enterprise auf Ihrem Airgapped-Kubernetes-Cluster bereitzustellen. Diese Befehle laden die KOTS-Admin-Images und die Wickr Enterprise-Images in die Private Docker Image Registry hoch. Nach Abschluss der Befehle werden Sie aufgefordert, auf die KOTS Admin Console zuzugreifen, um die Wickr Enterprise-Installation wie oben beschrieben abzuschließen.

```
kubectl kots admin-console push-images \
  ~/kotsadm.tar.gz $PRIVATE_REGISTRY_HOST \
  --registry-username $PRIVATE_REGISTRY_USER \
  --registry-password $PRIVATE_REGISTRY_PASSWORD

kubectl kots install wickr \
  --license-file ~/YOUR_LICENSE.yaml \
  --airgap-bundle ~/wickr.airgap \
  --kotsadm-registry $PRIVATE_REGISTRY_HOST \
  --registry-username $PRIVATE_REGISTRY_USER \
  --registry-password $PRIVATE_REGISTRY_PASSWORD
```

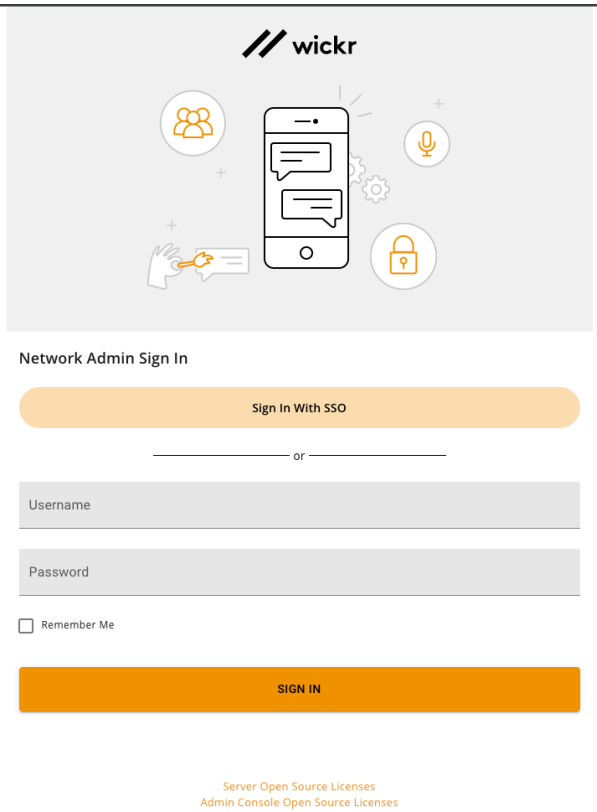
Mobile Benachrichtigung für Airgap-Installationen

Für Push-Benachrichtigungen vom Server-Backend an mobile Clients sind zusätzliche Netzwerkzulassungslisten erforderlich. Diese Anforderung ist darauf zurückzuführen, wie Apple iOS und Google Android diese Funktion für Offline- und Hintergrundgeräte implementieren. Lesen Sie in der Dokumentation zu diesen Diensten nach und listen Sie die angegebenen IP-Adressen und Ports auf.

- iOS
- Android

Wickr Admin-Konsole

Die Wickr Admin Console-Oberfläche wird für die Verwaltung der Wickr Enterprise-Anwendung selbst verwendet. Sie kann verwendet werden, um Netzwerke, Benutzer, Verbände und mehr einzurichten. Es ist über HTTPS unter dem DNS-Namen zugänglich, den Sie so konfiguriert haben, dass er auf Ihren Load Balancer verweist. Der Standardbenutzername ist admin mit dem Passwort Password123. Sie müssen dieses Passwort bei der ersten Anmeldung ändern.



Sicherheitseinstellungen

AWS Wickr Enterprise bietet Konfigurationseinstellungen, um einen erweiterten Sicherheitskontext für Ihre Bereitstellung durchzusetzen. Dieser höhere Sicherheitsstandard wird auf Pod- und Container-Ebene angewendet und ist für die Einhaltung des Security Technical Implementation Guide (STIG) erforderlich.

Legen Sie die folgenden Konfigurationsparameter fest, um den erweiterten Sicherheitskontext durchzusetzen:

```
podSecurityContext:
  runAsNonRoot: true
  seccompProfile:
    type: RuntimeDefault
containerSecurityContext:
  allowPrivilegeEscalation: false
  capabilities:
    drop: ["ALL"]
```

> **Warning**
>
> Für Opensearch deaktiviert diese Sicherheitskonfiguration den `fsgroup-volume` InitContainer, der die Berechtigungen für den persistenten Speicher aktualisiert, was zu Kompatibilitätsproblemen im Zusammenhang mit Berechtigungen führen kann.

Häufig gestellte Fragen

F: Meine Bereitstellung schlägt mit dem folgenden Fehler in helm stderr fehl:

`Error: UPGRADE FAILED: cannot patch "enterprise-init" with kind Job: Job.batch "enterprise-init" is invalid: spec.template: Invalid value: core.`

A: Dies kann passieren, wenn die Debug-Protokollierung aktiviert ist. Bitte deaktivieren Sie die Debug-Protokollierung, löschen Sie die problematischen Jobs und versuchen Sie es erneut.

Eingebetteter Cluster für Wickr Enterprise

Die eingebettete Cluster-Installationsoption für Wickr Enterprise bietet ein kleines, effizientes Installationsangebot für das Wickr Enterprise-Produkt. Es nutzt den Replicated Embedded Cluster, um mithilfe von k0s eine kleine Kubernetes-Installation bereitzustellen, auf der Wickr Enterprise installiert werden kann. Durch die Verwendung dieser Installationsmethode werden die technischen Anforderungen sowie die allgemeinen Hardwareanforderungen für eine Wickr Enterprise-Installation minimiert, da eine „all-in-one“ Lösung auf Kosten der Ausfallsicherheit und Hochverfügbarkeit bereitgestellt wird.

Topics

Erste Schritte mit Wickr Enterprise Embedded Cluster

Um mit der Nutzung der Wickr Enterprise Embedded Cluster-Option zu beginnen, wenden Sie sich an den Support, um eine Lizenz zu erhalten. Wenn Sie über eine bestehende Lizenz verfügen und diese Option nutzen möchten, wenden Sie sich an den Support, um Unterstützung bei der Aktualisierung Ihrer bestehenden Lizenz und zusätzliche Installationsanweisungen zu erhalten.

Anforderungen an den eingebetteten Wickr Enterprise-Cluster

Bevor Sie mit der Installation des integrierten Wickr Enterprise-Clusters beginnen, stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind.

Netzwerkanforderungen

Sie müssen den Zugriff auf Ihren Wickr-Server über die folgenden Ports zulassen:

- 443/TCP für HTTPS
- Nur TCP-Proxy aufrufen — Der TCP-Proxy-Port, der für TCP-Anrufverkehr in KOTS konfiguriert ist
- 16384-19999/UDP für UDP-Anrufverkehr
- Nur LAN — 30000/TCP für den Zugriff auf die KOTS-Administrationskonsole

Systemanforderungen

Stellen Sie vor der Installation sicher, dass Sie entweder über eine VM (virtuelle Maschine) oder eine physische Maschine verfügen, auf der ein Linux-basiertes Betriebssystem (OS) mit den folgenden Mindestressourcen ausgeführt wird:

- 8 CPU-Kerne
- 12 Gigabyte (GB) RAM
- 100 Gigabyte (GB) Festplattenspeicher auf der Partition/(Root)

Der eingebettete Wickr Enterprise-Cluster wurde auf den folgenden Linux-Betriebssystemen getestet, aber auch andere Linux-basierte Betriebssystemoptionen könnten geeignet sein:

- Red Hat Enterprise Linux 9.5
- Amazon Linux 2023
- Rocky Linux 9.5

Installation des eingebetteten Wickr Enterprise-Clusters (Standard)

Sobald Sie die Download-Anweisungen haben, laden Sie das Wickr Enterprise-Paket auf den Zielcomputer herunter und entpacken Sie es.

```
curl -f "https://replicated.app/embedded/wickr-enterprise-ha/stable/6.52" -H
 "Authorization: [redacted]" -o wickr-enterprise-ha-stable.tgz
tar xvf wickr-enterprise-ha-stable.tgz
```

Sie sollten jetzt zwei Dateien haben, `wickr-enterprise-ha` und`license.yaml`. Bei der `wickr-enterprise-ha` Datei handelt es sich um eine Binärdatei, die alle für die Installation des Embedded Clusters erforderlichen Teile enthält, wohingegen es sich um Ihre Wickr-Lizenz `license.yaml` handelt, die zur Validierung Ihrer Installation verwendet wird.

In diesem Stadium kann eine Basisinstallation durchgeführt werden, indem die `wickr-enterprise-ha` Datei ausgeführt wird:

```
./wickr-enterprise-ha install --license license.yaml
```

Sobald der Installationsvorgang beginnt, werden Sie aufgefordert, ein Passwort für die Admin-Konsole einzugeben. Geben Sie ein sicheres Passwort ein und stellen Sie sicher, dass Sie es speichern, da Sie es benötigen, wenn Sie auf die KOTS Admin-Konsole zugreifen, um mit der Konfiguration Ihrer Installation fortzufahren.

Nach Abschluss der Installation sieht die Ausgabe wie folgt aus:

```
   sudo ./wickr-enterprise-ha install --license license.yaml
? Set the Admin Console password (minimum 6 characters): *********
? Confirm the Admin Console password: *********
# Host files materialized!
# Host preflights succeeded!
# Node installation finished!
# Storage is ready!
# Embedded Cluster Operator is ready!
# Registry is ready!
# Application images are ready!
# Admin Console is ready!
Visit the Admin Console to configure and install wickr-enterprise-ha:
 http://192.168.1.100:30000
```

Gehen Sie nach der Standardinstallation mit einem Webbrowser zur URL der KOTS-Admin-Konsole, die in der Ausgabe angegeben ist. In diesem Beispiel lautet `http://192.168.1.100:30000` die URL. Ihre URL unterscheidet sich jedoch je nach Ihrer Netzwerkkonfiguration.

Installation mit mehreren Knoten

Wickr Enterprise Embedded Cluster-Installationen mit mehreren Knoten bieten Benutzern von Embedded Clustern die Möglichkeit, die Wickr Calling- und Wickr Messaging-Workloads auf verschiedene physische Maschinen zu verteilen. Zu diesem Zweck nutzt Wickr Enterprise das Replicated Embedded Cluster Multi-Node-Tooling.

Port-Anforderungen

Die folgenden Ports müssen für alle Mitglieder des Clusters geöffnet sein, damit die Multi-Node-Funktionalität ordnungsgemäß funktioniert. Diese müssen nur zwischen den Knoten selbst geöffnet sein und dürfen nicht für das gesamte Internet geöffnet sein.

- 53 TCP/UDP
- 2380/TCP
- 4789/UDP
- 6443/TCP
- 8080/TCP
- 9091/TCP
- 9443/TCP
- 10249/TCP
- 10250/TCP
- 10256/TCP
- 30000/TCP
- 50000/TCP

Anforderungen an die Lizenz

Die Konfigurationsoptionen für Wickr Embedded Cluster Multi-Node erfordern zusätzliche Lizenzrechte. Wenden Sie sich Support an, um sicherzustellen, dass Ihre Lizenz diese Funktion unterstützt.

Bei der Ersteinrichtung wird ein zusätzlicher Knoten erstellt

Wenn Sie den Wickr Enterprise Embedded Cluster zum ersten Mal konfigurieren, können Sie während des Einrichtungsvorgangs einen zusätzlichen aufrufenden Knoten erstellen. Folgen Sie zunächst dem unter Installation des eingebetteten Wickr Enterprise Clusters (Standard) beschriebenen Verfahren. Wenn Sie zum KOTS-Administrationsbereich wechseln, werden Sie aufgefordert, zusätzliche Knoten zu erstellen.

> **Note**
>
> Derzeit unterstützt Embedded Cluster Multi-Node nur einen aufrufenden Knoten und einen Knoten. messaging/controller

Deaktivieren Sie zunächst die Option Controller-Rolle und wählen Sie die Option Anruferrolle aus. Dadurch werden zusätzliche Befehlssätze für die Konfiguration des neuen Knotens aufgefüllt. Führen Sie diese Anweisungen auf dem neuen Knoten aus, um ihn so zu konfigurieren, dass er dem Cluster als aufrufender Knoten beitritt.

Führen Sie auf dem neuen Knoten Anweisungen aus, die den folgenden Beispielen ähneln:

1. Laden Sie die Binärdatei auf den neuen Knoten herunter:

```
curl -k https://172.31.42.64:30000/api/v1/embedded-cluster/binary -o wickr-enterprise-ha.tgz
```

2. Extrahieren Sie die Binärdatei:

```
tar -xvf wickr-enterprise-ha.tgz
```

3. Verbinden Sie den Knoten mit dem Cluster:

```
sudo ./wickr-enterprise-ha join 172.31.42.64:30000 AAAAAbbbbbbbbCCCCCCCzzzzz
```

Nachdem der Join-Befehl erfolgreich abgeschlossen wurde, wird der neue Knoten auf der Seite „Cluster konfigurieren“ mit der zugewiesenen Rolle „Anrufer“ angezeigt. Wählen Sie Weiter, um zur Wickr Enterprise-Konfigurationsseite zu gelangen. Folgen Sie den Anweisungen für die Konfigurationsoptionen für eingebettete Knoten, die in der Konfiguration der KOTS-Administratorkonsole beschrieben sind.

Hinzufügen eines zusätzlichen Knotens zu einer vorhandenen eingebetteten Cluster-Installation

Um einer vorhandenen Wickr Enterprise Embedded Cluster-Installation einen aufrufenden Knoten hinzuzufügen, navigieren Sie zur KOTS Admin Console. Melden Sie sich dazu über SSH oder einen anderen Mechanismus beim Knoten an und navigieren Sie zu dem Installationsverzeichnis, das die

für die Installation verwendete `wickr-enterprise-ha` Binärdatei enthält. Führen Sie `./wickr-enterprise-ha admin-console` den Befehl aus, um die KOTS Admin Console zu starten. Wenn dieser Befehl keine Ausgabe zurückgibt, läuft die KOTS Admin Console bereits und Sie können darauf zugreifen, indem Sie in einem Webbrowser zu Port 30000 auf der IP des Knotens navigieren, zum Beispiel:. `https://127.0.0.1:30000/`

Geben Sie das KOTS-Administratorkennwort ein, wenn Sie dazu aufgefordert werden, und führen Sie dann das folgende Verfahren aus, um einen zusätzlichen Knoten zu erstellen:

1. Sobald Sie angemeldet sind, navigieren Sie zur Seite Cluster Management oben links in der KOTS Admin Console.
2. Wählen Sie Knoten hinzufügen aus.
3. Deaktivieren Sie Controller unter. `Roles`
4. Wählen Sie Anrufen unter `Roles`
5. Folgen Sie den Anweisungen, um die Befehle auf dem neuen Knoten auszuführen, den Sie hinzufügen möchten.
6. Wenn Sie fertig sind, wählen Sie Schließen
7. Ihr neuer Knoten wird in der Knotenliste mit der Rolle Anrufer angezeigt.
8. Navigieren Sie zur Anwendungsseite oben links in der KOTS Admin Console
9. Wählen Sie in der Navigationsleiste oben auf der Seite die Option Config aus.
10. Navigieren Sie im linken Navigationsbereich zum Abschnitt Anrufe.
11. Wählen Sie Anrufende Knoten erforderlich aus, um die Verwendung des Anrufenden Knotens zuzulassen.
12. Scrollen Sie zum Ende der Seite und wählen Sie Konfiguration speichern.
13. Es erscheint ein Popup, das darauf hinweist, dass die Config aktualisiert wurde. Wählen Sie Gehe zur aktualisierten Version.
14. Auf der Seite mit der aktualisierten Version wird die aktuell installierte Version angezeigt. Unter den installierten Versionen wird ein neuer Zeileneintrag mit der Bezeichnung Config Change aufgeführt. Wählen Sie Deploy, um diese neue Version bereitzustellen und den neuen aufrufenden Knoten zu aktivieren.

Konfiguration der KOTS-Administratorkonsole

Die KOTS-Administrationskonsole verwendet zunächst ein selbstsigniertes Zertifikat, das Sie als Ausnahme in Ihrem Browser zulassen müssen. Sobald Sie diese Ausnahme akzeptiert haben, werden Sie vom Konfigurationsassistenten für die KOTS-Administrationskonsole begrüßt. Dieser Assistent führt Sie durch zusätzliche Konfigurationsschritte zur Konfiguration des Verhaltens der KOTS-Admin-Konsole, einschließlich der Option, bei Bedarf ein benutzerdefiniertes Zertifikat hinzuzufügen.

Nach Abschluss der Erstkonfiguration der KOTS-Admin-Konsole werden Sie aufgefordert, Ihr Passwort für die Admin-Konsole einzugeben, das Sie während des Installationsvorgangs erstellt haben. Bei Ihrer ersten Anmeldung müssen Sie den Cluster konfigurieren.

Wählen Sie Weiter, um zur KOTS-Administrationskonsole für Wickr zu gelangen.

Wählen Sie für einen eingebetteten Cluster mit einem Knoten Weiter, um zur KOTS-Administrationskonsole für Wickr zu gelangen. Informationen zu Installationen mit mehreren Knoten finden Sie unter Installation auf mehreren Knoten.

Sobald Sie sich in der KOTS-Administrationskonsole befinden, konfigurieren Sie Ihre Installation nach Ihren Bedürfnissen. Wenn Sie das eingebettete Cluster-Angebot verwenden, sollten einige wichtige Konfigurationseinstellungen festgelegt werden, um die ordnungsgemäße Funktionalität Ihrer Wickr Enterprise-Installation sicherzustellen.

- Hostname — Dies ist der Hostname, den Sie für die Kommunikation mit der Wickr-Installation verwenden. Stellen Sie sicher, dass Sie geeignete DNS-Einträge für diese Domain erstellen, die auf Ihre Wickr Enterprise-Installation verweisen.
- Aktivieren Sie unter Erweiterte Optionen die Option [] Ingress Controller konfigurieren, um einen Konfigurationsblock für die Konfiguration von Kubernetes Ingress verfügbar zu machen. Wählen Sie im Ingress-Konfigurationsblock Single Node Embedded Cluster aus und geben Sie dann die mit Ihrem Wickr-Server verknüpfte „öffentliche“ IP in das Textfeld Loadbalancer External IP (Only) ein. IPv4

  Wenn Sie sich nicht sicher sind, was diese IP ist, können Sie den folgenden Befehl über die Befehlszeile auf dem Wickr-Server ausführen, um diesen Wert zu ermitteln: `ip route get 1.1.1.1|awk '{print $7}'`
- Aktivieren Sie unter Erweiterte Optionen die Option Modus mit geringem Ressourcenverbrauch aktivieren.

- Wenn Sie einen eingebetteten Cluster mit einem einzelnen Knoten verwenden, stellen Sie sicher, dass die Option Anrufende Knoten erforderlich deaktiviert ist. Andernfalls, wenn Sie bei der Ersteinrichtung einen Calling-Knoten hinzugefügt haben, stellen Sie sicher, dass Require Calling Nodes ausgewählt ist.
- Wenn Sie eine Komplettlösung wünschen, die keine externe Datenbank oder keinen S3-kompatiblen Speicher für die gemeinsame Nutzung von Dateien verwendet, wählen Sie die internen Optionen für die folgenden Einstellungen aus:
  - Datenbank
  - S3-Speicherort

    Der interne S3-Speicherort bietet zusätzliche Optionen zur Konfiguration der Speicherkapazität. Es wird empfohlen, klein anzufangen und bei Bedarf zu erweitern, da eine Verkleinerung nach der Bereitstellung keine Option ist.

Nachdem Sie alle erforderlichen Funktionen konfiguriert haben, scrollen Sie zum Ende der Konfigurationsseite und wählen Sie Config speichern. Dadurch werden einige Preflight-Hostprüfungen eingeleitet. Sobald die Preflight-Prüfungen abgeschlossen sind, wählen Sie Deploy, um mit der Wickr Enterprise-Installation zu beginnen.

Jetzt können Sie mit der Konfiguration Ihrer Wickr Enterprise-Installation beginnen. Weitere Informationen zur Konfiguration von Wickr Enterprise finden Sie unter Was ist Wickr Enterprise? .

Zusätzliche allgemeine Installationsanforderungen

Installationen mit IP-Hostnamen

Wenn Ihre Installation einen IP-basierten Hostnamen erfordert, gibt es einige zusätzliche Konfigurationsoptionen. Diese Anweisungen sind spezifisch für IP-basierte Hostnamen. Es wird empfohlen, die anderen oben aufgeführten Anweisungen für die Grundkonfiguration zu befolgen.

Führen Sie im KOTS-Administrationsbereich die folgenden Schritte aus.

1. Stellen Sie den Hostnamen auf die IP ein, die Sie verwenden werden.
2. Wählen Sie unter Zertifikate die Option Zertifikat hochladen aus. Generieren Sie dann ein selbstsigniertes Zertifikat, indem Sie den Anweisungen für ein IP-basiertes Zertifikat folgen. Weitere Informationen finden Sie unter Generieren eines selbstsignierten Zertifikats.
3. Laden Sie die `.crt` Datei für das Zertifikat und die `.key` Datei für den privaten Schlüssel hoch

4. Laden Sie die `.crt` Datei für die Zertifikatskette erneut hoch.
5. Aktivieren Sie das Kontrollkästchen Ein angeheftetes Zertifikat einrichten.
6. Laden Sie das `.crt` für das angeheftete Zertifikat hoch.
7. Deaktivieren Sie unter Anrufen die Kontrollkästchen Öffentliche Server-IP-Adressen automatisch ermitteln und Primäre Host-IP-Adresse für Anruf-Verkehr verwenden.
8. Geben Sie unter Anrufen die IP-Adresse des Hostnamens in das Textfeld Hostname Override ein.
9. Aktivieren Sie unter Erweiterte Optionen das Kontrollkästchen Ingress Controller konfigurieren. Ein neuer Konfigurationsabschnitt namens Ingress wird unten angezeigt.
10. Wählen Sie unter Ingress die Option Single Node Embedded Cluster aus.
11. Geben Sie unter Ingress die IP für die „öffentliche“ Schnittstelle auf dem Wickr-Server ein. Dies kann sich von der IP unterscheiden, die Sie als Hostname verwenden. Weitere Informationen zu diesem Wert finden Sie in den grundlegenden Konfigurationsschritten.
12. Aktivieren Sie unter Ingress die Option Wildcard-Hostnamen verwenden.

SELinux Modus erzwingen

Wenn Sie die Verwendung von SELinux im Erzwingungsmodus benötigen, ändern Sie das Standarddatenverzeichnis, das für die Installation des eingebetteten Clusters verwendet wird. Es wird empfohlen, es zu verwenden, `/opt` da es getestet wurde, um mit den meisten SELinux Richtlinien für diesen Anwendungsfall zu funktionieren.

```
mkdir /opt/wickr
./wickr-enterprise-ha install --license license.yaml --data-dir /opt/wickr --ignore-
host-preflights
```

Die standardmäßigen Preflight-Prüfungen für die Installation von replizierten eingebetteten Clustern versuchen zu überprüfen, ob sich die Datei im permissiven Modus SELinux befindet, und schlagen fehl, wenn SELinux die Option Enforcing aktiviert ist. Um dies zu umgehen, müssen Sie das Befehlszeilenargument verwenden. `--ignore-host-preflights` Wenn Sie die Befehlszeilenoption verwenden, wird eine Eingabeaufforderung angezeigt, die der folgenden ähnelt. Geben Sie Ja ein, wenn Sie dazu aufgefordert werden.

```
#  1 host preflight failed
```

```
•  SELinux must be disabled or run in permissive mode. To run SELinux in permissive
 mode, edit /etc/selinux/config, change the line
'SELINUX=enforcing' to 'SELINUX=permissive', save the file, and reboot. You can run
 getenforce to verify the change."

? Are you sure you want to ignore these failures and continue installing? Yes
```

AirGap Installationen

Die eingebettete Cluster-Installationsoption für Wickr Enterprise unterstützt Airgapped-Installationen. Zusätzliche Konfigurationen und Aktivierungen für Ihre Lizenz sind erforderlich. Wenden Sie sich an den Support, wenn Sie daran interessiert sind, Wickr Enterprise Embedded Cluster in einer Airgap-Umgebung zu verwenden.

Bei der Durchführung einer Airgap-Installation unterscheiden sich die Download-Anweisungen von der Standardinstallationsmethode. Sie sollten wie folgt aussehen:

```
curl -f "https://replicated.app/embedded/wickr-enterprise-ha/stable/6.52?airgap=true" -
H "Authorization: [redacted]" -o wickr-enterprise-ha-stable.tgz
```

Laden Sie das Paket auf einen Computer mit Internetzugang herunter und übertragen Sie es dann mit Ihrer bevorzugten Datenübertragungsmethode in Ihre Airgap-Umgebung. Extrahieren Sie das Paket, sobald es übertragen wurde, wie Sie es mit jedem Standardinstallationspaket tun würden. Eine dritte Dateiwickr-enterprise-ha.airgap, die alle zugehörigen Wickr Enterprise-Anwendungsdienst-Images enthält, wird mitgeliefert.

```
tar xvf wickr-enterprise-ha-stable.tgz
```

Während der Installation muss das `--airgap-bundle` Befehlszeilenargument nach dem Extrahieren festgelegt werden. Andernfalls folgt der Vorgang dem Standardinstallationsverfahren.

```
./wickr-enterprise-ha install --license license.yaml --airgap-bundle wickr-enterprise-
ha.airgap
```

Aktualisierung eines eingebetteten AirGapped-Clusters

Gehen Sie wie folgt vor, um einen AirGapped eingebetteten Cluster zu aktualisieren.

1. Laden Sie das neue eingebettete Clusterpaket von Replicated herunter und übertragen Sie es mithilfe Ihrer Standard-Datenübertragungsmethoden für Ihre Airgapped-Umgebung auf den Host-Computer. Nachdem sich das neue Paket auf dem Host-Computer befindet, extrahieren Sie den Tarball:

```
tar xvf wickr-enterprise-ha-stable.tgz
```

2. Führen Sie das Update mit dem neuen Binär- und Airgap-Paket aus:

```
./wickr-enterprise-ha update --airgap-bundle wickr-enterprise-ha.airgap
#  Application images are ready!
#  Finished!
```

3. Starten Sie die KOTS-Administrationskonsole und melden Sie sich mit Ihren Standardmethoden für den Zugriff auf die KOTS-Administrationskonsole mit der angegebenen URL an

```
./wickr-enterprise-ha admin-console
```

4. Sobald Sie sich bei der KOTS-Admin-Konsole angemeldet haben, finden Sie links unter Version das neueste verfügbare Update und klicken Sie dann auf die Schaltfläche Gehe zum Versionsverlauf.
5. Wählen Sie unter Verfügbare Updates die Option Bereitstellen für die neue Version aus. Gehen Sie durch die Bildschirme:
    1. Ändern Sie alle Konfigurationsoptionen, scrollen Sie nach unten und wählen Sie dann Weiter.
    2. Stellen Sie sicher, dass keine Preflight-Prüfungen fehlgeschlagen sind, und wählen Sie Weiter: Bestätigen und bereitstellen aus.
    3. Wählen Sie Bereitstellen.

Zusätzliche Hinweise zum eingebetteten Wickr Enterprise-Cluster

- NAMESPACE: Im Gegensatz zu den meisten Wickr Enterprise-Installationen installiert die eingebettete Cluster-Installation die Wickr-Assets im Kotsadm-Namespace in Kubernetes und nicht in Wickr. Ändern Sie alle Skripts oder Befehle, die Sie gespeichert haben und die stattdessen `-n wickr` für kubectl, helm oder ein anderes Hilfsprogramm verwendet werden. `-n kotsadm`

- Interaktion mit dem Kubernetes-Cluster: Verwenden Sie auf dem Host-Computer die `./wickr-enterprise-ha` Binärdatei, um eine Shell mit entsprechenden Variablen zu erstellen, die so eingestellt sind, dass sie mit der Kubernetes-Installation interagieren, indem Sie sie ausführen. ./ wickr-enterprise-ha shell Dadurch wird das kubectl-Hilfsprogramm im PATH der Shell bereitgestellt und die entsprechende Kube-Konfiguration für die lokale Installation festgelegt.

Fehlerbehebung bei eingebetteten Wickr-Cluster-Installationen

Bei allen Schritten zur Fehlerbehebung wird davon ausgegangen, dass Sie Shell-Zugriff auf die Instanz haben, auf der die Wickr Embedded Cluster-Installation ausgeführt wird, und dass Sie den `./ wickr-enterprise-ha shell` Befehl ausgeführt haben, um direkt mit der Kubernetes-Installation interagieren zu können.

Allgemeine Probleme

Die Schaltfläche „Knoten hinzufügen“ fehlt auf dem Bildschirm „Clusterverwaltung“

Airgapped-Installationen

Wenn Sie eine Airgap-Installation verwenden, wenden Sie sich bitte an den Wickr-Support, um Support bei der Behebung dieses Verhaltens zu erhalten.

Standardinstallationen

Wenn Ihre Lizenz die Embedded Cluster Multi-Node-Berechtigung beinhaltet, führen Sie eine Lizenzsynchronisierung durch, um die neueste Version zu erhalten. Wenn Sie sich nicht sicher sind oder diesen Anspruch nicht haben, wenden Sie sich bitte an den Wickr-Support.

Führen Sie die folgenden Schritte aus, um eine Lizenzsynchronisierung durchzuführen.

1. Navigieren Sie zum KOTS-Bedienfeld.
2. Suchen Sie auf der Dashboard-Seite den Lizenzbereich oben rechts auf der Seite.
3. In diesem Abschnitt sollten Sie in der oberen rechten Ecke einen Hyperlink „Lizenz synchronisieren“ sehen. Wählen Sie den Hyperlink aus.
4. Sobald die Lizenz synchronisiert wurde, wird die Benutzeroberfläche aktualisiert und die Meldung Letzte Synchronisierung vor einigen Sekunden angezeigt.
5. Wählen Sie auf der KOTS-Dashboardseite im Abschnitt Version die Option Erneut bereitstellen aus.

6. Sobald die erneute Bereitstellung abgeschlossen ist, kehren Sie zur Clusterverwaltung zurück, wo Sie Knoten hinzufügen können.

Probleme beim Upgrade

Das Upgrade blieb beim Upgrade des Clusters hängen

Wenn Ihr Upgrade beim Upgrade des Clusters hängen bleibt, bedeutet dies wahrscheinlich, dass einige Pods nicht ordnungsgemäß beendet wurden. Melden Sie sich bei der Instanz an und verwenden Sie den `./wickr-enteprise-ha shell` Befehl, um die Shell-Umgebung für die Verwaltung der Kubernetes-Installation aufzurufen.

1. Identifizieren Sie die Pods, die noch laufen:

   ```
   kubectl -n kotsadm get pods | grep Running
   ```

2. `kubectl -n kotsadm delete pod` *`name-of-running-pod`*

   > **Note**
   >
   > Wenn einer der laufenden Pods `embedded-cluster-upgrade-XXXXXXXXXXXXXX-xxxxx kotsadm-xxxxxxx` oder ähnliches ist, löschen Sie ihn nicht, da diese Pods für die Durchführung des Upgrades erforderlich sind.

3. Stellen Sie sicher, dass keine Pods mehr laufen.

   ```
   kubectl -n kotsadm get pods | grep Running
   ```

Dieses Verfahren sollte es ermöglichen, das Cluster-Upgrade mit dem Wickr-Upgrade fortzusetzen.

Die Anwendung wurde während des Cluster-Upgrades nicht aktualisiert und kann keine neue Version bereitstellen

Wenn die Anwendung nach dem Upgrade weiterhin die alte Version verwendet, befindet sich die neue Version möglicherweise in einem inkonsistenten Zustand.

Überprüfen Sie die Kubernetes-Installationsaufzeichnungen:

1. Öffnen Sie die Kubernetes-Shell im Installationsprogramm.

   ```
   ./wickr-enterprise-ha shell
   ```

2. Führen Sie den folgenden kubectl-Befehl aus:

   ```
   kubectl get installations
   ```

3. Die Ausgabe wird ungefähr so aussehen:

   ```
   [root@ip-172-31-6-72 ~]# kubectl get installations
   NAME             STATE      INSTALLERVERSION   CREATEDAT              AGE
   20251113170603   Obsolete   2.1.3+k8s-1.30     2025-11-13T17:06:05Z   22h
   20251113180133   Failed     2.6.0+k8s-1.31     2025-11-13T18:01:37Z   21h
   ```

4. Löschen Sie die fehlgeschlagene Installation.

   ```
   kubectl delete installation 20251113180133
   ```

5. Versuchen Sie erneut, das Upgrade über das KOTS Admin-Panel auszuführen.

RabbitMQ Pod schlägt mit Protokollzeilen fehl **`Error while waiting for Mnesia tables: {timeout_waiting_for_tables}`**

Das Geheimnis und der Speicher von RabbitMQ sind nicht synchron. Dies passiert normalerweise, wenn mehrere RabbitMQ-Instanzen ausgeführt werden und es zu einem Fehler bei der Auswahl eines Leiters oder eines Quorums kommt. Um dieses Problem zu beheben, löschen Sie den RabbitMQ-Dienst und seine Speichervolumes und stellen Sie sie dann erneut bereit.

Gehen Sie wie folgt vor, um das fehlgeschlagene RabbitMQ zu löschen.

1. Lösche das RabbitMQ Statefulset.

   ```
   kubectl -n kotsadm delete statefulset rabbitmq –cascade=orphan
   ```

2. Löschen Sie die verbleibenden RabbitMQ-Pods. Wenn mehrere RabbitMQ-X-Pods laufen, geben Sie diesen Befehl mehrmals ein und aktualisieren Sie den RabbitMQ-X-Wert so, dass er den zusätzlichen Pod-Namen entspricht.

   ```
   kubectl -n kotsadm delete pod rabbitmq-0
   ```

3. PVCsLöschen Sie den entsprechenden. Wenn mehrere Pods ausgeführt werden, geben Sie diesen Befehl mehrmals ein und aktualisieren Sie sie so data-RabbitMQ-X, dass sie den entsprechenden Pods entsprechen.

   ```
   kubectl -n kotsadm delete pvc data-rabbitmq-0
   ```

4. Überprüfen Sie, ob noch Pods übrig sind. Wenn dies erfolgreich ist, sollte dies nichts ausgeben.

   ```
   kubectl -n kotsadm get pods|grep -i rabbitmq
   ```

5. Prüfen Sie PVCs, ob noch welche übrig sind. Bei Erfolg sollte nichts ausgegeben werden.

   ```
   kubectl -n kotsadm get pvc|grep -i rabbitmq
   ```

6. Stellen Sie die Lösung erneut über das KOTS Admin-Panel bereit.

Weitere Informationen zur Fehlerbehebung finden Sie unter Problembehandlung.

Dokumentverlauf

In der folgenden Tabelle werden die Dokumentationsversionen für Wickr Enterprise Automated Install Guide beschrieben.

Änderung	Beschreibung	Datum
Sicherheitseinstellungen	Sicherheitseinstellungen wurden hinzugefügt. Weitere Informationen finden Sie unter Sicherheitseinstellungen.	26. August 2025
Installation mit mehreren Knoten	Eine Installation mit mehreren Knoten wurde hinzugefügt. Weitere Informationen finden Sie unter Installation auf mehreren Knoten.	26. August 2025
Eingangseinstellungen aufrufen	Einstellungen für eingehende Anrufe wurden hinzugefügt. Weitere Informationen finden Sie unter Einstellungen für eingehende Anrufe.	26. August 2025
Automatische Bereitstellungsoptionen	Automatische Bereitstellungsoptionen wurden hinzugefügt. Weitere Informationen finden Sie unter Wickr Enterprise installieren.	23. Februar 2024
Ports, auf die Zulassungsliste gesetzt werden sollen	Port TCP/8443 wurde der Zulassungsliste hinzugefügt. Weitere Informationen finden Sie unter Anforderungen.	12. Februar 2024
Ressourcen und Ports auf der Zulassungsliste löschen	Anweisungen zum Zerstören von Ressourcen wurden hinzugefügt. Weitere Informati	17. August 2023

	onen finden Sie unter Ressourcen zerstören. Darüber hinaus wurden Ports zur Zulassungsliste hinzugefügt. Weitere Informationen finden Sie unter Anforderungen.	
Erstversion	Erste Version des Wickr Enterprise Automated Install Guide	4. August 2023

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.