aws

Guide d'installation automatique

Wickr Enterprise

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Wickr Enterprise: Guide d'installation automatique

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Qu'est-ce que Wickr Enterprise ? 1
Premiers pas 2
Prérequis 2
Installation des dépendances 2
Configuration 3
Sangle Bootstrap 6
Déploiement 6
Générer une configuration KOTS 7
Connexion à Kubernetes 8
Connexion par proxy via le bastion 8
Installation de Wickr Enterprise 10
Installation manuelle de Wickr Enterprise 10
Installation de Wickr Enterprise avec Lambda 10
Après l'installation 11
Console d'administration KOTS 11
Console d'administration Wickr 12
Valeurs contextuelles 13
Détruire des ressources 18
Résolution des problèmes 19
Supprimer l'espace de noms Wickr 19
Réinitialisation du mot de passe de la console d'administration KOTS 19
Problèmes de connexion au cluster EKS avec Bastion 19
Installation personnalisée 21
Prérequis 21
Configuration matérielle requise 21
Exigences en matière de logiciels 24
Exigences réseau 24
Architecture 26
Installation 26
Console d'administration KOTS 26
Paramètres d'entrée 27
Paramètres de base de données 28
Paramètres de base de données externes 28
Paramètres internes de la base de données 28

Stockage de fichiers S3 30
Paramètres de demande de volume persistants 31
Paramètres du certificat TLS 31
Let's Encrypt 31
Certificat épinglé 32
Fournisseurs de certificats 32
Génération d'un certificat auto-signé 32
Paramètres d'appel 33
Autoscaler du cluster Kubernetes (facultatif) 34
AWS 34
Cloud de Google 35
Azure 36
Sauvegardes 37
Installation à l'aide de la documentation Velero 38
Installation d'un entrefer 38
Notification mobile pour les installations d'airgap 39
Console d'administration Wickr 39
FAQ 39
Installation d'un cluster intégré 41
Premiers pas 41
Prérequis 41
Installation standard 42
Configuration de console d'administration KOTS 26
Exigences d'installation supplémentaires 45
Historique de la documentation 49
..... I

Qu'est-ce que Wickr Enterprise ?

Wickr Enterprise est un service end-to-end crypté et auto-hébergé qui aide les organisations et les agences gouvernementales à communiquer en toute sécurité par le biais one-to-one de la messagerie de groupe, des appels vocaux et vidéo, du partage de fichiers et du partage d'écran. Les clients peuvent utiliser Wickr Enterprise pour contourner les obligations de conservation des données associées aux applications de messagerie grand public et faciliter la collaboration en toute sécurité. Les contrôles de sécurité et administratifs avancés aident les entreprises à répondre aux exigences légales et réglementaires et à créer des solutions personnalisées pour relever les défis liés à la sécurité des données.

Les informations peuvent être enregistrées dans un magasin de données privé contrôlé par le client à des fins de conservation et d'audit. Les clients disposent d'un contrôle administratif complet sur les données, notamment en définissant des autorisations, en configurant des options de messagerie éphémère et en définissant des groupes de sécurité. Les administrateurs peuvent également automatiser les flux de travail en toute sécurité avec les robots Wickr. Wickr Enterprise s'intègre à des services supplémentaires tels qu'Active Directory et l'authentification unique (SSO) avec OpenID Connect (OIDC). Pour commencer à configurer Wickr Enterprise, consultez Getting started with Wickr Enterprise.

> **Note**
>
> Si vous ne possédez pas encore le package de déploiement Wickr Enterprise, consultez Contactez-nous pour les demandes commerciales.

Commencer à utiliser Wickr Enterprise

Rubriques

- Prérequis
- Installation des dépendances
- Configuration
- Sangle Bootstrap
- Déploiement
- Générer une configuration KOTS

Prérequis

Avant de commencer, vérifiez que les conditions suivantes sont remplies :

- Télécharger Node.js 16+
- AWS CLI configuré avec les informations d'identification de votre compte.

  Elles proviendront soit de votre fichier de configuration, `~/.aws/config` soit à l'aide des variables d'`AWS_`environnement.
- Installez kubectl. Pour plus d'informations, consultez Installation ou mise à jour de kubectl dans le guide Amazon EKSUser .
- Installez la CLI kots. Pour plus d'informations, consultez la section Installation de la CLI kots.
- Ports à autoriser : 443/TCP pour le trafic d'appel HTTPS et TCP ; 16384-19999/UDP pour le trafic d'appel UDP ; TCP/8443

Architecture

Installation des dépendances

Vous pouvez ajouter toutes les dépendances au package par défaut à l'aide de la commande suivante :

```
npm install
```

Configuration

AWS Cloud Development Kit (AWS CDK) utilise des valeurs de contexte pour contrôler la configuration de l'application. Wickr Enterprise utilise les valeurs contextuelles du CDK pour contrôler les paramètres tels que le nom de domaine de votre installation Wickr Enterprise ou le nombre de jours pendant lesquels les sauvegardes RDS sont conservées. Pour plus d'informations, consultez la section Contexte d'exécution dans le Guide du AWS Cloud Development Kit (AWS CDK) développeur.

Il existe plusieurs manières de définir les valeurs de contexte, mais nous vous recommandons de les modifier pour les adapter `cdk.context.json` à votre cas d'utilisation particulier. Seules les valeurs de contexte commençant par `wickr/` sont liées au déploiement de Wickr Enterprise ; les autres sont des valeurs de contexte spécifiques au CDK. Pour conserver les mêmes paramètres lors de votre prochaine mise à jour via le CDK, enregistrez ce fichier.

Au minimum, vous devez définir `wickr/licensePathwickr/domainName`, et soit `wickr/acm:certificateArn` soit `wickr/route53:hostedZoneId` et`wickr/route53:hostedZoneName`.

Avec une zone hébergée publique

Si vous avez une zone hébergée publique Route 53 dans votre Compte AWS, nous vous recommandons d'utiliser les paramètres suivants pour configurer le contexte de votre CDK :

- `wickr/domainName`- Le nom de domaine à utiliser pour ce déploiement de Wickr Enterprise. Si vous utilisez une zone hébergée publique Route 53, les enregistrements DNS et les certificats ACM pour ce nom de domaine seront automatiquement créés.
- `wickr/route53:hostedZoneName`- Nom de la zone hébergée Route 53 dans laquelle créer des enregistrements DNS.
- `wickr/route53:hostedZoneId`- ID de zone hébergée Route 53 dans laquelle créer des enregistrements DNS.

Cette méthode crée un certificat ACM en votre nom, ainsi que les enregistrements DNS pointant votre nom de domaine vers l'équilibreur de charge situé devant votre déploiement Wickr Enterprise.

Sans zone hébergée publique

Si votre compte ne possède pas de zone hébergée publique Route 53, un certificat ACM doit être créé manuellement et importé dans le CDK à l'aide de la valeur de `wickr/acm:certificateArn` contexte.

- `wickr/domainName`- Le nom de domaine à utiliser pour ce déploiement de Wickr Enterprise. Si vous utilisez une zone hébergée publique Route 53, les enregistrements DNS et les certificats ACM pour ce nom de domaine seront automatiquement créés.
- `wickr/acm:certificateArn`- L'ARN d'un certificat ACM à utiliser sur l'équilibreur de charge. Cette valeur doit être fournie si aucune zone hébergée publique de la Route 53 n'est disponible sur votre compte.

Importer un certificat dans ACM

Vous pouvez importer un certificat obtenu en externe à l'aide de la commande suivante :

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

La sortie sera l'ARN du certificat, qui doit être utilisé pour la valeur du paramètre de `wickr/acm:certificateArn` contexte. Il est important que le certificat téléchargé soit valide pour le`wickr/domainName`, sinon les connexions HTTPS ne pourront pas être validées. Pour plus d'informations, consultez la section Importation d'un certificat dans le guide de AWS Certificate Manager l'utilisateur.

Création d'enregistrements DNS

Aucune zone hébergée publique n'étant disponible, les enregistrements DNS doivent être créés manuellement une fois le déploiement terminé pour pointer vers l'équilibreur de charge situé devant votre déploiement Wickr Enterprise.

Déploiement dans un VPC existant

Si vous avez besoin d'un VPC existant, vous pouvez en utiliser un. Cependant, le VPC doit être configuré pour répondre aux spécifications nécessaires à EKS. Pour plus d'informations, consultez la section Afficher les exigences réseau d'Amazon EKS pour les VPC et les sous-réseaux dans le guide de l'utilisateur Amazon EKS, et assurez-vous que le VPC à utiliser répond à ces exigences.

En outre, il est vivement recommandé de vous assurer que vous disposez de points de terminaison VPC pour les services suivants :

- CLOUDWATCH
- CLOUDWATCH_LOGS
- EC2
- EC2_MESSAGES
- ECR
- ECR_DOCKER
- ELASTIC_LOAD_BALANCING
- KMS
- SECRETS_MANAGER
- SSM
- MESSAGES_SMS

Pour déployer des ressources dans un VPC existant, définissez les valeurs de contexte suivantes :

- `wickr/vpc:id`- L'ID VPC dans lequel déployer les ressources (par exemple`vpc-412beef`).
- `wickr/vpc:cidr`- Le IPv4 CIDR du VPC (par exemple)`172.16.0.0/16`.
- `wickr/vpc:publicSubnetIds`- Une liste de sous-réseaux publics séparés par des virgules dans le VPC. L'Application Load Balancer et les nœuds de travail EKS appelants seront déployés dans ces sous-réseaux (par exemple). `subnet-6ce9941,subnet-1785141,subnet-2e7dc10`
- `wickr/vpc:privateSubnetIds`- Une liste de sous-réseaux privés séparés par des virgules dans le VPC. Les nœuds de travail EKS et le serveur bastion seront déployés dans ces sous-réseaux (par exemple`subnet-f448ea8,subnet-3eb0da4,subnet-ad800b5`).
- `wickr/vpc:isolatedSubnetIds`- Une liste séparée par des virgules de sous-réseaux isolés dans le VPC. La base de données RDS sera déployée dans ces sous-réseaux (par exemple`subnet-d1273a2,subnet-33504ae,subnet-0bc83ac`).
- `wickr/vpc:availabilityZones`- Une liste de zones de disponibilité séparées par des virgules pour les sous-réseaux du VPC (par exemple). `us-east-1a,us-east-1b,us-east-1c`

Pour plus d'informations sur les points de terminaison VPC d'interface, consultez Accéder à un AWS service à l'aide d'un point de terminaison VPC d'interface.

Autres paramètres

Pour plus d'informations, consultez la section Valeurs de contexte.

Sangle Bootstrap

Si c'est la première fois que vous utilisez le CDK sur cette région en particulier Compte AWS , vous devez d'abord démarrer le compte pour commencer à utiliser le CDK.

```
npx cdk bootstrap
```

Déploiement

Ce processus prendra environ 45 minutes.

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

Une fois l'installation terminée, l'infrastructure a été créée et vous pouvez commencer à installer Wickr Enterprise.

Création d'enregistrements DNS

Cette étape n'est pas obligatoire si vous avez utilisé une zone hébergée publique lors de la configuration du CDK.

Le résultat du processus de déploiement inclura une valeur`WickrAlb.AlbDnsName`, qui est le nom DNS de l'équilibreur de charge. Le résultat ressemblera à ce qui suit :

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

Dans ce cas, le nom DNS est`Wickr-Alb-1Q5IBPJR4ZVZR-409483305.us-west-2.elb.amazonaws.com`. Il s'agit de la valeur à utiliser lors de la création d'un enregistrement CNAME ou A/AAAA (ALIAS) pour votre nom de domaine.

Si vous ne disposez pas du résultat du déploiement, exécutez la commande suivante pour afficher le nom DNS de l'équilibreur de charge :

```
aws cloudformation describe-stacks --stack-name WickrAlb \
```

```
    --query 'Stacks[0].Outputs[?OutputKey==`AlbDnsName`].OutputValue' \
    --output text
```

Générer une configuration KOTS

> **Warning**
>
> Ce fichier contient des informations sensibles concernant votre installation. Ne le partagez pas et ne l'enregistrez pas publiquement.

Le programme d'installation de Wickr Enterprise nécessite un certain nombre de valeurs de configuration relatives à l'infrastructure pour une installation réussie. Vous pouvez utiliser un script d'assistance pour générer les valeurs de configuration.

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

Si vous avez importé un certificat externe dans ACM lors de la première étape, transmettez l'`--ca-file`indicateur à ce script, par exemple :

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

Si vous recevez un message d'erreur indiquant que la pile n'existe pas, définissez la variable d'`AWS_REGION`environnement (`export AWS_REGION=us-west-2`) sur la région que vous avez sélectionnée et réessayez. Ou, si vous définissez la valeur du contexte`wickr/stackSuffix`, transmettez le suffixe avec le `--stack-suffix` drapeau.

Connexion au cluster Kubernetes

L'API Amazon EKS n'est accessible que via un hôte bastion créé dans le cadre du déploiement. Par conséquent, toutes les `kubectl` commandes doivent être exécutées sur l'hôte bastion lui-même ou être transmises par proxy via l'hôte bastion.

Connexion par proxy via le bastion

La première fois que vous vous connectez au cluster, vous devez mettre à jour votre fichier kubeconfig local à l'aide de la `aws eks update-kubeconfig` commande, puis le définir `proxy-url` dans votre configuration. Ensuite, chaque fois que vous souhaitez vous connecter au cluster, vous démarrez une session SSM avec l'hôte bastion pour transférer le port vers le proxy pour accéder à l'API.

Configuration unique

Il existe une valeur de sortie sur la `WickrEks` CloudFormation pile dont le nom commence par`WickrEnterpriseConfigCommand`. La valeur contient la commande complète nécessaire pour générer la configuration kubectl pour votre cluster. Cette sortie peut être visualisée à l'aide de la commande suivante :

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

Cela devrait générer une commande commençant par`aws eks update-kubeconfig`. Exécutez cette commande.

Ensuite, la configuration de Kubernetes doit être modifiée en fonction des requêtes proxy via l'hôte Bastion. Cela peut être fait à l'aide des commandes suivantes :

```
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
```

Si cela a fonctionné correctement, vous verrez une sortie comme `'Property "clusters.arn:aws:eks:us-west-2:012345678912:cluster/WickrEnterprise5B8BF472-1234a41c4ec48b7b615c6789d93dcce.proxy-url" set.'`

Port en avant vers le bastion

Pour vous connecter au cluster Amazon EKS, vous devez démarrer une session SSM afin de transférer les demandes vers le proxy exécuté sur votre hôte Bastion. La commande pour ce faire est fournie en sortie `BastionSSMProxyEKSCommand` sur la `WickrEks` pile. Exécutez la commande suivante pour afficher la valeur de sortie :

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

La commande qu'il émet commencera par`aws ssm start-session`. Exécutez cette commande pour démarrer un proxy local exécuté sur le port 8888 via lequel vous pouvez vous connecter au cluster Amazon EKS. Si le transfert de port fonctionne correctement, la sortie doit indiquer « En attente de connexions... ». Maintenez ce processus en cours pendant toute la durée nécessaire pour accéder au cluster Amazon EKS.

Si tout est correctement configuré, vous pourrez exécuter `kubectl get nodes` dans un autre terminal pour répertorier les nœuds de travail du 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
```

Installation de Wickr Enterprise

Une fois votre connexion au cluster Kubernetes établie, vous pouvez commencer à installer Wickr Enterprise à l'aide du plugin. `kubectl kots` Vous aurez besoin de votre fichier de licence KOTS (un `.yaml` fichier fourni par Wickr) et de votre fichier de valeurs de configuration, qui ont été enregistrés dans le fichier `wickr-config.json` dans la section Generate KOTS Config. Pour plus d'informations sur Generate KOTS Config, voir Generate KOTS Config.

Installation manuelle de Wickr Enterprise

La commande suivante lancera l'installation de Wickr Enterprise :

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

Vous serez invité à saisir un mot de passe pour la console d'administration KOTS. Enregistrez ce mot de passe car vous en aurez besoin pour mettre à niveau ou modifier la configuration de votre installation Wickr Enterprise à l'avenir.

Lorsque l'installation est terminée, un port local `kubectl kots` s'ouvre (généralement`http://localhost:8080`), qui donne accès à la console d'administration KOTS. Vous pouvez modifier ou surveiller l'état de votre installation de Wickr Enterprise sur ce site, ou commencer à configurer Wickr en consultant le nom de domaine que vous avez configuré pour votre installation dans votre navigateur.

Installation de Wickr Enterprise avec Lambda

Lors du déploiement du CDK, un Lambda est créé et invoqué pour terminer automatiquement l'installation de Wickr Enterprise en votre nom. Pour l'invoquer manuellement, ouvrez la AWS console et recherchez la fonction `WickrLambda-func*` lambda, sous l'onglet test, sélectionnez`test`, l'entrée n'est pas pertinente.

Après l'installation

Deux consoles Web sont disponibles pour gérer votre installation Wickr Enterprise : la console d'administration KOTS et la console d'administration Wickr.

> **Note**
>
> Apportez les modifications nécessaires pour refléter les politiques de sauvegarde et de journalisation de votre organisation (paramètres Amazon S3, journaux d'accès à Elastic Load Balancing, journaux de Amazon Virtual Private Cloud flux).

Console d'administration KOTS

Cette interface est utilisée pour gérer la version déployée de Wickr Enterprise. Vous pouvez consulter l'état de l'installation, modifier les configurations ou effectuer des mises à niveau. La console d'administration KOTS n'est accessible que via un port redirigé Kubernetes, qui peut être ouvert à l'aide de la commande suivante :

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

> **Note**
>
> Vous devez d'abord configurer votre connexion au bastion comme décrit dans la section Port forward to the bastion. Pour plus d'informations sur le transfert de port vers le bastion, voir Proxy des connexions via le bastion.

Lorsque le port forward est correctement configuré, la commande précédente affiche ce qui suit :

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

Utilisez l'URL fournie pour accéder à la console d'administration KOTS. Le mot de passe de connexion est celui que vous avez choisi lors de l'exécution `kubectl kots install` lors de l'installation. Si vous devez réinitialiser votre mot de passe, consultez Réinitialisation du mot de passe de la console d'administration KOTS.

Console d'administration Wickr

Cette interface est utilisée pour configurer votre installation Wickr Enterprise afin de configurer les réseaux, les utilisateurs et la fédération. Il est accessible via HTTPS sous le nom DNS que vous avez configuré pour pointer vers votre Load Balancer. Si le DNS a été configuré automatiquement avec une zone hébergée publique, le nom de domaine est la valeur de la valeur de `wickr/domainName` contexte.

Le nom d'utilisateur par défaut est`admin`, avec le mot de passe`Password123`. Il vous sera demandé de modifier ce mot de passe lors de votre première connexion.

Valeurs contextuelles

Les valeurs de contexte sont des paires clé-valeur qui peuvent être associées à une application, une pile ou une construction. Ils peuvent être fournis à votre application à partir d'un fichier (généralement dans le répertoire de votre projet `cdk.json` ou `cdk.context.json` dans le répertoire de votre projet) ou via la ligne de commande. CDK utilise des valeurs de contexte pour contrôler la configuration de l'application. Wickr Enterprise utilise les valeurs contextuelles du CDK pour contrôler les paramètres tels que le nom de domaine de votre installation Wickr Enterprise ou le nombre de jours pendant lesquels les sauvegardes RDS sont conservées.

Il existe plusieurs manières de définir les valeurs de contexte, mais nous vous recommandons de les modifier pour les adapter `cdk.context.json` à votre cas d'utilisation particulier. Seules les valeurs de contexte commençant par `wickr/` sont liées au déploiement de Wickr Enterprise.

Name (Nom)	Description	Par défaut
`wickr/licensePath`	Le chemin d'accès à votre licence KOTS (un `.yaml` fichier fourni par Wickr).	null
`wickr/domainName`	Le nom de domaine à utiliser pour ce déploiement de Wickr Enterprise. Si vous utilisez une zone hébergée publique Route 53, les enregistrements DNS et les certificats ACM pour ce nom de domaine seront automatiquement créés.	null
`wickr/route53:hostedZoneId`	ID de zone hébergée Route 53 dans laquelle créer des enregistrements DNS.	null
`wickr/route53:hostedZoneName`	Route 53 Nom de la zone hébergée dans laquelle créer les enregistrements DNS.	null

Name (Nom)	Description	Par défaut
`wickr/acm:certificateArn`	ARN d'un certificat ACM à utiliser sur le Load Balancer. Cette valeur doit être fournie si aucune zone hébergée publique de la Route 53 n'est disponible dans votre compte.	null
`wickr/caPath`	Chemin du certificat, uniquement requis lors de l'utilisation de certificats auto-signés.	null
`wickr/vpc:id`	ID du VPC dans lequel déployer les ressources. Nécessaire uniquement lors d'un déploiement dans un VPC existant. S'il n'est pas défini, un nouveau VPC sera créé.	null
`wickr/vpc:cidr`	IPv4 CIDR à associer au VPC créé. En cas de déploiement dans un VPC existant, définissez ce paramètre sur le CIDR du VPC existant.	172.16.0.0/16
`wickr/vpc:availabilityZones`	Liste des zones de disponibilité séparées par des virgules. Nécessaire uniquement lors d'un déploiement dans un VPC existant.	null

Name (Nom)	Description	Par défaut
`wickr/vpc:publicSubnetIds`	Liste des sous-réseaux publics séparés par des virgules. IDs Nécessaire uniquement lors d'un déploiement dans un VPC existant.	null
`wickr/vpc:privateSubnetIds`	Liste de sous-réseaux privés séparée par des virgules. IDs Nécessaire uniquement lors d'un déploiement dans un VPC existant.	null
`wickr/vpc:isolatedSubnetIds`	Liste séparée par des virgules des sous-réseaux isolés IDs pour la base de données RDS. Nécessaire uniquement lors d'un déploiement dans un VPC existant.	null
`wickr/rds:deletionProtection`	Activez la protection contre la suppression sur les instances RDS.	true
`wickr/rds:removalPolicy`	Politique de suppression pour les instances RDS « snapshot », « destroy » ou « retain ».	instantané
`wickr/rds:readerCount`	Nombre d'instances de lecteur à créer dans le cluster RDS.	1
`wickr/rds:instanceType`	Type d'instance à utiliser pour les instances RDS.	r6g.xlarge

Name (Nom)	Description	Par défaut
`wickr/rds:backupRetentionDays`	Nombre de jours pendant lesquels les sauvegardes sont conservées.	7
`wickr/eks:namespace`	Espace de noms par défaut pour les services Wickr dans EKS.	osier
`wickr/eks:defaultCapacity`	Nombre de nœuds de travail EKS pour l'infrastructure de messagerie.	3
`wickr/eks:defaultCapacityCalling`	Nombre de nœuds de travail EKS pour l'infrastructure d'appel.	2
`wickr/eks:instanceTypes`	Liste séparée par des virgules des types d'instances à utiliser pour les nœuds de travail de messagerie EKS.	m5.xlarge
`wickr/eks:instanceTypesCalling`	Liste des types d'instances séparés par des virgules à utiliser pour appeler les nœuds de travail EKS.	c5n.large
`wickr/eks:enableAutoscaler`	Active la fonctionnalité Cluster Autoscaler pour EKS.	true
`wickr/s3:expireAfterDays`	Nombre de jours après lesquels les téléchargements de fichiers seront supprimés du compartiment S3.	1095

Name (Nom)	Description	Par défaut
`wickr/eks:clusterVersion`	Versions de cluster, y compris la version Kubernetes, la version KubectLayer, la version AlbController, la version et plus encore. nodeGroupRelease	1,27
`wickr/stackSuffix`	Suffixe à appliquer aux noms de CloudFormation pile.	"
`wickr/autoDeployWickr`	Déployez automatiquement l'application Wickr avec Lambda.	true

Détruire des ressources

Pour supprimer tout ce qui a été créé par cette AWS CDK application, vous devez supprimer la `WickrRds` pile avant toutes les autres piles.

Pour que les ressources Amazon RDS soient correctement supprimées, la protection contre la suppression doit être désactivée et la politique de suppression doit être définie sur `snapshot` ou`destroy`. Si ce ne sont pas les paramètres actuels, modifiez les `wickr/rds:removalPolicy` valeurs `wickr/rds:deletionProtection` et dans votre AWS CDK contexte et redéployez la pile Amazon RDS en exécutant. `npx cdk deploy -e WickrRds`

Une fois que la protection contre les suppressions et la politique de suppression sont correctement définies, exécutez la `cdk destroy` commande pour la `WickrRds` pile :

```
npx cdk destroy WickrRds
```

Lorsque la `WickrRds` pile a fini de se détruire, les CloudFormation piles restantes peuvent être détruites à l'aide de la commande suivante :

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

Résolution des problèmes

Supprimer l'espace de noms Wickr

Si vous devez supprimer l'espace de `wickr` noms pour recommencer, il est important de sauvegarder d'abord tous les comptes de service créés par CDK dans cet espace de noms. Ces comptes de service permettent aux services Wickr de communiquer avec eux AWS APIs via des rôles IAM. Sans eux, les tâches telles que le téléchargement de fichiers via Amazon Simple Storage Service (Amazon S3) ne fonctionneront plus.

Utilisez la commande suivante pour sauvegarder les comptes de service, supprimer et recréer l'espace de `wickr` noms et les comptes de service appropriés :

```
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
```

Réinitialisation du mot de passe de la console d'administration KOTS

Vous pouvez réinitialiser le mot de passe de votre console d'administration KOTS à l'aide de la commande suivante :

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

Lorsque vous modifiez ce mot de passe, vous souhaiterez peut-être également mettre à jour le secret du Gestionnaire de `wickr/kots` Secrets, bien qu'il ne soit généralement pas réutilisé par aucune automatisation.

Problèmes de connexion au cluster EKS avec Bastion

Si votre connexion au cluster EKS via le bastion semble lente ou expire de temps en temps, le message d'erreur suivant peut s'afficher lors de l'exécution des `kubectl` commandes :

net/http : demande annulée en attendant la connexion (Client.Timeout dépassé pendant l'attente des en-têtes)

Ce problème peut souvent être résolu en vous connectant à l'hôte Bastion via SSM (voir le document `BastionSSMCommand` sur la WickrEks pile) et en redémarrant le service : `tinyproxy`

```
sudo systemctl restart tinyproxy
```

Installation personnalisée

Dans la section Installation personnalisée, vous apprendrez comment installer Wickr Enterprise.

Rubriques

- Prérequis
- Architecture
- Installation
- Paramètres d'entrée
- Paramètres de base de données
- Stockage de fichiers S3
- Paramètres de demande de volume persistants
- Paramètres du certificat TLS
- Paramètres d'appel
- Autoscaler du cluster Kubernetes (facultatif)
- Sauvegardes
- Installation d'un entrefer
- Console d'administration Wickr
- FAQ

Prérequis

Avant de commencer à installer Wickr Enterprise, vérifiez que les conditions suivantes sont remplies.

Configuration matérielle requise

Wickr Enterprise nécessite un cluster Kubernetes pour fonctionner. Il est possible d'opérer sur un seul nœud avec le mode Low Resource activé, mais cela n'est pas recommandé pour une utilisation en production générale. Dans un déploiement en production, nous recommandons un minimum de trois nœuds de travail de messagerie ainsi qu'un minimum de deux nœuds de travail d'appel.

Un nœud de travail doit avoir les spécifications minimales suivantes.

- 2 à 4 cœurs de processeur

- 8 Go de RAM
- 200 Go d'espace disque

Configuration matérielle minimale requise

Un cluster de nœuds de travail unique fonctionnant en mode faibles ressources nécessite un minimum de 3 000 Mo de processeur et 5 846 Mo de RAM. Cela n'inclut pas les pods du système Kube.

Besoins en ressources par module

Nom du pod	Propriétaire	CPU	Mémoire
admin-api	Osier	100 m	256 Mi
directory	Osier	100 m	128 milles
expirateur	Osier	100 m	128 milles
proxy de fichiers	Osier	100 m	256 Mi
oidc	Osier	100 m	128 milles
opensearch	Osier	500 m	100 milles
orville	Osier	50 m	128 milles
orville-redis	Osier	50 m	128 milles
dispositif push	Osier	100 m	128 milles
lapin mq	Osier	50 m	256 Mi
réagir	Osier	100 m	64 milles
reçus	Osier	250 m	128 milles
redis	Osier	50 m	128 milles
serveur-api	Osier	250 m	256 Mi

Nom du pod	Propriétaire	CPU	Mémoire
tableau électrique	Osier	250 m	512 milles
kotsadm	KOTS	50 m	50 milles
kotsadm-minio	KOTS	100 m	512 milles
kotsadm-ralite	KOTS	200 m	1 Gi
mini-opérateur	S3 interne	200 m	256 Mi
mini-locataire	S3 interne	100 m	256 Mi
mysql-primaire	MySQL interne	100 m	512 milles
mysql-secondaire	MySQL interne	100 m	512 milles

Exigences en matière de stockage

Wickr Enterprise nécessite une valeur par défaut StorageClass à utiliser lors de la création de réclamations de volume persistantes. Lors d'un déploiement dans un environnement isolé ou sur site, vous devrez peut-être en configurer un pour votre cluster. L'une des options disponibles est Longhorn. L'espace disque recommandé varie en fonction de l'utilisation de l'option S3 interne et de l'option Mysql interne et de la quantité d'espace dont vous souhaitez disposer pour le téléchargement de fichiers.

- Mise en cache interne des images : ~60 Go
- RabbitMQ : 24 Gi par défaut/8 Gi en mode ressources limitées
- Redis : 24 Gi par défaut/8 Gi en mode ressources faibles
- OpenSearch: 24 Gi par défaut/8 Gi en mode ressources faibles
- Mysql interne : 80 Gi par défaut/20 Gi en mode faible ressource
- S3 interne : 160 Gi par défaut/2 Gi en mode ressources faibles
- KOTA Mini : 4 Go
- MOTS Realite : 1 Go

Taille de stockage minimale

- 377 Gi par défaut avec S3 interne et Mysql interne
- 111 Gi en mode ressources limitées

Exigences relatives aux versions de Kubernetes

Wickr Enterprise s'appuie sur Replicated KOTS. Replicated, une plateforme commerciale de distribution de logiciels, fournit une liste des versions actuellement prises en charge de Kubernetes. Pour plus d'informations, consultez la section Compatibilité des versions de Kubernetes.

Exigences en matière de logiciels

Wickr Enterprise nécessite un cluster Kubernetes et KOTS pour fonctionner. Reportez-vous à la documentation KOTS pour connaître les versions de système d'exploitation et de Kubernetes prises en charge. Pour plus d'informations, consultez la section Configuration minimale requise.

Système hôte pour développeurs

Système d'exploitation — Les commandes de cette documentation sont conçues pour fonctionner sous Linux, macOS ou Windows avec WSL (Windows Subsystem for Linux) installé.

Services internes Stateful

Wickr Enterprise peut fournir des services internes pour la base de données MySQL et le stockage compatible S3, mais pour une utilisation générale en production, il est recommandé de fournir ces services en externe au cluster Kubernetes.

- Base de données MySQL 5.7
  - Base de données Amazon RDS MySQL 5.7 ou MySQL 5.7 (externe)
  - Tableau Mysql Bitnami Helm (interne)
  - Stockage de fichiers
    - Amazon S3 ou fournisseur de stockage compatible S3 (externe)
    - Tableau du casque Minio Operator (interne)

Exigences réseau

Wickr Enterprise nécessite un FQDN, des certificats SSL et des ports TCP et UDP ouverts spécifiques.

- FQDN : domaine ou sous-domaine à utiliser par le déploiement de Wickr Enterprise.
- Certificat SSL : paire de clés de certificat SSL signée par une autorité de certification publique ou paire de clés de certificat autosignée. Le certificat doit répertorier le FQDN dans le nom commun et également sous forme d'entrée DNS du SAN. Le certificat doit également activer l'extension ServerAuth. extendedKeyUsage
- Les installations en ligne nécessiteront un accès de sortie aux ressources répliquées et tierces. Replicated conserve une liste de ses adresses IP. Pour plus d'informations, consultez la section Adresses IP répliquées. Replicated tient également à jour une liste des ressources tierces nécessaires. Pour plus d'informations, consultez la section Ouvertures de pare-feu pour les installations en ligne.
- Les installations isolées nécessitent l'accès à un registre de conteneurs privé.

Nœuds de messagerie

Les nœuds de messagerie ne nécessitent pas d' IPV4 adresse publique et doivent être situés dans un sous-réseau privé. Le trafic de messages entrera dans le cluster par le biais du LoadBalancer ou Ingress.

Nœuds d'appel

Les nœuds appelants nécessitent une IPV4 adresse publique et doivent donc se trouver dans un sous-réseau public. Le contenu des appels est transféré via UDP par défaut. Lorsque les appels TCP sont activés, le proxy TCP accepte les connexions sur TCP 443 et les transmet au service Orville.

- TCP : 443 Appel d'un proxy TCP
- UDP : 16384-16484 Streams Audio/Video

Accès à l'installation et à la configuration

L'accès à la console d'administration KOTS pour l'installation et la configuration s'effectue via une redirection de port Kubernetes.

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

Exigences en matière de licence

L'installation nécessitera un fichier de licence au format .yaml, qui vous sera fourni par le support Wickr.

Architecture

Architecture de production recommandée

Le schéma ci-dessous montre Wickr Enterprise configuré conformément aux recommandations pour la production, avec les services MySQL et Object Storage situés en dehors du cluster Kubernetes.

Architecture interne ou de test

Le schéma ci-dessous montre la configuration de Wickr Enterprise, en utilisant les services internes MYSQL et Object Storage. Bien qu'il puisse répondre aux besoins spécifiques de certains déploiements, il n'est pas recommandé pour une utilisation en production générale.

Installation

1. Installez kubectl et kots CLI.
2. Connectez-vous au cluster Kubernetes.
3. Obtenez le fichier de licence Wickr Enterprise auprès du Support Wickr.
4. Installez Wickr Enterprise à l'aide de la commande suivante.

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

> **Note**
>
> license.yaml représente le fichier de licence que vous avez fourni.

Après l'installation initiale, la console d'administration KOTS fournira des options de gestion et de configuration au niveau du cluster.

Console d'administration KOTS

Cette interface est utilisée pour gérer la version déployée de Wickr Enterprise. Vous pouvez voir l'état de l'installation, modifier les configurations ou effectuer des mises à niveau de Wickr Enterprise. La

console d'administration KOTS n'est accessible que via un port redirigé Kubernetes, qui peut être ouvert à l'aide de la commande suivante :

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

Paramètres d'entrée

Contrôleur d'entrée

Wickr Enterprise prend en charge quatre types de contrôleurs d'entrée :

- LoadBalancer (Par défaut)
  - L'objet loadbalancer peut nécessiter une configuration explicite dans les installations entièrement sur site, même s'il est souvent fourni par des fournisseurs de cloud.
  - Déploie le service Ingress Controller (ingress-nginx) avec le type de service. LoadBalancer Cela nécessite que le cluster Kubernetes s'exécute sur une plate-forme qui prend en charge les équilibreurs de charge externes.
- ALB existant
  - Attache le contrôleur d'entrée à un ALB existant.
  - Vous devrez fournir l'ARN du groupe cible Application Load Balancer existant.
- NodePort
  - Le contrôleur d'entrée (ingress-nginx) sera configuré pour utiliser le type de NodePort service, qui ouvre un port sur tous les nœuds du cluster Kubernetes et transmet le trafic vers l'entrée. Le trafic client peut ensuite être dirigé vers ces nœuds via le DNS ou un équilibreur de charge externe.
  - Vous pouvez choisir une plage de ports comprise entre 1 et 65535, ou un port aléatoire compris entre 30000 et 32767 sera utilisé.
- Ingress
  - Apportez votre propre contrôleur d'entrée. Cette configuration acceptera un nom de classe d'entrée que les services utiliseront ensuite dans leurs manifestes d'entrée. Cela implique que le contrôleur d'entrée dispose d'une connectivité externe déjà configurée via un autre mécanisme d'équilibrage de charge.
  - Actuellement, seul le contrôleur ingress-nginx est pris en charge.

Nom d'hôte Wildcard

Par défaut, les routes d'entrée seront définies avec une valeur d'hôte de `*`. Désactivez ce paramètre pour utiliser le nom d'hôte défini pour le serveur Wickr Enterprise. Un nom d'hôte générique est requis pour les noms d'hôtes basés sur IP.

Paramètres de base de données

Wickr Enterprise nécessite une base de données MySQL 5.7. Nous vous recommandons d'utiliser une base de données externe à votre cluster Kubernetes, telle qu'Amazon RDS, mais vous avez également la possibilité de déployer une base de données MySQL interne à l'intérieur du cluster Kubernetes dans le cadre de l'installation.

Paramètres de base de données externes

- Nom d'hôte : nom d'hôte ou adresse IP du serveur de base de données.
- Nom d'hôte du lecteur : nom d'hôte ou adresse IP d'un point de terminaison en lecture seule pour le serveur de base de données (si disponible).
- Port : port sur lequel MySQL sera accessible.
- Nom de la base de données : nom de la base de données créée sur le serveur.
- Nom d'utilisateur : utilisateur autorisé à accéder à la base de données.
- Mot de passe : mot de passe de cet utilisateur.
- Certificat CA : certificat PEM pour la connexion à la base de données via TLS.

> **Note**
>
> Assurez-vous que votre installation MySQL 5.7 utilise le jeu de caractères latin1 par défaut avec le classement latin1_swedish_ci. Cela peut être accompli en vérifiant que votre serveur MySQL est démarré avec les indicateurs suivants : `"--character-set-server latin1", "--collation-server latin1_swedish_ci"`

Paramètres internes de la base de données

Le type de base de données interne en déploiera deux StatefulSets dans votre cluster pour une base MySQL principale et une base secondaire avec réplication binaire. Le secondaire ne reçoit aucun trafic et n'est disponible que pour la reprise après sinistre et les sauvegardes.

Taille de stockage : taille (en gibioctets) des volumes persistants pour les pods de base de données.

Augmenter la taille du stockage MySQL

> **Note**
>
> Le type de votre volume StorageClass doit prendre en charge l'extension du volume afin d'augmenter la taille de stockage. Pour plus d'informations, consultez la section Expansion du volume.

Les services MySQL utilisés dans Wickr Enterprise sont déployés sous forme de StatefulSet ressources dans Kubernetes. StatefulSets rendre immuables de nombreuses propriétés de la ressource, y compris les modèles Persistent Volume Claim. Pour contourner l'immuabilité de StatefulSets, les actions suivantes doivent être effectuées pour augmenter la taille des volumes utilisés par MySQL.

1. Modifiez les demandes de volume persistant pour `data-mysql-primary-0` et`data-mysql-secondary-0`.

   1. `kubectl -n wickr edit pvc data-mysql-primary-0`. Set `spec.resources.requests.storage`à la taille de stockage souhaitée.
   2. `kubectl -n wickr edit pvc data-mysql-secondary-0`. Set `spec.resources.requests.storage`à la taille de stockage souhaitée.

2. Supprimez les pods existants StatefulSets, mais laissez les pods en passant le `--cascade=orphan` drapeau.

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

3. Dans l'interface utilisateur KOTS, mettez à jour le paramètre de taille de stockage pour qu'il corresponde à la valeur que vous avez définie à l'étape 1. Enregistrez et déployez cette configuration.

4. Redémarrez le StatefulSets pour augmenter les volumes et remettre les services MySQL en ligne.

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

Stockage de fichiers S3

Wickr Enterprise nécessite un service de stockage compatible S3. Nous vous recommandons d'utiliser un service S3 externe à votre cluster Kubernetes, tel qu'Amazon S3, mais vous avez également la possibilité de déployer un service S3 interne au sein du cluster Kubernetes dans le cadre de l'installation.

Paramètres S3 externes

- Nom du compartiment : nom du compartiment S3 dans lequel les téléchargements de fichiers seront stockés.
- Région : AWS région du compartiment S3.
- Point de terminaison : définissez le point de terminaison que Wickr utilisera pour interagir avec l'API S3. La valeur par défaut est le point de terminaison du service S3 de la région.
- Nom du compte du service Fileproxy : Amazon S3 uniquement. Le nom d'un compte de service Kubernetes existant à utiliser pour s'authentifier auprès de S3 à l'aide de rôles IAM pour les comptes de service.
- Clé d'accès S3 externe : il s'agit de votre clé d'accès S3 existante.
- Clé secrète S3 externe : il s'agit de votre clé secrète S3 existante.

Paramètres internes du S3

Le type S3 interne déploiera par défaut 4 pods de serveur MinIO contenant chacun 4 réclamations de volume persistantes. La configuration par défaut utilise le codage d'effacement de MinIO pour augmenter la tolérance aux pannes.

- Nombre de serveurs S3 internes : nombre de pods de serveur MinIO à créer. La valeur par défaut est de 4 pour un déploiement tolérant aux pannes. Cette valeur peut être définie comme inférieure à 1 pour un development/test déploiement.
- Nombre de volumes S3 internes : nombre de volumes MinIO à créer dans chaque pod de serveur MinIO. La valeur par défaut est de 4 pour un déploiement tolérant aux pannes. Cette valeur peut être définie comme inférieure à 1 pour un development/test déploiement.
- Taille du volume S3 interne : taille en Go des volumes MinIO créés dans les pods du serveur MinIO, la valeur par défaut est de 10 Go.

- Un déploiement S3 interne par défaut utilisera 4 serveurs dont 4 PVCs. Chaque PVC a une capacité de 10 Gi, ce qui donne un stockage brut de 160 Gi avec un stockage codé d'effacement de 120 Gi à la disposition des utilisateurs.
- Le calculateur de codage Minio Erasure est disponible. Pour plus d'informations, consultez la section Calculateur de code d'effacement.

Paramètres de demande de volume persistants

Wickr Enterprise a besoin de Persistent Volume Claims pour stocker des données dynamiques. Ce paramètre vous permet de spécifier le nom de la classe de stockage que vous souhaitez utiliser. Si ce champ est laissé vide, Wickr essaiera d'utiliser la classe de stockage par défaut. La modification de la classe de stockage après le déploiement de Wickr n'est pas prise en charge.

Les fournisseurs de cloud fournissent souvent une option par défaut StorageClass pour les réclamations de volume persistantes, mais dans le cas d'installations entièrement sur site, une configuration explicite peut être requise à l'aide d'un service tiers tel que Longhorn.

Paramètres du certificat TLS

Téléchargez un certificat PEM et une clé privée pour mettre fin au protocole TLS. Le nom alternatif du sujet sur le certificat doit correspondre au nom d'hôte configuré dans les paramètres de votre déploiement Wickr Enterprise.

Pour le champ de chaîne de certificats, concaténez tous les certificats intermédiaires (si nécessaire) avec le certificat de l'autorité de certification racine avant le téléchargement.

Let's Encrypt

Sélectionnez cette option pour générer automatiquement un certificat à l'aide de Let's Encrypt. Les certificats sont émis à l'aide du défi HTTP-01 par l'intermédiaire de l'opérateur cert-manager.

Le défi HTTP-01 nécessite que le nom DNS souhaité corresponde au point d'entrée de votre cluster (généralement un Load Balancer) et que le trafic vers le port TCP 80 soit ouvert au public. Ces certificats sont de courte durée et seront renouvelés régulièrement. Il est nécessaire de laisser le port 80 ouvert pour permettre aux certificats de se renouveler automatiquement.

> **Note**
>
> Cette section fait explicitement référence au certificat utilisé par l'application Wickr Enterprise elle-même.

Certificat épinglé

Wickr Enterprise nécessite l'épinglage des certificats lors de l'utilisation de certificats auto-signés ou de certificats non approuvés par les appareils clients. Si le certificat présenté par votre Load Balancer est auto-signé ou signé par une autorité de certification différente de celle de l'installation de Wickr Enterprise, téléchargez-le ici pour que les clients puissent l'épingler à la place.

Dans la plupart des cas, ce paramètre n'est pas obligatoire.

Fournisseurs de certificats

Si vous envisagez d'acheter un certificat à utiliser avec Wickr Enterprise, vous trouverez ci-dessous une liste des fournisseurs dont les certificats sont connus pour fonctionner correctement par défaut. Si un fournisseur est répertorié ci-dessous, ses certificats ont été validés explicitement avec le logiciel.

- Digicert
- SSL rapide

Génération d'un certificat auto-signé

Si vous souhaitez créer votre propre certificat auto-signé à utiliser avec Wickr Enterprise, l'exemple de commande ci-dessous contient tous les indicateurs requis pour la génération.

```
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"
```

Si vous souhaitez créer un certificat auto-signé basé sur l'adresse IP, utilisez plutôt la commande suivante. Pour utiliser le certificat basé sur l'adresse IP, assurez-vous que le champ Wildcard Hostname est activé dans les paramètres d'entrée. Pour plus d'informations, consultez la section Paramètres d'entrée.

```
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**
>
> Remplacez $YOUR_DOMAIN dans l'exemple par le nom de domaine ou l'adresse IP que vous souhaitez utiliser.

Paramètres d'appel

- Exiger des nœuds d'appel : lorsque ce paramètre est activé, les services d'appel de Wickr ne sont déployés que sur les nœuds Kubernetes portant l'étiquette. `role=calling` Désactivez ce paramètre pour déployer les services d'appel et de messagerie sur les mêmes nœuds ou pour les déploiements à nœud unique.

  Vous souhaiterez généralement également désactiver le proxy TCP appelant lorsque ce paramètre est désactivé, car le service de proxy TCP s'exécute sur le port 443.
- Activer le proxy TCP : ce paramètre contrôle si le service pour le mode de secours TCP sur les appels est déployé ou non. Désactivez ce paramètre si d'autres services s'exécutent sur 443/TCP ou si vous n'avez pas besoin du mode de secours TCP pour les appels.
- Découverte automatique des adresses IP publiques du serveur : lorsque ce paramètre est activé, les services d'appel découvrent leur adresse IP publique en envoyant des requêtes HTTPS à https://ipv4.icanhazip.com/ ethttps://ipv6.icanhazip.com/. Lorsqu'il est désactivé, vous devez activer le paramètre « Utiliser l'adresse IP principale de l'hôte pour le trafic d'appels » ou « Remplacer le nom d'hôte », sinon les services d'appel ne démarreront pas.
- Utiliser l'adresse IP principale de l'hôte pour le trafic d'appels : utilisez l'adresse IP principale des nœuds Kubernetes pour les services d'appel. Cela implique que tous les clients Wickr peuvent se connecter à vos nœuds Kubernetes sur l'adresse IP principale du nœud, telle que présentée dans `status.hostIP` l'API Downward.
- Modification du nom d'hôte : fournissez un nom d'hôte ou une adresse IP à renvoyer comme point de connectivité pour les services d'appel. Ce paramètre ne doit être utilisé que lors de l'exécution d'un seul serveur d'appel, car la même valeur est renvoyée pour toutes les répliques du service. Lorsqu'un remplacement de nom d'hôte est défini et que le paramètre « utiliser l'adresse IP principale de l'hôte » est activé, le paramètre de l'adresse IP principale de l'hôte est prioritaire.

Autoscaler du cluster Kubernetes (facultatif)

Kubernetes Cluster Autoscaler est une valeur de configuration facultative pour l'installation de Wickr Enterprise. Cela vous aidera à dimensionner vos groupes de nœuds Kubernetes en cas d'augmentation du trafic ou d'autres restrictions de ressources susceptibles d'entraîner de mauvaises performances.

L'installation de Wickr Enterprise prend en charge 3 intégrations de fournisseurs de cloud : AWS Google Cloud et Azure. Chaque fournisseur de cloud a des exigences différentes pour cette intégration. Veuillez suivre les instructions spécifiques à votre fournisseur de cloud ci-dessous pour activer cette fonctionnalité.

AWS

Si vous n'avez pas utilisé le WickrEnterprise CDK pour installer votre environnement Wickr AWS, vous devrez prendre des mesures supplémentaires pour activer le Cluster Autoscaler.

1. Ajoutez les balises suivantes à vos groupes de nœuds. Cela permet au Cluster Autoscaler de découvrir automatiquement les nœuds appropriés.
    1. `k8s.io/cluster-autoscaler/`**`clusterName`** `= owned`où ClusterName est le nom de votre cluster Kubernetes
    2. `k8s.io/cluster-autoscaler-enabled = true`
2. Ajoutez un compte de service Kubernetes dans l'espace de noms du système kube et associez-le à une politique IAM qui autorise le dimensionnement automatique et les actions ec2. Pour plus d'informations et des instructions détaillées, consultez la section Configuration d'un compte de service Kubernetes pour assumer un rôle IAM dans le guide de l'utilisateur Amazon EKS.
    1. Vous devrez utiliser l'espace de noms « kube-system » lors de la configuration du compte de service
    2. La politique suivante peut être utilisée pour le compte de service :

       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"
        }
    ]
}
```

Dans l'interface utilisateur répliquée, lors de la configuration du Cluster Autoscaler, sélectionnez-le AWScomme fournisseur de cloud et fournissez le nom du compte de service que vous avez créé ci-dessus pour demander au Cluster Autoscaler d'utiliser ce compte de service.

Cloud de Google

Il est fortement recommandé d'utiliser les fonctionnalités de mise à l'échelle automatique intégrées de GKE pour le pilote automatique et les clusters standard. Toutefois, si vous souhaitez procéder à cette intégration, les conditions suivantes doivent être remplies avant de procéder.

Prérequis:

1. Les groupes d'instances gérés (MIG) doivent être créés avec une portée de sécurité incluant au minimum les ressources « lecture/écriture » du moteur de calcul. Cela ne peut pas être ajouté au MIG ultérieurement pour le moment.
2. La fédération des identités de charge de travail doit être activée sur le cluster. Vous pouvez l'activer sur un cluster existant en exécutant : `gcloud container clusters update ${CLUSTER_NAME} --workload-pool=${PROJECT_ID}.svc.id.goog`
3. Un compte de service Google Cloud Platform (GCP) avec accès au rôle « Roles/compute.InstanceAdmin.v1`. Cela peut être créé à l'aide des instructions suivantes :

```
# 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) fournit un dimensionnement automatique du cluster intégré pour la plupart des déploiements et il est vivement recommandé d'utiliser ces méthodes pour le dimensionnement automatique du cluster. Toutefois, si vos exigences sont telles que ces méthodes ne fonctionnent pas, nous avons fourni une intégration Kubernetes Cluster Autoscaler pour Azure Kubernetes Service. Pour utiliser cette intégration, vous devez recueillir les informations suivantes et les placer dans la configuration du panneau d'administration KOTS sous Cluster Autoscaler après avoir sélectionné Azure comme fournisseur de cloud.

Authentification Azure

ID d'abonnement : L'identifiant d'abonnement peut être obtenu via le portail Azure en suivant la documentation officielle. Pour plus d'informations, consultez Obtenir un abonnement et un locataire IDs sur le portail Azure.

Les paramètres suivants peuvent être obtenus en créant un AD Service Principal à l'aide de l'utilitaire de ligne de commande az.

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

Identifiant de l'application :

Mot de passe du client :

Numéro du locataire :

Configuration d'Azure Cluster Autoscaler

Outre les exigences d'authentification, les champs suivants sont nécessaires au bon fonctionnement de l'autoscaler du cluster. Les commandes permettant d'obtenir ces informations ont été fournies pour des raisons pratiques, mais elles peuvent nécessiter certaines modifications en fonction de votre configuration AKS spécifique.

Groupe de ressources de nœuds gérés Azure : cette valeur est le groupe de ressources gérées créé par Azure lorsque vous avez établi le cluster AKS et non le groupe de ressources que vous avez défini. Pour obtenir cette valeur, vous avez besoin du CLUSTER_NAME et du RESOURCE_GROUP datant de la création du cluster. Une fois que vous avez obtenu ces valeurs, vous pouvez les obtenir en exécutant :

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

Nom VMSS du pool de nœuds d'application : il s'agit du nom du Virtual Machine Scaling Set (VMSS) associé à votre pool de nœuds AKS pour l'application Wickr. Il s'agit de la ressource qui sera redimensionnée à la hausse ou à la baisse en fonction des besoins de votre cluster. Pour obtenir cette valeur, vous pouvez exécuter la commande az suivante :

```
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 Nom VMSS du pool de nœuds (facultatif) : il s'agit du nom du VMSS associé à votre pool de nœuds appelant, si vous en avez un. Pour obtenir cette valeur, vous pouvez exécuter une version modifiée de la commande pour le nom VMSS du pool de nœuds d'application en remplaçant la valeur CLUSTER_NODEPOOL_NAME par le nom du pool de nœuds de votre pool de nœuds appelant.

Sauvegardes

Wickr Enterprise utilise Velero à des fins de Backup. Velero fournit les outils nécessaires pour sauvegarder et restaurer les ressources du cluster Kubernetes et les volumes persistants, qu'ils fonctionnent chez un fournisseur de cloud ou sur site.

Sauvegardes Velero avec Minio : Actuellement, les sauvegardes Velero ne sont activées que pour Minio en mode ressources limitées.

Installation à l'aide de la documentation Velero

- Installez la CLI Velero. Pour plus d'informations, consultez la section Installation de la CLI Velero.
- Installez Velero sur votre cluster et configurez le stockage en fonction de votre fournisseur :
  - AWS.
  - GCP.
  - Azure.
  - Autres fournisseurs.

Installation d'un entrefer

Wickr Enterprise et KOTS prennent tous deux en charge le déploiement dans un cluster Kubernetes entièrement aéré. Vous devez fournir l'accès à un registre d'images Docker privé accessible depuis le cluster Kubernetes airgapped. Le registre d'images Docker privé fourni à KOTS doit être sécurisé par une username/password authentification pour fonctionner correctement à cette fin. KOTS utilisera le registre d'images Docker privé pour héberger toutes les images Wickr Enterprise.

- Licence Wickr Enterprise .yaml avec airgap activé (contactez le service commercial ou l'équipe de support client de Wickr)
- Bundle d'archives Wickr Enterprise wickr.airgap (contactez le service commercial ou l'équipe de support client de Wickr)
- Accès à un registre d'images Docker privé.
- Accès à un cluster Kubernetes déployé dans l'environnement airgap.
- Kubectl installé.
- CLI KOTS installée.
- kotsadm.tar.gz téléchargé.

Exécutez les commandes suivantes pour déployer KOTS et Wickr Enterprise sur votre cluster Kubernetes Airgapped. Ces commandes téléchargent les images d'administration KOTS et les images Wickr Enterprise dans le registre d'images Docker privé. Une fois les commandes terminées, vous serez invité à accéder à la console d'administration KOTS pour terminer l'installation de Wickr Enterprise comme ci-dessus.

```
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
```

Notification mobile pour les installations d'airgap

Des listes d'autorisations réseau supplémentaires sont nécessaires pour les notifications push envoyées par le serveur principal aux clients mobiles. Cette exigence est due à la manière dont Apple iOS et Google Android implémentent cette fonctionnalité pour les appareils hors ligne et en arrière-plan. Reportez-vous à la documentation de ces services et autorisez la liste des adresses IP et des ports spécifiés.

- iOS
- Android

Console d'administration Wickr

L'interface de la console d'administration Wickr est utilisée pour administrer l'application Wickr Enterprise elle-même. Il peut être utilisé pour configurer des réseaux, des utilisateurs, des fédérations, etc. Il est accessible via HTTPS sous le nom DNS que vous avez configuré pour pointer vers votre Load Balancer. Le nom d'utilisateur par défaut est admin, avec le mot de passe Password123. Il vous sera demandé de modifier ce mot de passe lors de votre première connexion.

FAQ

Q : Mon déploiement échoue avec l'erreur suivante dans helm stderr :

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

R : Cela peut se produire lorsque la journalisation du débogage est activée. Désactivez la journalisation du débogage, supprimez les tâches problématiques et réessayez.

Cluster intégré pour Wickr Enterprise

L'option d'installation en cluster intégrée pour Wickr Enterprise fournit une petite offre d'installation efficace pour le produit Wickr Enterprise. Il utilise le cluster intégré répliqué pour fournir une petite installation Kubernetes utilisant des k0s sur laquelle Wickr Enterprise peut être installé. L'utilisation de cette méthode d'installation minimise les exigences en matière de compétences techniques ainsi que les exigences matérielles globales pour une installation de Wickr Enterprise en fournissant une solution « all-in-one » au détriment de la résilience et de la haute disponibilité.

Rubriques

- Commencer à utiliser le cluster intégré Wickr Enterprise
- Exigences relatives au cluster intégré Wickr Enterprise
- Installation du cluster intégré Wickr Enterprise (standard)
- Configuration de console d'administration KOTS
- Exigences d'installation communes supplémentaires

Commencer à utiliser le cluster intégré Wickr Enterprise

Pour commencer à utiliser l'option de cluster intégré Wickr Enterprise, contactez le support pour recevoir une licence. Si vous possédez une licence existante et que vous souhaitez utiliser cette option, contactez le support pour obtenir de l'aide pour mettre à jour votre licence existante et des instructions d'installation supplémentaires.

Exigences relatives au cluster intégré Wickr Enterprise

Avant de commencer à installer le cluster intégré Wickr Enterprise, vérifiez que les conditions requises suivantes sont respectées.

Exigences relatives au réseau

Vous devrez autoriser l'accès à votre serveur Wickr sur les ports suivants :

- 443/TCP pour le trafic d'appels HTTPS et TCP
- 16384-19999/UDP pour le trafic d'appels UDP
- LAN uniquement - 30000/TCP pour accéder à la console d'administration KOTS

Configuration système requise

Avant l'installation, assurez-vous que vous disposez d'une machine virtuelle ou d'une machine physique exécutant un système d'exploitation Linux avec les ressources minimales disponibles suivantes :

- 8 Cœurs de CPU
- 12 gigaoctets (Go) de RAM
- 100 gigaoctets (Go) de stockage sur disque sur la partition/(root)

Le cluster intégré Wickr Enterprise a été testé sur les systèmes d'exploitation Linux suivants, mais d'autres options de système d'exploitation basées sur Linux peuvent également convenir :

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

Installation du cluster intégré Wickr Enterprise (standard)

Une fois que vous avez les instructions de téléchargement, téléchargez le bundle Wickr Enterprise sur la machine de destination et décompressez-le.

```
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
```

Vous devriez maintenant avoir deux fichiers, `wickr-enterprise-ha` et`license.yaml`. Le `wickr-enterprise-ha` fichier est un fichier binaire qui inclut toutes les pièces nécessaires à l'installation du cluster intégré, tandis que `license.yaml` c'est votre licence Wickr qui sera utilisée pour valider votre installation.

Une installation de base peut être effectuée à ce stade en exécutant le `wickr-enterprise-ha` fichier :

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

Lorsque le processus d'installation commence, vous êtes invité à entrer un mot de passe de console d'administration. Entrez un mot de passe sécurisé et assurez-vous de l'enregistrer car vous en aurez besoin pour accéder à la console d'administration KOTS pour continuer à configurer votre installation.

Une fois l'installation terminée, le résultat ressemble à ce qui suit :

```
   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
```

Après l'installation standard, accédez à l'URL de la console d'administration KOTS fournie dans la sortie à l'aide d'un navigateur Web. Dans cet exemple, l'URL esthttp://192.168.1.100:30000. Cependant, votre URL sera différente en fonction de votre configuration réseau.

Configuration de console d'administration KOTS

La console d'administration KOTS utilise initialement un certificat auto-signé, que vous devrez autoriser à titre d'exception dans votre navigateur. Une fois que vous avez accepté cette exception, vous êtes accueilli par l'assistant de configuration de la console d'administration KOTS. Cet assistant vous guide à travers les étapes de configuration supplémentaires pour configurer le comportement de la console d'administration KOTS, y compris la possibilité d'ajouter un certificat personnalisé si nécessaire.

Une fois la configuration initiale de la console d'administration KOTS terminée, vous êtes invité à saisir le mot de passe de la console d'administration que vous avez créé lors du processus d'installation. Lors de votre première connexion, vous devez configurer le cluster.

Choisissez Continuer pour accéder à la console d'administration KOTS pour Wickr.

> **Note**
>
> Les installations multi-nœuds sont actuellement en version bêta et Wickr ne les prend pas en charge.

Une fois dans la console d'administration KOTS, configurez votre installation en fonction de vos besoins. Lorsque vous utilisez l'offre de cluster intégré, certains paramètres de configuration clés doivent être définis pour garantir le bon fonctionnement de votre installation Wickr Enterprise.

- Nom d'hôte - Il s'agit du nom d'hôte que vous utilisez lorsque vous communiquez avec l'installation de Wickr. Assurez-vous de créer des enregistrements DNS appropriés pour ce domaine afin qu'ils pointent vers votre installation Wickr Enterprise.
- Sous Options avancées, cochez l'option [] Configurer le contrôleur d'entrée pour exposer un bloc de configuration permettant de configurer Kubernetes Ingress. Dans le bloc de configuration Ingress, sélectionnez Single Node Embedded Cluster, puis entrez l'adresse IP « publique » associée à votre serveur Wickr dans la zone de texte intitulée Loadbalancer External IP (Only). IPv4

  Si vous ne savez pas quelle est cette adresse IP, vous pouvez exécuter la commande suivante depuis la ligne de commande du serveur Wickr pour déterminer cette valeur : `ip route get 1.1.1.1|awk '{print $7}'`
- Sous Options avancées, cochez l'option Activer le mode à faibles ressources.
- Sous appel, assurez-vous que Require Calling Nodes est désactivé.
- Si vous souhaitez une solution tout-en-un qui n'utilise pas de base de données externe ou de stockage compatible S3 pour le partage de fichiers, sélectionnez les options internes pour les paramètres suivants :
  - Base de données
  - Emplacement de stockage S3

    L'emplacement de stockage interne S3 fournit des options supplémentaires pour configurer la capacité de stockage. Il est recommandé de commencer modestement et de développer le cas échéant, car il n'est pas possible de réduire la taille après le provisionnement.

Après avoir configuré toutes les fonctionnalités nécessaires, faites défiler l'écran de configuration vers le bas et choisissez Enregistrer la configuration. Cela lancera certaines vérifications d'hôte avant

le vol. Une fois les vérifications préalables au vol terminées, choisissez Deploy pour commencer l'installation de Wickr Enterprise.

Vous êtes maintenant prêt à commencer à configurer votre installation Wickr Enterprise. Pour plus d'informations sur la configuration de Wickr Enterprise, consultez Qu'est-ce que Wickr Enterprise ? .

Exigences d'installation communes supplémentaires

Installations de noms d'hôtes IP

Si votre installation nécessite un nom d'hôte basé sur l'adresse IP, des options de configuration supplémentaires sont disponibles. Ces instructions sont spécifiques aux noms d'hôtes basés sur IP, et il est recommandé de suivre les autres instructions pour la configuration de base répertoriées ci-dessus.

Dans le panneau d'administration KOTS, effectuez les étapes suivantes.

1. Définissez le nom d'hôte sur l'adresse IP que vous utiliserez.
2. Sous Certificats, sélectionnez Charger un certificat. Générez ensuite un certificat auto-signé en suivant les instructions relatives à un certificat basé sur IP. Pour plus d'informations, consultez Génération d'un certificat auto-signé.
3. Téléchargez le `.crt` fichier pour le certificat et le `.key` fichier pour la clé privée
4. Pour la chaîne de certificats, chargez à nouveau le `.crt` fichier.
5. Cochez la case Définir un certificat épinglé.
6. Téléchargez le `.crt` pour le certificat épinglé.
7. Sous Appels, décochez les cases Découvrir automatiquement les adresses IP publiques du serveur et Utiliser l'adresse IP principale de l'hôte pour le trafic d'appels.
8. Sous Calling, saisissez l'adresse IP du nom d'hôte dans la zone de texte Hostname Override.
9. Sous Options avancées, cochez la case Configurer le contrôleur d'entrée. Une nouvelle section de configuration intitulée Ingress apparaît ci-dessous.
10. Sous Ingress, sélectionnez Single Node Embedded Cluster.
11. Sous Ingress, entrez l'adresse IP de l'interface « publique » sur le serveur Wickr. Cela peut être différent de l'adresse IP utilisée comme nom d'hôte. Pour plus d'informations sur cette valeur, reportez-vous aux étapes de configuration de base.
12. Sous Ingress, cochez Utiliser un nom d'hôte générique.

SELinux Mode d'application

Si vous avez besoin de l'utiliser SELinux en mode d'application, modifiez le répertoire de données par défaut utilisé pour installer le cluster intégré. Il est recommandé de l'utiliser `/opt` car il a été testé pour fonctionner avec la plupart des SELinux politiques de ce cas d'utilisation.

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

Les vérifications préalables à l'installation par défaut des clusters intégrés répliqués tenteront de valider le mode permissif et échoueront s' SELinux ils sont en mode Enforcing. SELinux Pour contourner cela, il est nécessaire d'utiliser l'argument de ligne de `--ignore-host-preflights` commande. Lorsque vous utilisez l'option de ligne de commande, une invite similaire à celle ci-dessous s'affiche. Entrez Oui lorsque vous y êtes invité.

```
#  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 installations

L'option d'installation du cluster intégré pour Wickr Enterprise prend en charge les installations airgapped. Une configuration et des activations supplémentaires sont requises pour votre licence. Contactez le support si vous souhaitez utiliser le cluster intégré Wickr Enterprise dans un environnement airgapped.

Lors de l'installation d'un airgap, les instructions de téléchargement diffèrent de la méthode d'installation standard. Ils devraient ressembler à ce qui suit :

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

Téléchargez le bundle sur une machine ayant accès à Internet, puis transférez-le vers votre environnement airgapped en utilisant votre méthode de transport de données préférée. Une fois le bundle transféré, extrayez-le comme vous le feriez avec n'importe quel bundle d'installation standard. Un troisième fichier `wickr-enterprise-ha.airgap` contenant toutes les images du service d'application Wickr Enterprise associées sera inclus.

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

Lors de l'installation, il est nécessaire de définir l'argument de ligne de `--airgap-bundle` commande après l'extraction ; dans le cas contraire, le processus suit la procédure d'installation standard.

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

Mise à jour d'un cluster intégré AirGapped

Pour mettre à jour un cluster AirGapped intégré, effectuez les étapes suivantes.

1. Téléchargez le nouveau package de cluster intégré depuis Replicated et transférez-le vers la machine hôte en utilisant vos méthodes de transfert de données standard pour votre environnement airgapped. Une fois le nouveau bundle installé sur la machine hôte, extrayez l'archive tar :

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

2. Exécutez la mise à jour en utilisant le nouveau bundle binaire et airgap :

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

3. Démarrez la console d'administration KOTS et connectez-vous à l'URL fournie en utilisant vos méthodes standard d'accès à la console d'administration KOTS

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

4. Une fois connecté à la console d'administration KOTS, recherchez la dernière mise à jour disponible sur la gauche sous Version, puis appuyez sur le bouton Accéder à l'historique des versions.
5. Choisissez Déployer pour la nouvelle version sous Mises à jour disponibles. Parcourez les écrans :

    1. Modifiez les options de configuration, faites défiler l'écran vers le bas, puis choisissez Next.
    2. Vérifiez qu'aucune vérification préalable au vol n'a échoué, choisissez Next : Confirm and deploy.
    3. Choisissez Déployer.

Remarques supplémentaires sur le cluster intégré Wickr Enterprise

- NAMESPACE : Contrairement à la plupart des installations de Wickr Enterprise, l'installation de cluster intégré installe les actifs Wickr dans l'espace de noms kotsadm dans kubernetes et non dans Wickr. Modifiez tous les scripts ou commandes que vous avez enregistrés et qui sont utilisés `-n wickr` à la place par kubectl, helm ou tout autre utilitaire. `-n kotsadm`
- Interaction avec le cluster Kubernetes : depuis la machine hôte, utilisez le `./wickr-enterprise-ha` binaire pour créer un shell avec des variables appropriées définies pour interagir avec l'installation de Kubernetes en l'exécutant. ./wickr-enterprise-ha shell Cela fournira l'utilitaire kubectl dans le PATH du shell et définira la configuration kube appropriée pour l'installation locale.

Historique du document

Le tableau suivant décrit les versions de documentation du guide d'installation automatisée de Wickr Enterprise.

Modification	Description	Date
Options de déploiement automatique	Des options de déploiement automatique ont été ajoutées. Pour plus d'informations, consultez Installation de Wickr Enterprise.	23 février 2024
Ports à autoriser	Le port TCP/8443 a été ajouté à la liste des ports autorisés . Pour plus d'informations, consultez la section Exigences .	12 février 2024
Détruire les ressources et les ports à autoriser	Des instructions sur la façon de détruire les ressources ont été ajoutées. Pour plus d'informations, consultez la section Détruire des ressources. De plus, des ports ont été ajoutés à la liste des ports autorisés. Pour plus d'informations, consultez la section Exigences.	17 août 2023
Première version	Publication initiale du guide d'installation automatisée de Wickr Enterprise	4 août 2023

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.