aws

Guida all'installazione automatizzata

Impresa Wickr

Impresa Wickr: Guida all'installazione automatizzata

Table of Contents

Cos'è Wickr Enterprise?

Wickr Enterprise è un servizio end-to-end criptato e ospitato autonomamente che aiuta le organizzazioni e le agenzie governative a comunicare in modo sicuro tramite messaggistica di gruppo, chiamate vocali one-to-one e video, condivisione di file e condivisione dello schermo. I clienti possono utilizzare Wickr Enterprise per superare gli obblighi di conservazione dei dati associati alle app di messaggistica di livello consumer e facilitare la collaborazione in modo sicuro. I controlli amministrativi e di sicurezza avanzati aiutano le organizzazioni a soddisfare i requisiti legali e normativi e a creare soluzioni personalizzate per le sfide legate alla sicurezza dei dati.

Le informazioni possono essere registrate in un archivio dati privato e controllato dal cliente per scopi di conservazione e controllo. I clienti hanno un controllo amministrativo completo sui dati, che include l'impostazione delle autorizzazioni, la configurazione di opzioni di messaggistica effimere e la definizione di gruppi di sicurezza. Gli amministratori possono anche automatizzare in modo sicuro i flussi di lavoro con i bot Wickr. Wickr Enterprise si integra con servizi aggiuntivi come Active Directory e Single Sign-On (SSO) con OpenID Connect (OIDC). Per iniziare a configurare Wickr Enterprise, vedi Guida introduttiva a Wickr Enterprise.

> **Note**
>
> Se non disponi già del pacchetto di distribuzione Wickr Enterprise, consulta Contattaci per domande commerciali.

Guida introduttiva a Wickr Enterprise

Argomenti

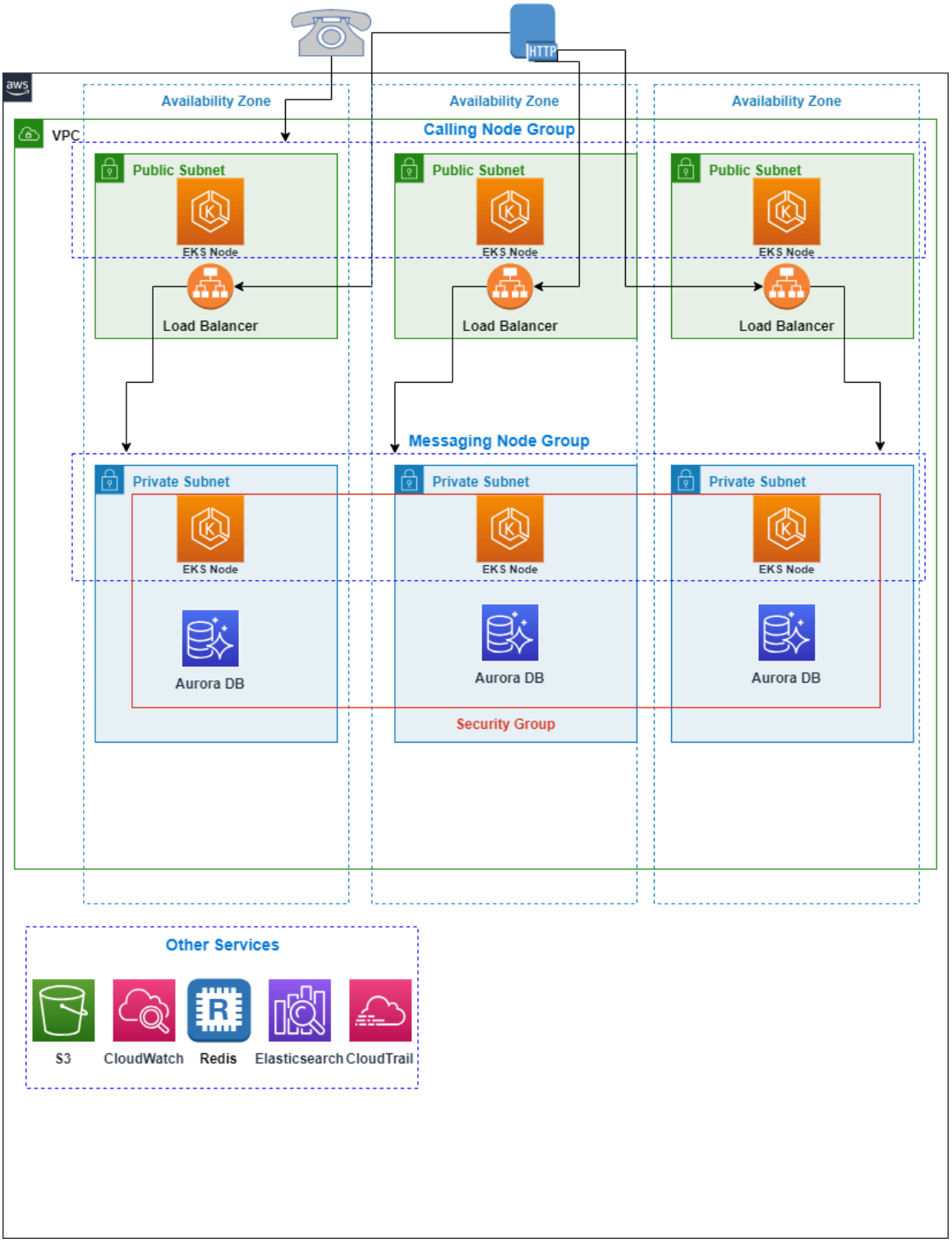
Requisiti

Prima di iniziare, verifica che siano soddisfatti i seguenti requisiti:

- Scarica Node.js 16+
- AWS CLI configurato con le credenziali per il tuo account.

  Questi verranno ricavati dal file di configurazione in `~/.aws/config` o utilizzando le `AWS_` variabili di ambiente.
- Installa kubectl. Per ulteriori informazioni, consulta Installazione o aggiornamento di kubectl nella Amazon Guide. EKSUser
- Installa kots CLI. Per ulteriori informazioni, consulta Installazione della CLI kots.
- Porte da consentire: 443/TCP per il traffico di chiamate HTTPS e TCP; 16384-19999/UDP per il traffico di chiamate UDP; TCP/8443

Architettura



Installare le dipendenze

È possibile aggiungere tutte le dipendenze al pacchetto predefinito con il seguente comando:

```
npm install
```

Configura

AWS Cloud Development Kit (AWS CDK) utilizza valori di contesto per controllare la configurazione dell'applicazione. Wickr Enterprise utilizza i valori contestuali CDK per controllare impostazioni come il nome di dominio dell'installazione di Wickr Enterprise o il numero di giorni di conservazione dei backup RDS. Per ulteriori informazioni, consulta Runtime context nella Developer Guide.AWS Cloud Development Kit (AWS CDK)

Esistono diversi modi per impostare i valori di contesto, ma consigliamo di modificarli `cdk.context.json` per adattarli al caso d'uso specifico. Solo i valori di contesto che iniziano con `wickr/` sono correlati alla distribuzione di Wickr Enterprise; il resto sono valori di contesto specifici di CDK. Per mantenere le stesse impostazioni la prossima volta che effettui un aggiornamento tramite CDK, salva questo file.

Come minimo, è necessario impostare `wickr/licensePathwickr/domainName`, e `wickr/acm:certificateArn` oppure `wickr/route53:hostedZoneId` and`wickr/route53:hostedZoneName`.

Con una zona pubblica ospitata

Se disponi di una zona ospitata pubblica Route 53 Account AWS, ti consigliamo di utilizzare le seguenti impostazioni per configurare il contesto CDK:

- `wickr/domainName`- Il nome di dominio da usare per questa implementazione di Wickr Enterprise. Se utilizzi una zona ospitata pubblica Route 53, i record DNS e i certificati ACM per questo nome di dominio verranno creati automaticamente.
- `wickr/route53:hostedZoneName`- Nome della zona ospitata da Route 53 in cui creare record DNS.
- `wickr/route53:hostedZoneId`- ID della zona ospitata Route 53 in cui creare record DNS.

Questo metodo crea un certificato ACM per conto dell'utente, insieme ai record DNS che indirizzano il nome di dominio verso il sistema di bilanciamento del carico prima della distribuzione di Wickr Enterprise.

Senza una zona ospitata pubblica

Se non disponi di una zona ospitata pubblica Route 53 nel tuo account, devi creare manualmente un certificato ACM e importarlo nel CDK utilizzando il valore di `wickr/acm:certificateArn` contesto.

- `wickr/domainName`- Il nome di dominio da utilizzare per questa distribuzione di Wickr Enterprise. Se utilizzi una zona ospitata pubblica Route 53, i record DNS e i certificati ACM per questo nome di dominio verranno creati automaticamente.
- `wickr/acm:certificateArn`- L'ARN di un certificato ACM da utilizzare sul sistema di bilanciamento del carico. Questo valore deve essere fornito se una zona ospitata pubblica Route 53 non è disponibile nel tuo account.

Importazione di un certificato in ACM

È possibile importare un certificato ottenuto esternamente con il seguente comando:

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

L'output sarà l'ARN del certificato, che dovrebbe essere usato per il valore dell'impostazione del `wickr/acm:certificateArn` contesto. È importante che il certificato caricato sia valido per le connessioni HTTPS`wickr/domainName`, altrimenti non sarà possibile convalidarlo. Per ulteriori informazioni, consulta Importazione di un certificato nella Guida per l'AWS Certificate Manager utente.

Creare record DNS

Poiché non è disponibile una zona ospitata pubblica, i record DNS devono essere creati manualmente al termine della distribuzione per indirizzare al sistema di bilanciamento del carico che precede la distribuzione di Wickr Enterprise.

Implementazione in un VPC esistente

Se è necessario utilizzare un VPC esistente, è possibile utilizzarne uno. Tuttavia, il VPC deve essere configurato per soddisfare le specifiche necessarie per EKS. Per ulteriori informazioni, consulta Visualizza i requisiti di rete di Amazon EKS per VPC e sottoreti nella Guida per l'utente di Amazon EKS e assicurati che il VPC da utilizzare soddisfi questi requisiti.

Inoltre, si consiglia vivamente di assicurarsi di disporre di endpoint VPC per i seguenti servizi:

- CLOUDWATCH
- CLOUDWATCH_LOGS
- EC2
- EC2_MESSAGGI
- ECR
- ECR_DOCKER
- ELASTIC_LOAD_BALANCING
- KMS
- GESTORE_SEGRETI
- SSM
- SSM_MESSAGES

Per distribuire risorse in un VPC esistente, imposta i seguenti valori di contesto:

- `wickr/vpc:id`- L'ID VPC in cui distribuire le risorse (ad esempio). `vpc-412beef`
- `wickr/vpc:cidr`- Il IPv4 CIDR del VPC (`172.16.0.0/16`ad es.).
- `wickr/vpc:publicSubnetIds`- Un elenco separato da virgole di sottoreti pubbliche nel VPC. L'Application Load Balancer e i nodi di lavoro EKS di chiamata verranno distribuiti in queste sottoreti (ad esempio). `subnet-6ce9941,subnet-1785141,subnet-2e7dc10`
- `wickr/vpc:privateSubnetIds`- Un elenco separato da virgole di sottoreti private nel VPC. I nodi di lavoro EKS e il server bastion verranno distribuiti in queste sottoreti (ad esempio). `subnet-f448ea8,subnet-3eb0da4,subnet-ad800b5`
- `wickr/vpc:isolatedSubnetIds`- Un elenco separato da virgole di sottoreti isolate nel VPC. Il database RDS verrà distribuito in queste sottoreti (ad esempio). `subnet-d1273a2,subnet-33504ae,subnet-0bc83ac`
- `wickr/vpc:availabilityZones`- Un elenco separato da virgole di zone di disponibilità per le sottoreti nel VPC (ad esempio). `us-east-1a,us-east-1b,us-east-1c`

Per ulteriori informazioni sugli endpoint VPC di interfaccia, consulta Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia.

Altre impostazioni

Per ulteriori informazioni, consulta Valori di contesto.

tirante

Se è la prima volta che usi CDK su questa particolare Account AWS regione, devi prima avviare l'account per iniziare a usare CDK.

```
npx cdk bootstrap
```

Implementazione

Questo processo richiederà circa 45 minuti.

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

Una volta completata, l'infrastruttura è stata creata e puoi iniziare a installare Wickr Enterprise.

Crea record DNS

Questo passaggio non è necessario se hai utilizzato una zona ospitata pubblica durante la configurazione del CDK.

L'output del processo di distribuzione includerà un valore`WickrAlb.AlbDnsName`, che è il nome DNS del load balancer. L'output sarà simile a:

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

In questo caso, il nome DNS è`Wickr-Alb-1Q5IBPJR4ZVZR-409483305.us-west-2.elb.amazonaws.com`. Questo è il valore da utilizzare quando si crea un record CNAME o A/AAAA (ALIAS) per il nome di dominio.

Se non disponi dell'output della distribuzione, esegui il comando seguente per visualizzare il nome DNS del load balancer:

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

Genera Config KOTS

> **Warning**
>
> Questo file contiene informazioni riservate sull'installazione. Non condividerlo o salvarlo pubblicamente.

Il programma di installazione di Wickr Enterprise richiede una serie di valori di configurazione relativi all'infrastruttura per poter essere installato correttamente. È possibile utilizzare uno script di supporto per generare i valori di configurazione.

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

Se hai importato un certificato esterno in ACM nel primo passaggio, passa il `--ca-file` flag a questo script, ad esempio:

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

Se ricevi un errore che indica che lo stack non esiste, imposta la variabile di `AWS_REGION` ambiente (`export AWS_REGION=us-west-2`) sulla regione selezionata e riprova. Oppure, se imposti il valore di contesto`wickr/stackSuffix`, passa il suffisso con il `--stack-suffix` flag.

Connessione al cluster Kubernetes

L'API Amazon EKS è accessibile solo tramite un host bastion creato come parte della distribuzione. Di conseguenza, tutti `kubectl` i comandi devono essere eseguiti sull'host bastion stesso o essere inoltrati tramite proxy tramite l'host bastion.

Connessioni tramite proxy attraverso il bastione

La prima volta che ti connetti al cluster, devi aggiornare il file kubeconfig locale usando il `aws eks update-kubeconfig` comando, quindi impostarlo nella tua configurazione. `proxy-url` Quindi, ogni volta che desideri connetterti al cluster, avvii una sessione SSM con l'host bastion per effettuare il port forward al proxy per l'accesso all'API.

Configurazione una tantum

C'è un valore di output nello `WickrEks` CloudFormation stack con un nome che inizia con. `WickrEnterpriseConfigCommand` Il valore contiene il comando completo necessario per generare la configurazione kubectl per il cluster. Questo output può essere visualizzato con il seguente comando:

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

Questo dovrebbe generare un comando che inizia con`aws eks update-kubeconfig`. Esegui questo comando.

Successivamente, la configurazione di Kubernetes deve essere modificata in base alle richieste proxy tramite l'host bastion. Questo può essere fatto usando i seguenti comandi:

```
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
```

Se ha funzionato correttamente, vedrete un risultato simile `'Property "clusters.arn:aws:eks:us-west-2:012345678912:cluster/WickrEnterprise5B8BF472-1234a41c4ec48b7b615c6789d93dcce.proxy-url" set.'`

Avanti verso il bastione

Per connetterti al cluster Amazon EKS, devi avviare una sessione SSM per inoltrare le richieste al proxy in esecuzione sul tuo host bastion. Il comando per eseguire questa operazione viene fornito come output nello `BastionSSMProxyEKSCommand` stack. `WickrEks` Eseguite il comando seguente per visualizzare il valore di output:

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

Il comando che emette inizierà con`aws ssm start-session`. Esegui questo comando per avviare un proxy locale in esecuzione sulla porta 8888 tramite cui puoi connetterti al cluster Amazon EKS. Se il port forward ha funzionato correttamente, l'output dovrebbe dire «In attesa di connessioni...». Mantieni attivo questo processo per tutto il tempo necessario per accedere al cluster Amazon EKS.

Se tutto è configurato correttamente, potrai eseguire `kubectl get nodes` in un altro terminale per elencare i nodi di lavoro nel cluster Amazon EKS:

```
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
```

Installazione di Wickr Enterprise

Dopo aver effettuato la connessione al cluster Kubernetes, puoi iniziare a installare Wickr Enterprise utilizzando il plug-in. `kubectl kots` Avrai bisogno del tuo file di licenza KOTS (un `.yaml` file fornito da Wickr) e del tuo file Config Values, che sono stati salvati nel file `wickr-config.json` nella sezione Genera configurazione KOTS. Per ulteriori informazioni su Generate KOTS Config, consulta Generate KOTS Config.

Installazione manuale di Wickr Enterprise

Il seguente comando avvierà l'installazione di Wickr Enterprise:

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

Ti verrà richiesto di inserire una password per la console di amministrazione KOTS. Salva questa password perché ti servirà per aggiornare o modificare la configurazione della tua installazione di Wickr Enterprise in futuro.

Una volta completata l'installazione, si `kubectl kots` aprirà una porta locale (di solitohttp:// `localhost:8080`), che fornisce l'accesso alla console di amministrazione KOTS. Puoi modificare o monitorare lo stato dell'installazione di Wickr Enterprise su questo sito o iniziare a configurare Wickr visitando il nome di dominio che hai configurato per l'installazione nel tuo browser.

Installazione di Wickr Enterprise con Lambda

Durante la distribuzione CDK, viene creata e richiamata una Lambda per completare automaticamente l'installazione di Wickr Enterprise per tuo conto. Per richiamarla manualmente, apri la AWS console e trova la funzione `WickrLambda-func*` lambda, nella scheda test, seleziona, l'input è irrilevante. `test`

Dopo l'installazione

Sono disponibili due console web per gestire l'installazione di Wickr Enterprise: la KOTS Admin Console e la Wickr Admin Console.

> **Note**
>
> Apporta le modifiche necessarie per riflettere le politiche di backup e registrazione della tua organizzazione (impostazioni Amazon S3, log di accesso Elastic Load Balancing, Amazon Virtual Private Cloud log di flusso).

Console di amministrazione KOTS

Questa interfaccia viene utilizzata per gestire la versione distribuita di Wickr Enterprise. È possibile visualizzare lo stato dell'installazione, modificare le configurazioni o eseguire aggiornamenti. La console di amministrazione KOTS è accessibile solo tramite un port forward Kubernetes, che può essere aperto utilizzando il seguente comando:

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

> **Note**
>
> Devi prima configurare la tua connessione al bastione come descritto nella sezione port forward to the bastion. Per ulteriori informazioni sul port forward to the bastion, consulta Proxying connections through the bastion.

Quando il port forward è configurato correttamente, il comando precedente produrrà quanto segue:

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

Usa l'URL fornito per accedere alla console di amministrazione KOTS. La password per accedere è quella che hai scelto durante l'esecuzione `kubectl kots install` durante l'installazione. Se devi reimpostare la password, vedi Reimpostazione della password della console di amministrazione KOTS.

Console di amministrazione Wickr

Questa interfaccia viene utilizzata per configurare l'installazione di Wickr Enterprise per configurare reti, utenti e federazioni. È accessibile tramite HTTPS con il nome DNS che hai configurato in modo che punti al tuo Load Balancer. Se il DNS è stato configurato automaticamente con una zona ospitata pubblica, il nome di dominio è il valore del valore di contesto. `wickr/domainName`

Il nome utente predefinito è`admin`, con la password`Password123`. Ti verrà richiesto di modificare questa password al primo accesso.

Valori contestuali

I valori di contesto sono coppie chiave-valore che possono essere associate a un'app, uno stack o un costrutto. Possono essere forniti all'app da un file (di solito o `cdk.context.json` nella directory del progetto) `cdk.json` o dalla riga di comando. CDK utilizza i valori di contesto per controllare la configurazione dell'applicazione. Wickr Enterprise utilizza i valori contestuali CDK per controllare impostazioni come il nome di dominio dell'installazione di Wickr Enterprise o il numero di giorni di conservazione dei backup RDS.

Esistono diversi modi per impostare i valori di contesto, ma consigliamo di modificarli in `cdk.context.json` base al caso d'uso specifico. Solo i valori di contesto che iniziano con `wickr/` sono correlati alla distribuzione di Wickr Enterprise.

Nome	Descrizione	Predefinita
`wickr/licensePath`	Il percorso verso la licenza KOTS (un `.yaml` file fornito da Wickr).	null
`wickr/domainName`	Il nome di dominio da usare per questa distribuzione di Wickr Enterprise. Se si utilizza una zona ospitata pubblica Route 53, i record DNS e i certificati ACM per questo nome di dominio verranno creati automaticamente.	null
`wickr/route53:hostedZoneId`	ID della zona ospitata da Route 53 in cui creare record DNS.	null
`wickr/route53:hostedZoneName`	Nome della zona ospitata Route 53 in cui creare record DNS.	null
`wickr/acm:certificateArn`	ARN di un certificato ACM da utilizzare su Load Balancer.	null

Nome	Descrizione	Predefinita
	Questo valore deve essere fornito se una zona ospitata pubblica Route 53 non è disponibile nell'account.	
`wickr/caPath`	Percorso del certificato, richiesto solo quando si utilizzano certificati autofirmati.	null
`wickr/vpc:id`	L'ID del VPC in cui distribui re le risorse. Richiesto solo quando si implementa in un VPC esistente. Se non è impostato, verrà creato un nuovo VPC.	null
`wickr/vpc:cidr`	IPv4 CIDR da associare al VPC creato. Se esegui la distribuzione in un VPC esistente, impostalo sul CIDR del VPC esistente.	172.16.0.0/16
`wickr/vpc:availabi lityZones`	Elenco separato da virgole delle zone di disponibilità. Richiesto solo quando si implementa in un VPC esistente.	null
`wickr/vpc:publicSu bnetIds`	Elenco separato da virgole della sottorete pubblica. IDs Richiesto solo quando si implementa in un VPC esistente.	null

Nome	Descrizione	Predefinita
`wickr/vpc:privateSubnetIds`	Elenco separato da virgole di sottorete private. IDs Richiesto solo quando si implementa in un VPC esistente.	null
`wickr/vpc:isolatedSubnetIds`	Elenco separato da virgole di IDs sottoreti isolate per il database RDS. Richiesto solo quando si implementa in un VPC esistente.	null
`wickr/rds:deletionProtection`	Abilita la protezione da eliminazione sulle istanze RDS.	true
`wickr/rds:removalPolicy`	Politica di rimozione per le istanze RDS «snapshot», «destroy» o «retention».	snapshot
`wickr/rds:readerCount`	Numero di istanze di lettura da creare nel cluster RDS.	1
`wickr/rds:instanceType`	Tipo di istanza da utilizzare per le istanze RDS.	r6g.xlarge
`wickr/rds:backupRetentionDays`	Numero di giorni per conservare i backup.	7
`wickr/eks:namespace`	Namespace predefinito per i servizi Wickr in EKS.	wickr
`wickr/eks:defaultCapacity`	Numero di nodi di lavoro EKS per l'infrastruttura di messaggistica.	3
`wickr/eks:defaultCapacityCalling`	Numero di nodi di lavoro EKS per l'infrastruttura Calling.	2

Nome	Descrizione	Predefinita
`wickr/eks:instanceTypes`	Elenco separato da virgole di tipi di istanza da utilizzare per i nodi di lavoro EKS di messaggistica.	m5.xlarge
`wickr/eks:instanceTypesCalling`	Elenco separato da virgole dei tipi di istanza da utilizzare per chiamare i nodi di lavoro EKS.	c5n.large
`wickr/eks:enableAutoscaler`	Attiva l'attivazione della funzionalità Cluster Autoscaler per EKS.	true
`wickr/s3:expireAfterDays`	Numero di giorni dopo i quali i caricamenti di file verranno rimossi dal bucket S3.	1095
`wickr/eks:clusterVersion`	Versioni del cluster, tra cui la versione Kubernetes, la versione KuberectLayer, la versione AlbController, la versione e altro. nodeGroup Release	1.27
`wickr/stackSuffix`	Un suffisso da applicare ai nomi degli CloudFormation stack.	"
`wickr/autoDeployWickr`	Implementa automaticamente l'applicazione Wickr con lambda.	true

Distruggere le risorse

Per eliminare tutto ciò che è stato creato da questa AWS CDK applicazione, è necessario eliminare lo `WickrRds` stack prima di tutti gli altri stack.

Affinché le risorse Amazon RDS vengano eliminate correttamente, la protezione da eliminazione deve essere disabilitata e la politica di rimozione deve essere impostata su `snapshot` o`destroy`. Se queste non sono le impostazioni correnti, modifica i `wickr/rds:removalPolicy` valori `wickr/rds:deletionProtection` and nel tuo AWS CDK contesto e ridistribuisci lo stack Amazon RDS eseguendolo. `npx cdk deploy -e WickrRds`

Una volta impostata correttamente la politica di protezione e rimozione da eliminazione, `cdk destroy` esegui lo stack: `WickrRds`

```
npx cdk destroy WickrRds
```

Quando lo `WickrRds` stack ha terminato la distruzione, gli CloudFormation stack rimanenti possono essere distrutti con il seguente comando:

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

Risoluzione dei problemi

Eliminazione del namespace Wickr

Se è necessario eliminare lo spazio dei `wickr` nomi per ricominciare da capo, è importante prima eseguire il backup di tutti gli account di servizio creati da CDK all'interno di tale spazio dei nomi. Questi account di servizio consentono ai servizi Wickr di comunicare tramite ruoli IAM. AWS APIs Senza di essi, attività come il caricamento di file tramite Amazon Simple Storage Service (Amazon S3) non funzioneranno più.

Utilizza il seguente comando per eseguire il backup degli account di servizio ed eliminare e ricreare il `wickr` namespace e gli account di servizio appropriati:

```
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
```

Reimpostazione della password della console di amministrazione KOTS

Puoi reimpostare la password della console di amministrazione KOTS con il seguente comando:

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

Quando modifichi questa password, potresti voler aggiornare anche il segreto di `wickr/kots` Secrets Manager, sebbene in genere non venga riutilizzato da alcuna automazione.

Problemi di connessione al cluster EKS con bastion

Se la connessione al cluster EKS tramite il bastion sembra lenta o occasionalmente scade, potresti visualizzare il seguente errore durante l'esecuzione dei comandi: `kubectl`

net/http: richiesta annullata in attesa della connessione (Client.Timeout superato in attesa delle intestazioni)

Questo problema può spesso essere risolto accedendo al bastion host tramite SSM (vedi sullo stack) e riavviando il servizio: `BastionSSMCommand` WickrEks `tinyproxy`

```
sudo systemctl restart tinyproxy
```

Installazione personalizzata

Nella sezione Installazione personalizzata, imparerai come installare Wickr Enterprise.

Argomenti

Requisiti

Prima di iniziare a installare Wickr Enterprise, verifica che siano soddisfatti i seguenti requisiti.

Requisiti hardware

Wickr Enterprise richiede un cluster Kubernetes per funzionare. È possibile operare su un singolo nodo con la modalità Low Resource abilitata, ma questa opzione non è consigliata per un uso generico in ambito di produzione. In un'implementazione di produzione, consigliamo un minimo di tre nodi di lavoro di messaggistica e un minimo di due nodi di lavoro chiamanti.

Un nodo di lavoro deve avere le seguenti specifiche minime.

- Da 2 a 4 core CPU
- 8 GB di RAM
- 200 GB di spazio su disco

Requisiti hardware minimi

Un cluster a nodo di lavoro singolo in esecuzione in modalità Low Resource richiede almeno 3000 MB di CPU e 5846 Mi Ram. Questo non include i pod del sistema kube.

Requisiti di risorse per pod

Nome del pod	Owner	CPU	Memoria
admin-api	wickr	100 metri	256 Mi
directory	Wickr	100 metri	128 Mi
scadente	wickr	100 metri	128 Mi
file proxy	Wickr	100 metri	256 Mi
oidc	Wickr	100 metri	128 Mi
opensearch	Wickr	500 metri	100 Mi
Morville	di vimini	50 metri	128 Mi
Corville-Redis	di vimini	50 metri	128 Mi
dispositivo push	wickr	100 metri	128 Mi
coniglio mq	Wickr	50 metri	256 Mi
reagire	wickr	100 metri	64 Mi
incassi	wickr	250 metri	128 Mi
redis	Wickr	50 metri	128 Mi
API del server	Wickr	250 metri	256 Mi

Nome del pod	Owner	CPU	Memoria
centralino	wickr	250 metri	512 Mi
kotsadm	NODI	50 m	50 Mi
kotsadm-minio	NODI	100 m	512 Mi
kotsadm-rqlite	LOTTI	200 m	1 Gi
minio-operatore	S3 interno	200 m	256 Mi
mini inquilino	S3 interno	100 m	256 Mi
mysql-primario	MySQL interno	100 m	512 Mi
mysql - secondario	MySQL interno	100 m	512 Mi

Requisiti di archiviazione

Wickr Enterprise richiede un valore predefinito StorageClass da utilizzare per la creazione di Persistent Volume Claims. Quando si esegue l'implementazione in un ambiente airgapped o in locale, potrebbe essere necessario configurarne uno per il cluster. Un'opzione disponibile è Longhorn. I requisiti di spazio su disco consigliati variano in base all'uso dell'opzione Internal S3 e dell'opzione Internal Mysql e alla quantità di spazio che desideri avere a disposizione per il caricamento dei file.

- Memorizzazione interna delle immagini: ~60 Gi
- RabbitMQ: 24 Gi predefinito/8 Gi in modalità Low Resource
- Redis: 24 Gi predefinito/8 Gi in modalità Low Resource
- OpenSearch: 24 Gi predefinito/8 Gi in modalità Low Resource
- Mysql interno: 80 Gi predefinito/20 Gi in modalità Low Resource
- S3 interno: 160 Gi predefinito/2Gi in modalità Low Resource
- KOTS Mini: 4 Gi
- KOTS Ralite: 1 GB

Dimensione minima di archiviazione

- 377 Gi Default con S3 interno e Mysql interno
- 111 Gi in modalità Low Resource

Requisiti della versione Kubernetes

Wickr Enterprise si affida a Replicated KOTS. Replicated, una piattaforma di distribuzione software commerciale, fornisce un elenco delle versioni di Kubernetes attualmente supportate. Per ulteriori informazioni, consulta Compatibilità delle versioni di Kubernetes.

Requisiti software

Wickr Enterprise richiede un cluster Kubernetes e KOTS per funzionare. Fai riferimento alla documentazione KOTS per le versioni del sistema operativo e di Kubernetes supportate. Per ulteriori informazioni, consulta Requisiti minimi di sistema.

Sistema host per sviluppatori

Sistema operativo: i comandi in questa documentazione sono progettati per funzionare su Linux, macOS o Windows con WSL (Windows Subsystem for Linux) installato.

Servizi interni Stateful

Wickr Enterprise può fornire servizi interni sia per il database MySQL che per lo storage compatibile con S3, tuttavia per un uso generico di produzione si consiglia di fornire questi servizi all'esterno del cluster Kubernetes.

- Database MySQL 5.7
  - Database Amazon RDS MySQL 5.7 o MySQL 5.7 (esterno)
  - Grafico Mysql Bitnami Helm (interno)
  - Archiviazione di file
    - Provider di storage compatibile con Amazon S3 o S3 (esterno)
    - Tabella di comando dell'operatore Minio (interna)

Requisiti di rete

Wickr Enterprise richiede un FQDN, certificati SSL e porte TCP e UDP aperte specifiche.

- FQDN: un dominio o sottodominio che deve essere utilizzato dalla distribuzione di Wickr Enterprise.
- Certificato SSL: una coppia di chiavi di certificato SSL firmata da una CA pubblica o una coppia di chiavi di certificato autofirmata. Il certificato deve elencare l'FQDN nel nome comune e anche come voce SAN DNS. Il certificato deve inoltre abilitare l'estensione ServerAuth. extendedKeyUsage
- Le installazioni online richiedono l'accesso in uscita a risorse replicate e di terze parti. Replicated mantiene un elenco dei propri indirizzi IP. Per ulteriori informazioni, consulta Indirizzi IP replicati. Replicated mantiene anche un elenco di risorse di terze parti necessarie. Per ulteriori informazioni, vedere Firewall Openings for Online Installations.
- Le installazioni Air-gapped richiedono l'accesso a un registro di container privato.

Nodi di messaggistica

I nodi di messaggistica non richiedono un IPV4 indirizzo pubblico e devono trovarsi in una sottorete privata. Il traffico di messaggi entrerà nel cluster tramite LoadBalancer o Ingress.

Nodi di chiamata

I nodi di chiamata richiedono un IPV4 indirizzo pubblico, quindi devono trovarsi in una sottorete pubblica. I contenuti multimediali delle chiamate vengono trasferiti tramite UDP per impostazione predefinita. Quando la chiamata TCP è abilitata, il proxy TCP accetterà connessioni su TCP 443 e le inoltrerà al servizio Orville.

- TCP: 443 Chiamata al proxy TCP
- UDP: 16384-16484 Stream Audio/Video

Accesso all'installazione e alla configurazione

L'accesso alla console di amministrazione KOTS per l'installazione e la configurazione avviene tramite un port forward Kubernetes.

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

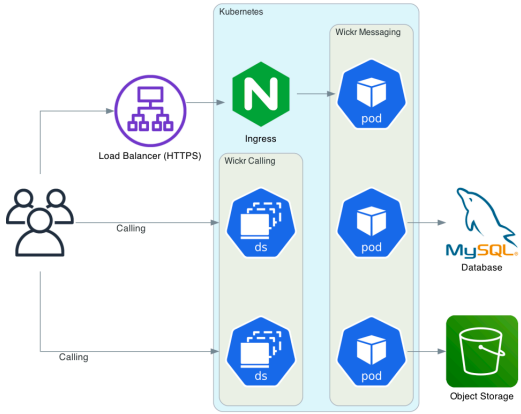
Requisiti di licenza

L'installazione richiederà un file di licenza in formato.yaml, che ti verrà fornito da Wickr Support.

Architecture

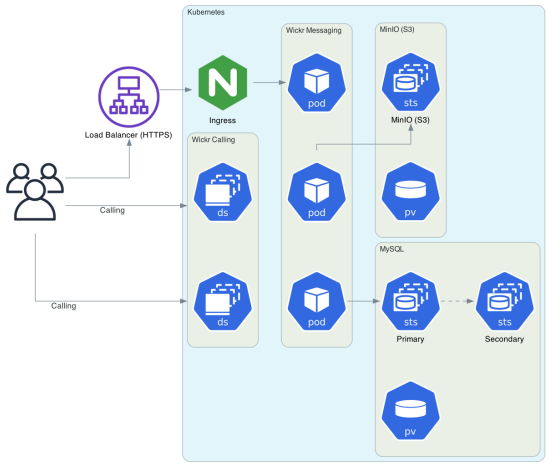
Architettura di produzione consigliata

Il diagramma seguente mostra Wickr Enterprise configurato come consigliato per la produzione, con i servizi MySQL e Object Storage situati all'esterno del cluster Kubernetes.



Architettura interna o di test

Il diagramma seguente mostra la configurazione di Wickr Enterprise, utilizzando i servizi interni MYSQL e Object Storage. Sebbene possa soddisfare le esigenze specifiche di determinate implementazioni, non è consigliato per l'uso generico in produzione.



Installazione

1. Installa kubectl e kots CLI.
2. Connect al cluster Kubernetes.
3. Ottieni il file di licenza di Wickr Enterprise da Wickr Support.
4. Installa Wickr Enterprise usando il seguente comando.

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

> **Note**
>
> license.yaml rappresenta il file di licenza fornito.

Dopo l'installazione iniziale, la console di amministrazione KOTS fornirà opzioni di gestione e configurazione a livello di cluster.

Console di amministrazione KOTS

Questa interfaccia viene utilizzata per gestire la versione distribuita di Wickr Enterprise. È possibile visualizzare lo stato dell'installazione, modificare le configurazioni o eseguire aggiornamenti di Wickr Enterprise. La console di amministrazione KOTS è accessibile solo tramite un port forward Kubernetes, che può essere aperto utilizzando il seguente comando:

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

Impostazioni di ingresso

Controller di ingresso

Wickr Enterprise supporta quattro tipi di controller di ingresso:

- LoadBalancer (Impostazione predefinita)
  - L'oggetto loadbalancer può richiedere una configurazione esplicita in installazioni completamente locali, anche se spesso viene fornito dai provider di servizi cloud.
  - Implementa il servizio di controllo di ingresso (ingress-nginx) con il tipo di servizio. LoadBalancer Ciò richiede che il cluster Kubernetes sia in esecuzione su una piattaforma che supporti bilanciatori di carico esterni.
- ALB esistente
  - Collega il controller di ingresso a un ALB esistente.
  - È necessario fornire l'ARN esistente dell'Application Load Balancer Target Group.
- NLB esistente
  - Collega il controller di ingresso a un NLB esistente.
  - Dovrai fornire l'ARN esistente del Network Load Balancer Target Group.
- NodePort
  - Il controller di ingresso (ingress-nginx) sarà configurato per utilizzare il tipo di NodePort servizio, che apre una porta su tutti i nodi del cluster Kubernetes e inoltra il traffico all'ingresso. Il traffico client può quindi essere indirizzato a questi nodi tramite DNS o un sistema di bilanciamento del carico esterno.
  - È possibile scegliere un intervallo di porte compreso tra 1 e 65535 oppure verrà utilizzata una porta casuale compresa tra 30000-32767.
- Ingresso

- Porta il tuo controller di ingresso. Questa configurazione accetterà un nome di classe di ingresso che i servizi utilizzeranno poi nei loro manifesti Ingress. Ciò implica che il controller di ingresso disponga di una connettività esterna già configurata tramite un altro meccanismo di bilanciamento del carico.
- Attualmente è supportato solo il controller ingress-nginx.

Nome host Wildcard

Per impostazione predefinita, i percorsi di ingresso saranno definiti con un valore host pari a `*`. Disabilita questa impostazione per utilizzare il nome host definito per il Wickr Enterprise Server. Wildcard Hostname è richiesto per i nomi host basati su IP.

Impostazioni del database

Wickr Enterprise richiede un database MySQL 8.0. Se utilizzi MySQL 5.7, consulta per l'aggiornamento. Aggiornamento a MySQL 8.0 Ti consigliamo di utilizzare un database esterno al cluster Kubernetes, come Amazon RDS, ma hai anche la possibilità di implementare un database MySQL interno all'interno del cluster Kubernetes come parte dell'installazione.

Impostazioni del database esterno

- Nome host: nome host o indirizzo IP del server del database.
- Nome host del lettore: nome host o indirizzo IP di un endpoint di sola lettura per il server del database (se disponibile).
- Porta: la porta su cui verrà effettuato l'accesso a MySQL.
- Nome del database: il nome del database creato sul server.
- Nome utente: l'utente che dispone delle autorizzazioni per accedere al database.
- Password: la password per quell'utente.
- Certificato CA: certificato PEM per la connessione al database tramite TLS.

> **Note**
>
> Assicurati che l'installazione di MySQL utilizzi il set di caratteri latin1 predefinito con regole di confronto latin1_swedish_ci. Ciò può essere ottenuto verificando che il server MySQL sia avviato con i seguenti flag:

```
"--character-set-server latin1", "--collation-server
latin1_swedish_ci"
```

Impostazioni interne del database

Il tipo di database interno ne distribuirà due StatefulSets nel cluster per un MySQL primario e secondario con replica binaria. Il secondario non riceve traffico ed è disponibile solo per il disaster recovery e i backup.

Dimensione di archiviazione: dimensione (in gibibyte) dei volumi persistenti per i pod del database.

Aumento delle dimensioni dello storage MySQL

> **Note**
>
> Il tipo di volume StorageClass deve supportare l'espansione del volume per aumentare le dimensioni di archiviazione. Per ulteriori informazioni, consulta Espansione del volume.

I servizi MySQL utilizzati in Wickr Enterprise vengono distribuiti come risorse in Kubernetes. StatefulSet StatefulSets rendono immutabili molte proprietà della risorsa, inclusi i modelli Persistent Volume Claim. Come soluzione alternativa per l'immutabilità di StatefulSets, è necessario eseguire le seguenti azioni per aumentare la dimensione dei volumi utilizzati da MySQL.

1. Modifica le dichiarazioni di volume persistente per e. `data-mysql-primary-0` `data-mysql-secondary-0`
    1. `kubectl -n wickr edit pvc data-mysql-primary-0.` Set `spec.resources.requests.storage`alla dimensione di archiviazione desiderata.
    2. `kubectl -n wickr edit pvc data-mysql-secondary-0.` Set `spec.resources.requests.storage`alla dimensione di archiviazione desiderata.
2. Elimina i Pod esistenti StatefulSets, ma lascia i Pod passando la `--cascade=orphan` bandiera.

    `kubectl -n wickr delete statefulset --cascade=orphan mysql-primary mysql-secondary`.
3. Nell'interfaccia utente di KOTS, aggiorna l'impostazione della dimensione di archiviazione in modo che corrisponda al valore impostato nel passaggio 1. Salva e distribuisci questa configurazione.

4. Riavviare il StatefulSets per espandere i volumi e riportare online i servizi MySQL.

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

Aggiornamento a MySQL 8.0

Database esterno (RDS)

Per mettere Wickr Backend offline, completa i seguenti passaggi.

1. Trova lo spazio dei nomi di ingresso `kubectl get deployments --all-namespaces`

   Nell'esempio seguente, lo spazio dei nomi è Wickr e le repliche sono 3.

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

2. Ridurre l'ingresso `kubectl scale deployment/ingress-nginx-controller --replicas=0 -n wickr`
3. Scatta un'istantanea per il backup del DB. Per ulteriori informazioni, consulta la sezione Gestione dei backup manuali nella Guida per l'utente di Amazon Relational Database Service.
4. Aggiorna la versione del motore a MySQL 8.0.x (MySQL 8.4 non è supportato). Per ulteriori informazioni, consulta Aggiornamento di una versione del motore di istanze DB nella Amazon Relational Database Service User Guide.

Per portare Wickr Backend online, riduci gli accessi `kubectl scale deployment/ingress-nginx-controller --replicas=3 -n wickr`

Database interno

Per ulteriori informazioni, consulta Backup e ripristino MySQL.

Archiviazione di file S3

Wickr Enterprise richiede un servizio di archiviazione compatibile con S3. Ti consigliamo di utilizzare un servizio S3 esterno al cluster Kubernetes, come Amazon S3, ma hai anche la possibilità di implementare un servizio S3 interno all'interno del cluster Kubernetes come parte dell'installazione.

Impostazioni S3 esterne

- Nome del bucket: il nome del bucket S3 in cui verranno archiviati i file caricati.
- Regione: la AWS regione del bucket S3.
- Endpoint: imposta l'endpoint che Wickr utilizzerà per interagire con l'API S3. L'impostazione predefinita è l'endpoint del servizio S3 della regione.
- Nome account del servizio Fileproxy: solo Amazon S3. Il nome di un account di servizio Kubernetes esistente da utilizzare per l'autenticazione su S3 utilizzando i ruoli IAM per gli account di servizio.
- Chiave di accesso S3 esterna: questa è la tua chiave di accesso S3 esistente.
- Chiave segreta S3 esterna: questa è la tua chiave segreta S3 esistente.

Impostazioni S3 interne

Il tipo S3 interno implementerà un numero predefinito di 4 pod server MinIO, ciascuno contenente 4 Persistent Volume Claims. La configurazione predefinita utilizza l'Erasure Coding di MiniO per aumentare la tolleranza agli errori.

- Numero di server S3 interni: il numero di pod server MinIO da creare, l'impostazione predefinita è 4 per un'implementazione con tolleranza agli errori. Questo valore può essere impostato a partire da 1 per una distribuzione. development/test
- Numero di volumi S3 interni: il numero di volumi MiniO da creare in ogni pod del server MinIO, l'impostazione predefinita è 4 per un'implementazione con tolleranza agli errori. Questo valore può essere impostato a partire da 1 per una distribuzione. development/test
- Dimensione interna del volume S3: la dimensione in GB dei volumi MinIO creati nei pod del server MinIO, l'impostazione predefinita è 10 GB.
- Una distribuzione S3 interna predefinita utilizzerà 4 server con 4. PVCs Ogni PVC è a 10 Gi, con una resa di 160 Gi di storage Raw e 120 Gi di storage con codice Erasure a disposizione degli utenti.

- È disponibile il calcolatore di codifica Minio Erasure. Per ulteriori informazioni, vedere Erasure Code Calculator.

Impostazioni persistenti per la richiesta di volume

Wickr Enterprise richiede Persistent Volume Claims per archiviare dati statici. Questa impostazione consente di specificare il nome della classe di archiviazione che si desidera utilizzare. Se lasciato vuoto, Wickr tenterà di utilizzare la classe di archiviazione predefinita. La modifica della classe di archiviazione dopo l'implementazione di Wickr non è supportata.

L'impostazione predefinita StorageClass per Persistent Volume Claims viene spesso fornita dai provider di servizi cloud, tuttavia nelle installazioni completamente locali può richiedere una configurazione esplicita utilizzando un servizio di terze parti come Longhorn.

Impostazioni del certificato TLS

Carica un certificato PEM e una chiave privata per terminare TLS. Il nome alternativo dell'oggetto sul certificato deve corrispondere al nome host configurato nelle impostazioni della distribuzione di Wickr Enterprise.

Per il campo Catena di certificati, concatena tutti i certificati intermedi (se necessario) con il certificato CA principale prima del caricamento.

Let's Encrypt

Seleziona questa opzione per generare automaticamente un certificato utilizzando Let's Encrypt. I certificati vengono emessi utilizzando la sfida HTTP-01 tramite l'operatore cert-manager.

La sfida HTTP-01 richiede che il nome DNS desiderato si risolva nel punto di ingresso del cluster (di solito un Load Balancer) e che il traffico verso la porta TCP 80 sia aperto al pubblico. Questi certificati sono di breve durata e verranno rinnovati regolarmente. È necessario mantenere aperta la porta 80 per consentire il rinnovo automatico dei certificati.

> **Note**
>
> Questa sezione si riferisce esplicitamente al certificato utilizzato dall'applicazione Wickr Enterprise stessa.

Certificato bloccato

Wickr Enterprise richiede il blocco dei certificati quando si utilizzano certificati autofirmati o certificati non considerati affidabili dai dispositivi client. Se il certificato presentato dal tuo Load Balancer è autofirmato o è firmato da una CA diversa da quella dell'installazione di Wickr Enterprise, carica il certificato CA qui per far sì che i client lo utilizzino come pin.

Nella maggior parte dei casi, questa impostazione non è richiesta.

Fornitori di certificati

Se prevedi di acquistare un certificato da utilizzare con Wickr Enterprise, consulta di seguito un elenco di fornitori i cui certificati sono noti per funzionare correttamente per impostazione predefinita. Se un fornitore è elencato di seguito, i suoi certificati sono stati convalidati esplicitamente con il software.

- Digicert
- SSL rapido

Generazione di un certificato autofirmato

Se desideri creare il tuo certificato autofirmato da utilizzare con Wickr Enterprise, il comando di esempio riportato di seguito contiene tutti i flag necessari per la generazione.

```
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"
```

Se desideri creare un certificato autofirmato basato su IP, usa invece il seguente comando. Per utilizzare il certificato basato su IP, assicuratevi che il campo Wildcard Hostname sia abilitato nelle impostazioni di Ingress. Per ulteriori informazioni, consulta Impostazioni Ingress.

```
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**
>
> Sostituisci $YOUR_DOMAIN nell'esempio con il nome di dominio o l'indirizzo IP che intendi utilizzare.

Impostazioni di chiamata

- Richiedi nodi di chiamata: quando questa impostazione è abilitata, i servizi di chiamata di Wickr vengono distribuiti solo sui nodi Kubernetes con l'etichetta. `role=calling` Disabilita questa impostazione per distribuire i servizi di chiamata e messaggistica sugli stessi nodi o per implementazioni a nodo singolo.

  In genere si desidera disabilitare anche il proxy TCP chiamante quando questa impostazione è disabilitata, poiché il servizio Proxy TCP viene eseguito sulla porta 443.
- Abilita proxy TCP: questa impostazione controlla se il servizio per la modalità fallback TCP sulle chiamate viene distribuito o meno. Disabilita questa impostazione se hai altri servizi in esecuzione su 443/tcp o se non richiedi la modalità fallback TCP per le chiamate. Questa opzione deve essere abilitata per le distribuzioni che prevedono di utilizzare Wickr Open Access.
- Rileva automaticamente gli indirizzi IP pubblici del server: Quando questa impostazione è abilitata, i servizi di chiamata scopriranno il loro indirizzo IP pubblico effettuando richieste HTTPS a e. https://ipv4.icanhazip.com/ https://ipv6.icanhazip.com/ Se disabilitata, è necessario abilitare l'impostazione «Usa l'indirizzo IP primario dell'host per il traffico di chiamata» o «Sostituzione del nome host», altrimenti i servizi di chiamata non si avvieranno.
- Usa l'indirizzo IP primario dell'host per chiamare il traffico: utilizza l'indirizzo IP primario dei nodi Kubernetes per chiamare i servizi. Ciò implica che tutti i client Wickr sono in grado di connettersi ai nodi Kubernetes sull'indirizzo IP principale del nodo, come indicato dall'API Downward. `status.hostIP`
- Sostituzione del nome host: fornisci un nome host o un indirizzo IP da restituire come punto di connettività per i servizi di chiamata. Questa impostazione deve essere utilizzata solo quando si esegue un singolo server di chiamata, poiché viene restituito lo stesso valore per tutte le repliche del servizio. Quando viene impostata l'override del nome host e l'impostazione «usa l'indirizzo IP primario dell'host» è abilitata, l'impostazione dell'indirizzo IP primario dell'host ha la precedenza.
- Rete host di chiamata abilitata: per impostazione predefinita, i call pod utilizzano la rete host dei nodi per la connettività. Disabilita questa opzione per esporre un NodePort servizio per il traffico delle chiamate. Se l'ingresso delle chiamate è abilitato, assicurati che sia configurato un servizio

appropriato per consentire il traffico in ingresso. Questo deve essere disabilitato per la conformità STIG.

Impostazioni di ingresso delle chiamate

Wickr supporta un'impostazione di ingresso delle chiamate, che consente a un client di connettersi a qualsiasi nodo di chiamata all'interno del cluster e di indirizzare la chiamata al server di chiamata corretto. Wickr supporta quattro tipi di chiamate in ingresso:

- LoadBalancer (impostazione predefinita)
  - LoadBalancer Verrà fornito dal provider di servizi cloud (le installazioni completamente in sede richiederanno una configurazione aggiuntiva). Dopo il LoadBalancer provisioning, la configurazione KOTS deve essere nuovamente aggiornata per fornire il nome host o gli indirizzi IP del sistema di bilanciamento del carico.
- NodePort
  - Espone un NodePort servizio su ogni nodo chiamante che fungerà da punto di ingresso per il traffico di chiamata. È necessario fornire un nome host che si risolva su uno o più nodi o un indirizzo IP di uno o più nodi. È possibile scegliere un intervallo di porte compreso tra 30000 e 32767 per il traffico UDP e, facoltativamente, TCP.
- NLB esistente
  - Collega il servizio di ingresso delle chiamate a un NLB esistente. Dovrai fornire l'ARN del gruppo target per UDP e, facoltativamente, il traffico TCP.
- Nessun servizio
  - Seleziona questa opzione se non hai bisogno di un servizio Kubernetes aggiuntivo per consentire il traffico in ingresso. Questo verrà in genere utilizzato con l'impostazione della rete host per indirizzare il traffico in ingresso delle chiamate direttamente ai nodi di chiamata.

Considerazioni

- Per garantire la retrocompatibilità con i client precedenti e le reti federate senza dover effettuare chiamate in ingresso, quando è abilitata la modalità di chiamata in ingresso, la modalità di chiamata precedente è ancora disponibile (connessione diretta ai server di chiamata). Se modificate le porte predefinite, assicuratevi di non avere collisioni di porte sui nodi chiamanti.
- Il traffico UDP dual-stack NLBs che serve deve avere obiettivi di backend. IPv6 Per ulteriori informazioni, consulta Network Load Balancer Target Groups.

- Se è richiesta la conformità STIG, è necessario disabilitare l'opzione di rete host per le chiamate. Se i nodi sono configurati in modalità dual-stack, ma il cluster no, si rischia di perdere la IPv6 connettività (supponendo che si tratti di un cluster). IPv4
- La chiamata in ingresso richiede nomi host o indirizzi IP predefiniti. Il ridimensionamento dei nodi o la fornitura di un routing personalizzato possono richiedere la modifica della configurazione.
- Le porte di ingresso predefinite per le chiamate sono 8443 per TCP e 16384 per UDP. Assicurati che i firewall e i gruppi di sicurezza consentano il traffico verso queste porte o porte alternative se le impostazioni predefinite vengono ignorate.

Architetture di riferimento

Ingresso con sistema di bilanciamento del carico

Questa opzione espone un singolo sistema di bilanciamento del carico come punto di ingresso per tutto il traffico di chiamata.

1. Per Calling Ingress Type, scegli Load Balancer o Existing NLB. Per ulteriori informazioni su Existing NLB, fate riferimento allo stack NLB nell'esempio di Wickr Enterprise CDK su. GitHub
2. Effettuate una delle seguenti operazioni, a seconda del tipo di ingresso della chiamata:

   - Per Existing NLB, fornite il gruppo target ARNs per il traffico UDP e TCP e il nome host del NLB.
   - Per Load Balancer, fornisci il nome host dopo che è stato fornito da Kubernetes.

   In alternativa, per entrambi i tipi di Calling Ingress, puoi fornire gli indirizzi IP del load balancer o un nome host personalizzato che punti al load balancer.
3. (Facoltativo) Per combinare il traffico di messaggistica e chiamate in un unico NLB, scegli NLB esistente nella sezione Ingress e fornisci un gruppo target HTTPS.

Ingresso con NodePort

Questa opzione è utile se la rete host è disattivata e non si desidera esporre un sistema di bilanciamento del carico aggiuntivo.

> **Note**
>
> Assicurati che i firewall e i gruppi di sicurezza consentano il traffico per. NodePorts

1. Per Calling Ingress Type, scegli. NodePort
2. Aggiungi i nomi host o gli indirizzi IP del nodo chiamante.
3. Disattiva Calling Host Network.

Ingresso diretto con HostNetwork

Questa opzione non espone alcun servizio Kubernetes aggiuntivo e consente al traffico di ingresso delle chiamate di connettersi direttamente tramite la rete host dei nodi chiamanti. Questo approccio è preferibile se è richiesta la connettività. IPv6

1. Per Calling Ingress Type, seleziona Nessun servizio.
2. Aggiungi i nomi host o gli indirizzi IP del nodo chiamante.
3. Abilita Calling Host Network.

Autoscaler del cluster Kubernetes (opzionale)

Kubernetes Cluster Autoscaler è un valore di configurazione opzionale per l'installazione di Wickr Enterprise. Aiuterà a scalare i gruppi di nodi Kubernetes in caso di aumento del traffico o di altre restrizioni delle risorse che potrebbero portare a prestazioni scadenti.

L'installazione di Wickr Enterprise supporta 3 integrazioni di provider cloud: AWS Google Cloud e Azure. Ogni provider di servizi cloud ha requisiti diversi per questa integrazione. Segui le istruzioni riportate di seguito per il tuo provider di servizi cloud specifico per abilitare questa funzionalità.

AWS

Se non hai utilizzato il WickrEnterprise CDK per installare il tuo ambiente Wickr AWS, dovrai eseguire alcuni passaggi aggiuntivi per abilitare Cluster Autoscaler.

1. Aggiungi i seguenti tag ai tuoi gruppi di nodi. Ciò consente a Cluster Autoscaler di scoprire automaticamente i nodi appropriati.

1. `k8s.io/cluster-autoscaler/`**`clusterName`** `= owneddove` ClusterName è il nome del tuo cluster Kubernetes
2. `k8s.io/cluster-autoscaler-enabled = true`

2. Aggiungi un Kubernetes Service Account, nello spazio dei nomi del sistema kube e associalo a una policy IAM che consenta la scalabilità automatica e le azioni ec2. Per ulteriori informazioni e istruzioni dettagliate, consulta Configurazione di un account di servizio Kubernetes per assumere un ruolo IAM nella Amazon EKS User Guide.

   1. Dovrai utilizzare lo spazio dei nomi «kube-system» durante la configurazione dell'account di servizio
   2. La seguente politica può essere utilizzata per l'account di servizio:

      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"
        }
    ]
}
```

Nell'interfaccia utente replicata, durante la configurazione di Cluster Autoscaler, seleziona AWScome provider cloud e fornisci il nome dell'account di servizio creato in precedenza per indicare a Cluster Autoscaler di utilizzare quell'account di servizio.

Google cloud

Si consiglia vivamente di utilizzare le funzionalità di Autoscaling integrate di GKE sia per Autopilot che per i cluster standard. Tuttavia, se si desidera procedere con questa integrazione, è necessario soddisfare i seguenti requisiti prima di procedere.

Requisiti:

1. I Managed Instance Groups (MIG) devono essere creati con Security Scope che includa almeno le risorse di «lettura/scrittura» su Compute Engine. Al momento non può essere aggiunto al MIG in un secondo momento.
2. Il cluster deve avere la Workload Identity Federation abilitata. Puoi abilitarlo su un cluster esistente eseguendo: `gcloud container clusters update ${CLUSTER_NAME} --workload-pool=${PROJECT_ID}.svc.id.goog`
3. Un account di servizio Google Cloud Platform (GCP) con accesso al ruolo `roles/compute.InstanceAdmin.v1`. Questo può essere creato utilizzando queste istruzioni:

```
# 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

Il servizio Azure Kubernetes (AKS) offre la scalabilità automatica integrata dei cluster per la maggior parte delle distribuzioni ed è altamente consigliato utilizzare questi metodi per la scalabilità automatica del cluster. Tuttavia, se le tue esigenze sono tali che tali metodi non funzionano, abbiamo fornito un'integrazione Kubernetes Cluster Autoscaler per il servizio Azure Kubernetes. Per utilizzare

questa integrazione dovrai raccogliere le seguenti informazioni e inserirle nella configurazione del pannello di amministrazione KOTS in Cluster Autoscaler dopo aver selezionato Azure come provider di servizi cloud.

Autenticazione di Azure

ID di sottoscrizione: l'ID dell'abbonamento può essere ottenuto tramite il portale di Azure seguendo la documentazione ufficiale. Per altre informazioni, vedi Ottieni sottoscrizione e tenant IDs nel portale di Azure.

I seguenti parametri possono essere ottenuti creando un AD Service Principal utilizzando l'utilità della riga di comando az.

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

ID dell'app:

Password del cliente:

ID inquilino:

Configurazione di Azure Cluster Autoscaler

Oltre ai requisiti di autenticazione, i seguenti campi sono necessari per il corretto funzionamento del cluster autoscaler. I comandi per ottenere queste informazioni sono stati forniti per comodità, tuttavia potrebbero richiedere alcune modifiche a seconda della configurazione AKS specifica.

Azure Managed Node Resource Group: questo valore è il Managed Resource Group creato da Azure al momento della creazione del cluster AKS e non il Resource Group definito. Per ottenere questo valore, sono necessari i valori CLUSTER_NAME e RESOURCE_GROUP utilizzati al momento della creazione del cluster. Una volta ottenuti questi valori, è possibile ottenerli eseguendo:

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

Nome VMSS dell'Application Node Pool: questo è il nome del Virtual Machine Scaling Set (VMSS) associato all'applicazione AKS Nodepool for the Wickr. Questa è la risorsa che verrà scalata verso l'alto o verso il basso in base alle esigenze del cluster. Per ottenere questo valore puoi eseguire il seguente comando az:

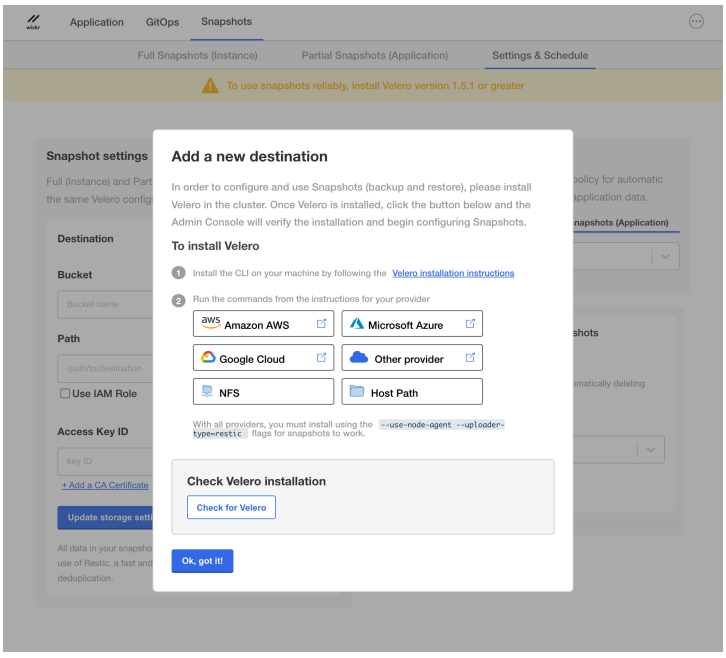
```
CLUSTER_NODEPOOL_NAME="(Your-NodePool-Name)"
CLUSTER_RESOURCE_GROUP="(Your-Managed-Node-Resource-Group-As-Defined-Above>)"
az vmss list -g ${CLUSTER_RESOURCE_GROUP} --query '[?tags."aks-managed-
poolName"==`'''${CLUSTER_NODEPOOL_NAME}'''`].{VMSS_name:name}' -o tsv
```

ACalling Nome VMSS del pool di nodi (opzionale): questo è il nome del VMSS associato al Nodepool chiamante, se ne hai uno. Per ottenere questo valore, puoi eseguire una versione modificata del comando per Application Node Pool VMSS Name sostituendo il valore CLUSTER_NODEPOOL_NAME per il nome del nodepool di nodi per il nodepool chiamante.

Backup

Wickr Enterprise utilizza Velero per scopi di Backup. Velero fornisce gli strumenti necessari per il backup e il ripristino delle risorse del cluster Kubernetes e dei volumi persistenti, indipendentemente dal fatto che operino su un provider di cloud o in locale.

Backup Velero con Minio: attualmente i backup Velero sono abilitati solo per Minio in modalità Low Resource.



Installazione utilizzando la documentazione Velero

- Installa la CLI Velero. Per ulteriori informazioni, vedere Installazione della CLI Velero.
- Installa Velero sul tuo cluster e configura lo storage in base al tuo provider:

- AWS.
- GCP.
- Azzurro.
- Altri fornitori.

Limitazione

Per impostazione predefinita, nel backup non è incluso alcun volume. Se alcuni pod montano un volume di cui è necessario eseguire il backup, è necessario configurare il backup con un'annotazione che elenchi i volumi specifici da includere nel backup.

Per ogni volume che richiede un backup, aggiungi backup.velero. io/backup-volumes annotation. The annotation name is backup.velero.io/backup-volumes e il valore è un elenco separato da virgole di volumi da includere nel backup. Per ulteriori informazioni, consulta Configurare le istantanee.

Installazione di Airgap

Wickr Enterprise e KOTS supportano entrambi l'implementazione in un cluster Kubernetes completamente airgapped. È necessario fornire l'accesso a un registro di immagini Docker privato raggiungibile dal cluster Kubernetes airgapped. Il Private Docker Image Registry fornito a KOTS deve essere protetto con l'autenticazione per funzionare correttamente a questo scopo. username/password KOTS utilizzerà il Private Docker Image Registry per ospitare tutte le immagini di Wickr Enterprise.

- Wickr Enterprise license.yaml con airgap abilitato (contatta il team di vendita o assistenza clienti di Wickr)
- Pacchetto di archiviazione wickr Enterprise wickr.airgap (contatta il team di vendita o assistenza clienti di Wickr)
- Accesso a un registro di immagini Docker privato.
- Accesso a un cluster Kubernetes distribuito nell'ambiente airgap.
- Kubectl installato.
- CLI KOTS installata.
- kotsadm.tar.gz scaricato.

Esegui i seguenti comandi per distribuire KOTS e Wickr Enterprise sul tuo cluster kubernetes airgapped. Questi comandi caricano le immagini di amministrazione di KOTS e le immagini di Wickr Enterprise nel registro delle immagini Docker Private. Al termine dei comandi, ti verrà richiesto di accedere alla console di amministrazione di KOTS per completare l'installazione di Wickr Enterprise come sopra.

```
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
```

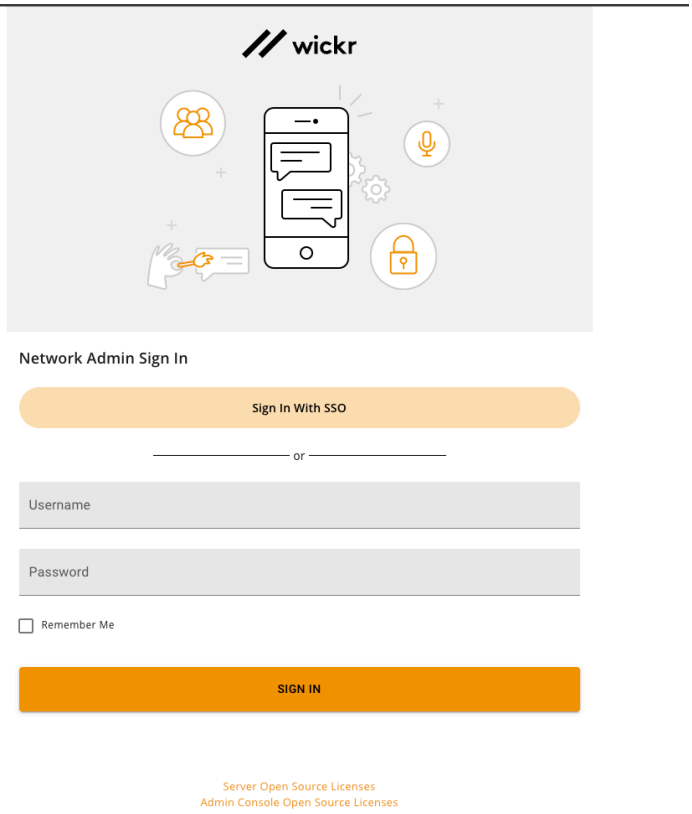
Notifica mobile per le installazioni di airgap

Sono necessari elenchi di connessioni di rete aggiuntivi per le notifiche push dal backend del server ai client mobili. Questo requisito è dovuto al modo in cui Apple iOS e Google Android implementano questa funzionalità per i dispositivi offline e in background. Consulta la documentazione relativa a questi servizi e consenti elenca gli indirizzi IP e le porte specificati.

- iOS
- Android

Console di amministrazione Wickr

L'interfaccia della console di amministrazione di Wickr viene utilizzata per amministrare l'applicazione Wickr Enterprise stessa. Può essere utilizzata per configurare reti, utenti, federazioni e altro ancora. È accessibile tramite HTTPS con il nome DNS che hai configurato in modo che punti al tuo Load Balancer. Il nome utente predefinito è admin, con la password Password123. Ti verrà richiesto di modificare questa password al primo accesso.



Impostazioni di sicurezza

AWS Wickr Enterprise fornisce impostazioni di configurazione per applicare un contesto di sicurezza avanzato per la distribuzione. Questo standard di sicurezza più elevato viene applicato a livello di pod e container ed è necessario per la conformità alla Security Technical Implementation Guide (STIG).

Imposta i seguenti parametri di configurazione per applicare il contesto di sicurezza avanzato:

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

> **Warning**
>
> Per Opensearch, questa configurazione di sicurezza disabilita `fsgroup-volume` InitContainer che aggiorna le autorizzazioni sulla memoria persistente, il che può causare problemi di compatibilità relativi alle autorizzazioni.

Domande frequenti

D: La mia distribuzione fallisce con il seguente errore in helm stderr:

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

R: Questo può accadere quando il Debug Logging è abilitato. Disattiva la registrazione di debug, elimina i job problematici e riprova.

Cluster integrato per Wickr Enterprise

L'opzione di installazione del cluster integrato per Wickr Enterprise offre un'offerta di installazione piccola ed efficiente per il prodotto Wickr Enterprise. Sfrutta il Replicated Embedded Cluster per fornire una piccola installazione Kubernetes utilizzando k0s su cui è possibile installare Wickr Enterprise. L'utilizzo di questo metodo di installazione riduce al minimo i requisiti di competenze tecniche e i requisiti hardware complessivi per un'installazione di Wickr Enterprise fornendo una soluzione «» a scapito della resilienza e dell'alta disponibilità. all-in-one

Argomenti

Guida introduttiva al cluster integrato Wickr Enterprise

Per iniziare a utilizzare l'opzione cluster integrato Wickr Enterprise, contatta l'assistenza per ricevere una licenza. Se disponi di una licenza esistente e desideri utilizzare questa opzione, contatta l'assistenza per ricevere assistenza sull'aggiornamento della licenza esistente e istruzioni di installazione aggiuntive.

Requisiti del cluster integrato Wickr Enterprise

Prima di iniziare a installare il cluster integrato Wickr Enterprise, verifica che siano soddisfatti i seguenti requisiti.

Requisiti di rete

Dovrai consentire l'accesso al tuo server Wickr sulle seguenti porte:

- 443/TCP per HTTPS
- Chiamata solo proxy TCP: la porta proxy TCP configurata per il traffico di chiamate TCP in KOTS

- 16384-19999/UDP per il traffico di chiamate UDP
- Solo LAN: 30000/TCP per l'accesso alla console di amministrazione KOTS

Requisiti di sistema

Prima dell'installazione, assicurati di disporre di una VM (macchina virtuale) o di una macchina fisica su cui sia in esecuzione un sistema operativo (OS) basato su Linux con le seguenti risorse minime disponibili:

- 8 core CPU
- 12 gigabyte (GB) di RAM
- 100 gigabyte (GB) di spazio di archiviazione su disco nella partizione/(root)

Il cluster integrato Wickr Enterprise è stato testato sui seguenti sistemi operativi Linux, ma potrebbero essere adatte anche altre opzioni di sistema operativo basate su Linux:

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

Installazione del cluster integrato Wickr Enterprise (standard)

Una volta ottenute le istruzioni per il download, scaricate il pacchetto Wickr Enterprise sul computer di destinazione e decomprimetelo.

```
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
```

Ora dovresti avere due file, e. `wickr-enterprise-ha` `license.yaml` Il `wickr-enterprise-ha` file è un file binario che include tutti i pezzi necessari per l'installazione di Embedded Cluster, mentre `license.yaml` è la licenza Wickr che verrà utilizzata per convalidare l'installazione.

In questa fase è possibile eseguire un'installazione di base eseguendo il file: `wickr-enterprise-ha`

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

Una volta avviato il processo di installazione, ti viene richiesto di inserire una password dell'Admin Console. Inserisci una password sicura e assicurati di salvarla perché ti servirà quando accedi alla console di amministrazione KOTS per continuare a configurare l'installazione.

Una volta completata l'installazione, l'output è simile al seguente:

```
   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
```

Dopo l'installazione standard, procedi all'URL della console di amministrazione KOTS fornito nell'output utilizzando un browser web. Per questo esempio, l'URL è`http://192.168.1.100:30000`. Tuttavia, l'URL sarà diverso in base alla configurazione di rete.

Installazione multi-nodo

Le installazioni multi-nodo di Wickr Enterprise Embedded Cluster offrono agli utenti di Embedded Cluster la possibilità di separare i carichi di lavoro di Wickr Calling e Wickr Messaging su macchine fisiche diverse. Per fare ciò, Wickr Enterprise sfrutta gli strumenti Replicated Embedded Cluster Multi-Node.

Requisiti porta

Le seguenti porte devono essere aperte su tutti i membri del cluster affinché la funzionalità Multi-Node funzioni correttamente. Queste devono essere aperte solo tra i nodi stessi e non verso una rete Internet più ampia.

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

Requisiti di licenza

Le opzioni di configurazione multi-nodo di Wickr Embedded Cluster richiedono privilegi di licenza aggiuntivi. Contattateci Supporto per assicurarvi che la vostra licenza supporti questa funzionalità.

Creazione di un nodo aggiuntivo durante la configurazione iniziale

Quando configuri inizialmente il Wickr Enterprise Embedded Cluster, puoi creare un nodo di chiamata aggiuntivo durante il processo di configurazione. Inizia seguendo la procedura descritta in Installazione del cluster integrato Wickr Enterprise (standard) Quando accedi al pannello di amministrazione di KOTS, ti verrà richiesto di creare nodi aggiuntivi.

> **Note**
>
> Attualmente, Embedded Cluster Multi-Node supporta solo 1 nodo di chiamata e 1 nodo. messaging/controller

Per iniziare, deselezionate l'opzione Ruolo controller e selezionate l'opzione Calling role. Questo compila set di istruzioni aggiuntivi per la configurazione del nuovo nodo. Esegui queste istruzioni sul nuovo nodo per configurarlo per unirsi al cluster come nodo chiamante.

Esegui istruzioni simili ai seguenti esempi sul nuovo nodo:

1. Scarica il file binario sul nuovo nodo:

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

2. Estrai il file binario:

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

3. Unisci il nodo al cluster:

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

Una volta completato correttamente il comando join, il nuovo nodo viene visualizzato nella pagina Configura cluster con il ruolo Calling assegnato. Scegli Continua per passare alla pagina di configurazione di Wickr Enterprise. Segui le istruzioni per le opzioni di configurazione dei nodi incorporati descritte nella configurazione della console di amministrazione KOTS.

Aggiungere un nodo aggiuntivo a un'installazione di cluster incorporato esistente

Per aggiungere un nodo di chiamata a un'installazione esistente di Wickr Enterprise Embedded Cluster, vai alla Console di amministrazione KOTS. Per fare ciò, accedi al nodo tramite ssh o altro meccanismo e vai alla directory di installazione che contiene il file `wickr-enterprise-ha` binario usato per l'installazione. Esegui `./wickr-enterprise-ha admin-console` per avviare la console di amministrazione KOTS. Se questo comando non restituisce alcun output, la console di amministrazione KOTS è già in esecuzione ed è possibile accedervi accedendo alla porta 30000 sull'IP del nodo in un browser web, ad esempio:. `https://127.0.0.1:30000/`

Inserisci la password di amministratore KOTS quando richiesta, quindi esegui la seguente procedura per creare un nodo aggiuntivo:

1. Una volta effettuato l'accesso, vai alla pagina di gestione del cluster in alto a sinistra della console di amministrazione KOTS.
2. Scegliere Add node (Aggiungi nodo).
3. Deseleziona Controller sotto. `Roles`
4. Seleziona Chiamata sotto `Roles`
5. Segui le istruzioni fornite per eseguire i comandi sul nuovo nodo che desideri aggiungere.
6. Al termine, scegli Chiudi
7. Il nuovo nodo viene visualizzato nell'elenco dei nodi con il ruolo Chiamante.
8. Vai alla pagina dell'applicazione in alto a sinistra della console di amministrazione KOTS
9. Scegli Config dalla barra di navigazione nella parte superiore della pagina.
10. Vai alla sezione Chiamate nel pannello di navigazione a sinistra.
11. Seleziona Require Calling Nodes per consentire l'uso del nodo Calling.
12. Scorri fino alla fine della pagina e scegli Salva configurazione.
13. Viene visualizzato un popup che indica che il Config è stato aggiornato. Scegli Vai alla versione aggiornata.
14. Nella pagina della versione aggiornata, viene visualizzata la versione attualmente installata. Una nuova voce è elencata nelle versioni installate con la denominazione Config Change. Scegli Deploy per distribuire questa nuova versione e abilitare il nuovo nodo di chiamata.

Configurazione della console di amministrazione KOTS

La console di amministrazione KOTS utilizza inizialmente un certificato autofirmato, che dovrai consentire come eccezione nel tuo browser. Una volta accettata questa eccezione, sarai accolto dalla procedura guidata di configurazione per la console di amministrazione KOTS. Questa procedura guidata ti guida attraverso passaggi di configurazione aggiuntivi per configurare il comportamento della console di amministrazione KOTS, inclusa l'opzione per aggiungere un certificato personalizzato, se necessario.

Una volta completata la configurazione iniziale della console di amministrazione KOTS, ti viene richiesto di inserire la password della console di amministrazione che hai creato durante il processo di installazione. Al primo accesso è necessario configurare il cluster.

Scegli Continua per passare alla console di amministrazione KOTS per Wickr.

Per un cluster incorporato a nodo singolo, scegli Continua per passare alla console di amministrazione KOTS per Wickr. Per le installazioni multi-nodo, vedi Installazione multi-nodo.

Una volta nella console di amministrazione KOTS, configura l'installazione in base alle tue esigenze. Quando si utilizza l'offerta di cluster incorporati, è necessario impostare alcune impostazioni di configurazione chiave per garantire la corretta funzionalità dell'installazione di Wickr Enterprise.

- Nome host: questo è il nome host che usi per comunicare con l'installazione di Wickr. Assicurati di creare i record DNS appropriati per questo dominio in modo che rimandino alla tua installazione di Wickr Enterprise.
- In Opzioni avanzate, seleziona l'opzione [] Configure Ingress Controller per esporre un blocco di configurazione per la configurazione di Kubernetes Ingress. Nel blocco di configurazione Ingress, seleziona Single Node Embedded Cluster, quindi inserisci l'IP «pubblico» associato al tuo server Wickr nella casella di testo denominata Loadbalancer External IP (solo). IPv4

  Se non sei sicuro di quale sia questo IP, puoi eseguire il seguente comando dalla riga di comando sul server Wickr per determinare questo valore: `ip route get 1.1.1.1|awk '{print $7}'`
- In Opzioni avanzate, seleziona l'opzione Abilita la modalità a basso consumo di risorse.
- In Chiamata, se utilizzi un cluster incorporato a nodo singolo, assicurati che l'opzione Richiedi nodi di chiamata sia deselezionata. Altrimenti, se hai aggiunto un nodo di chiamata durante la configurazione iniziale, assicurati che sia selezionato Require Calling Nodes.
- Se desideri una soluzione tutto in uno che non utilizzi un database esterno o uno storage compatibile con S3 per la condivisione di file, seleziona le opzioni interne per le seguenti impostazioni:
  - Database
  - Posizione di archiviazione S3

    La posizione di archiviazione interna di S3 offre opzioni aggiuntive per la configurazione della capacità di archiviazione. Si consiglia di iniziare con dimensioni ridotte ed espanderle se necessario, poiché la scalabilità verso il basso non è un'opzione dopo il provisioning.

Dopo aver configurato tutte le funzionalità necessarie, scorri fino alla fine della pagina di configurazione e scegli Salva configurazione. Ciò avvierà alcuni controlli preliminari all'host. Una volta completati i controlli preliminari, scegli Deploy per iniziare l'installazione di Wickr Enterprise.

Ora sei pronto per iniziare a configurare l'installazione di Wickr Enterprise. Per ulteriori informazioni sulla configurazione di Wickr Enterprise, vedi Cos'è Wickr Enterprise? .

Requisiti di installazione comuni aggiuntivi

Installazioni di nomi host IP

Se l'installazione richiede un nome host basato su IP, sono disponibili alcune opzioni di configurazione aggiuntive. Queste istruzioni sono specifiche per i nomi host basati su IP e si consiglia di seguire le altre istruzioni per la configurazione di base elencate sopra.

Nel pannello di amministrazione di KOTS, completa i seguenti passaggi.

1. Imposta il nome host sull'IP che utilizzerai.
2. In Certificati, seleziona Carica un certificato. Quindi, genera un certificato autofirmato seguendo le istruzioni per un certificato basato su IP. Per ulteriori informazioni, consulta Generazione di un certificato autofirmato.
3. Carica il `.crt` file per il certificato e il `.key` file per la chiave privata
4. Per la catena di certificati, carica nuovamente il `.crt` file.
5. Seleziona la casella di controllo Imposta un certificato bloccato.
6. Carica il file `.crt` per il certificato bloccato.
7. In Chiamata, deseleziona le caselle di controllo Scopri automaticamente gli indirizzi IP pubblici del server e Usa l'indirizzo IP primario dell'host per il traffico di chiamata.
8. In Chiamata, inserisci l'indirizzo IP del nome host nella casella di testo Hostname Override.
9. In Opzioni avanzate, seleziona la casella di controllo Configure Ingress Controller. Di seguito viene visualizzata una nuova sezione di configurazione chiamata Ingress.
10. In Ingress, seleziona Single Node Embedded Cluster.
11. In Ingress, inserisci l'IP per l'interfaccia «pubblica» sul server Wickr. Potrebbe essere diverso dall'IP utilizzato come nome host. Visualizza ulteriori informazioni su questo valore nei passaggi di configurazione di base.
12. In Ingress, seleziona Usa hostname wildcard.

SELinux Modalità di applicazione

Se è necessario utilizzare la modalità SELinux di applicazione, modificare la directory di dati predefinita utilizzata per installare il cluster incorporato. Si consiglia di utilizzarla `/opt` poiché è stata testata per funzionare con la maggior parte delle SELinux politiche per questo caso d'uso.

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

I controlli preliminari di installazione predefiniti dei cluster incorporati replicati tenteranno di convalidare la modalità permissiva e falliranno se SELinux è in Enforce. SELinux Per aggirare questo problema, è necessario utilizzare l'argomento della riga di comando. `--ignore-host-preflights` Quando si utilizza l'opzione della riga di comando, viene visualizzato un prompt simile a quello riportato di seguito. Immettere Sì quando richiesto.

```
#  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 installazioni

L'opzione di installazione cluster integrata per Wickr Enterprise supporta installazioni airgapped. Sono necessarie configurazioni e abilitazioni aggiuntive per la licenza. Contatta l'assistenza se sei interessato a utilizzare il cluster integrato Wickr Enterprise in un ambiente airgapped.

Quando si esegue un'installazione airgap, le istruzioni per il download differiscono dal metodo di installazione standard. Dovrebbero essere simili alle seguenti:

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

Scaricate il pacchetto su una macchina con accesso a Internet, quindi trasferitelo nel vostro ambiente airgapped utilizzando il metodo di trasporto dati che preferite. Una volta trasferito il pacchetto, estrailo come faresti con qualsiasi pacchetto di installazione standard. Verrà incluso un terzo file`wickr-enterprise-ha.airgap`, contenente tutte le immagini dei servizi applicativi Wickr Enterprise associati.

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

Durante l'installazione, è necessario impostare l'argomento della riga di `--airgap-bundle` comando dopo l'estrazione; in caso contrario, il processo segue la procedura di installazione standard.

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

Aggiornamento di un cluster integrato AirGapped

Per aggiornare un cluster AirGapped incorporato, completare i seguenti passaggi.

1. Scaricate il nuovo pacchetto cluster integrato da Replicated e trasferitelo sulla macchina host utilizzando i metodi di trasferimento dati standard per l'ambiente airgapped. Dopo aver installato il nuovo pacchetto sulla macchina host, estraete il tarball:

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

2. Esegui l'aggiornamento utilizzando il nuovo pacchetto binario e airgap:

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

3. Avvia la console di amministrazione KOTS e accedi all'URL fornito utilizzando i metodi standard di accesso alla console di amministrazione KOTS

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

4. Una volta effettuato l'accesso alla console di amministrazione KOTS, trova l'ultimo aggiornamento disponibile a sinistra sotto Versione, quindi premi il pulsante Vai alla cronologia delle versioni.
5. Scegli Deploy per la nuova versione in Aggiornamenti disponibili. Passeggia tra le schermate:

   1. Modifica le opzioni di configurazione, scorri verso il basso e scegli Avanti.

2. Verifica che nessun controllo preliminare non sia andato a buon fine, scegli Avanti: conferma e distribuisci.
3. Seleziona Implementa.

Note aggiuntive sul cluster integrato Wickr Enterprise

- NAMESPACE : A differenza della maggior parte delle installazioni di Wickr Enterprise, l'installazione del cluster integrato installa le risorse Wickr nello spazio dei nomi kotsadm in kubernetes e non in wickr. Modifica gli script o i comandi che hai salvato e che utilizzi invece per kubectl, helm o qualsiasi altra utilità. `-n wickr -n kotsadm`
- Interazione con il cluster Kubernetes: dalla macchina host, usa il `./wickr-enterprise-ha` file binario per creare una shell con le variabili appropriate impostate per interagire con l'installazione di Kubernetes mediante l'esecuzione. ./wickr-enterprise-ha shell Ciò fornirà l'utilità kubectl all'interno del PATH della shell e imposterà la configurazione kube appropriata per l'installazione locale.

Risoluzione dei problemi relativi alle installazioni dei cluster integrati in Wickr

Tutte le istanze di queste procedure di risoluzione dei problemi presuppongono che tu abbia accesso tramite shell all'istanza che esegue l'installazione di Wickr Embedded Cluster e che tu abbia eseguito il `./wickr-enterprise-ha shell` comando per poter interagire direttamente con l'installazione di Kubernetes.

Problemi generali

Il pulsante Aggiungi nodo non è presente nella schermata di gestione del cluster

Installazioni Airgapped

Se stai utilizzando un'installazione airgap, contatta il supporto di Wickr per ricevere assistenza nella correzione di questo comportamento.

Installazioni standard

Se la licenza include l'autorizzazione Embedded Cluster Multi-Node, esegui una sincronizzazione della licenza per ottenere la versione più recente. Se non sei sicuro o non hai questo diritto, contatta Wickr Support.

Per eseguire una sincronizzazione della licenza, completa i seguenti passaggi.

1. Vai al pannello di controllo KOTS.
2. Nella pagina Dashboard, individua la sezione della licenza nell'area in alto a destra della pagina.
3. All'interno di questa sezione, nell'angolo in alto a destra, dovresti vedere un collegamento ipertestuale Sync License. Seleziona il collegamento ipertestuale.
4. Una volta sincronizzata la licenza, viene visualizzata l'interfaccia utente aggiornata e viene visualizzata l'ultima sincronizzazione di pochi secondi fa.
5. Scegli Redeploy dalla sezione Versione della pagina del pannello di controllo di KOTS.
6. Al termine della ridistribuzione, torna alla gestione del cluster e puoi aggiungere nodi.

Problemi di aggiornamento

Aggiornamento bloccato durante l'aggiornamento del cluster

Se l'aggiornamento si blocca su Upgrading Cluster, probabilmente significa che alcuni pod non vengono terminati correttamente. Accedi all'istanza e usa il `./wickr-enteprise-ha shell` comando per accedere all'ambiente shell per la gestione dell'installazione di Kubernetes.

1. Identifica i pod ancora in esecuzione:

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

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

   > **Note**
   >
   > Se uno dei running pod è `embedded-cluster-upgrade-XXXXXXXXXXXXXX-xxxxx` `kotsadm-xxxxxxx` o è simile, non eliminatelo perché questi pod sono necessari per eseguire l'aggiornamento.

3. Verifica che non ci siano running pod rimanenti.

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

Questa procedura dovrebbe consentire all'aggiornamento del cluster di procedere con l'aggiornamento di Wickr.

L'applicazione non è stata aggiornata durante l'aggiornamento del cluster e non può distribuire una nuova versione

Se l'applicazione rimane sulla vecchia versione dopo l'aggiornamento, la nuova versione potrebbe trovarsi in uno stato incoerente.

Controlla i record di installazione di Kubernetes:

1. Apri la shell Kubernetes dal programma di installazione.

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

2. Esegui il seguente comando kubectl:

   ```
   kubectl get installations
   ```

3. L'output sarà simile a questo:

   ```
   [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. Eliminare l'installazione non riuscita.

   ```
   kubectl delete installation 20251113180133
   ```

5. Tenta di eseguire nuovamente l'aggiornamento tramite il pannello di amministrazione di KOTS.

Errore del pod RabbitMQ con righe di registro **`Error while waiting for Mnesia tables: {timeout_waiting_for_tables}`**

Il segreto e lo spazio di archiviazione di RabbitMQ non sono sincronizzati. Questo di solito accade quando vengono eseguite più istanze di RabbitMQ e causano un errore di selezione del leader o del quorum. Per risolvere questo problema, elimina il servizio RabbitMQ e i relativi volumi di archiviazione, quindi ridistribuiscilo.

Per eliminare il RabbitMQ non funzionante, completa i seguenti passaggi.

1. Eliminare il set di stato di RabbitMQ.

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

2. Elimina i pod RabbitMQ rimanenti. Se ci sono più pod RabbitMQ-X in esecuzione, esegui questo comando più volte aggiornando il valore di RabbitMQ-x in modo che corrisponda ai nomi dei pod aggiuntivi.

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

3. Eliminare il corrispondente. PVCs Se ci sono più pod in esecuzione, esegui questo comando più volte aggiornando il file in data-RabbitMQ-X modo che corrisponda ai pod appropriati.

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

4. Controlla se ci sono dei pod rimanenti, in caso di successo non dovrebbe generare alcun risultato.

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

5. Controlla se ce ne sono ancora PVCs, in caso di successo non dovrebbe produrre nulla.

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

6. Ridistribuisci tramite il pannello di amministrazione di KOTS.

Per ulteriori informazioni sulla risoluzione dei problemi, vedi Risoluzione dei problemi.

Cronologia dei documenti

La tabella seguente descrive le versioni della documentazione per Wickr Enterprise Automated Install Guide.

Modifica	Descrizione	Data
Impostazioni di sicurezza	Sono state aggiunte impostazioni di sicurezza. Per ulteriori informazioni, consulta Impostazioni di sicurezza.	26 agosto 2025
Installazione multi-nodo	È stata aggiunta l'installazione multinodo. Per ulteriori informazioni, consulta Installazione multinodo.	26 agosto 2025
Impostazioni di ingresso delle chiamate	Sono state aggiunte le impostazioni relative all'ingresso delle chiamate. Per ulteriori informazioni, consulta le impostazioni di ingresso delle chiamate.	26 agosto 2025
Opzioni di distribuzione automatica	Sono state aggiunte opzioni di distribuzione automatica. Per ulteriori informazioni, consulta Installazione di Wickr Enterprise.	23 febbraio 2024
Porte da elencare	La porta TCP/8443 è stata aggiunta alla lista delle autorizzazioni. Per ulteriori informazioni, consulta Requisiti.	12 febbraio 2024

Distruzione di risorse e porte da inserire nella lista consentita	Sono state aggiunte istruzioni su come distruggere le risorse. Per ulteriori informazioni, consulta Distruggere le risorse. Inoltre, sono state aggiunte le porte alla lista consentita. Per ulteriori informazioni, consulta Requisiti.	17 agosto 2023
Versione iniziale	Versione iniziale della Guida all'installazione automatica di Wickr Enterprise	4 agosto 2023

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.