aws

Guía de instalación automatizada

Wickr Enterprise

Wickr Enterprise: Guía de instalación automatizada

Table of Contents

¿Qué es Wickr Enterprise?

Wickr Enterprise es un servicio end-to-end cifrado y autohospedado que ayuda a las organizaciones y agencias gubernamentales a comunicarse de forma segura a través one-to-one de mensajes grupales, llamadas de voz y video, intercambio de archivos y uso compartido de pantalla. Los clientes pueden utilizar Wickr Enterprise para superar las obligaciones de retención de datos asociadas a las aplicaciones de mensajería del consumidor y a facilitar la colaboración de forma segura. Los controles administrativos y de seguridad avanzados ayudan a las organizaciones a cumplir los requisitos legales y reglamentarios y a crear soluciones personalizadas para los desafíos de seguridad de datos.

La información se puede registrar en un almacén de datos privado controlado por el cliente con fines de retención y auditoría. Los Clientes tienen un control administrativo exhaustivo sobre los datos, que incluye la configuración de permisos, la configuración de opciones de mensajería efímera y la definición de grupos de seguridad. Los administradores también pueden automatizar los flujos de trabajo de forma segura con los bots de Wickr. Wickr Enterprise se integra con servicios adicionales como Active Directory inicio de sesión único (SSO) y con OpenID Connect (OIDC). Para empezar a configurar Wickr Enterprise, consulte Cómo empezar a usar Wickr Enterprise.

> **Note**
>
> Si aún no tiene el paquete de implementación de Wickr Enterprise, consulte la sección Contactar con nosotros para consultas comerciales.

Introducción a Wickr Enterprise

Temas

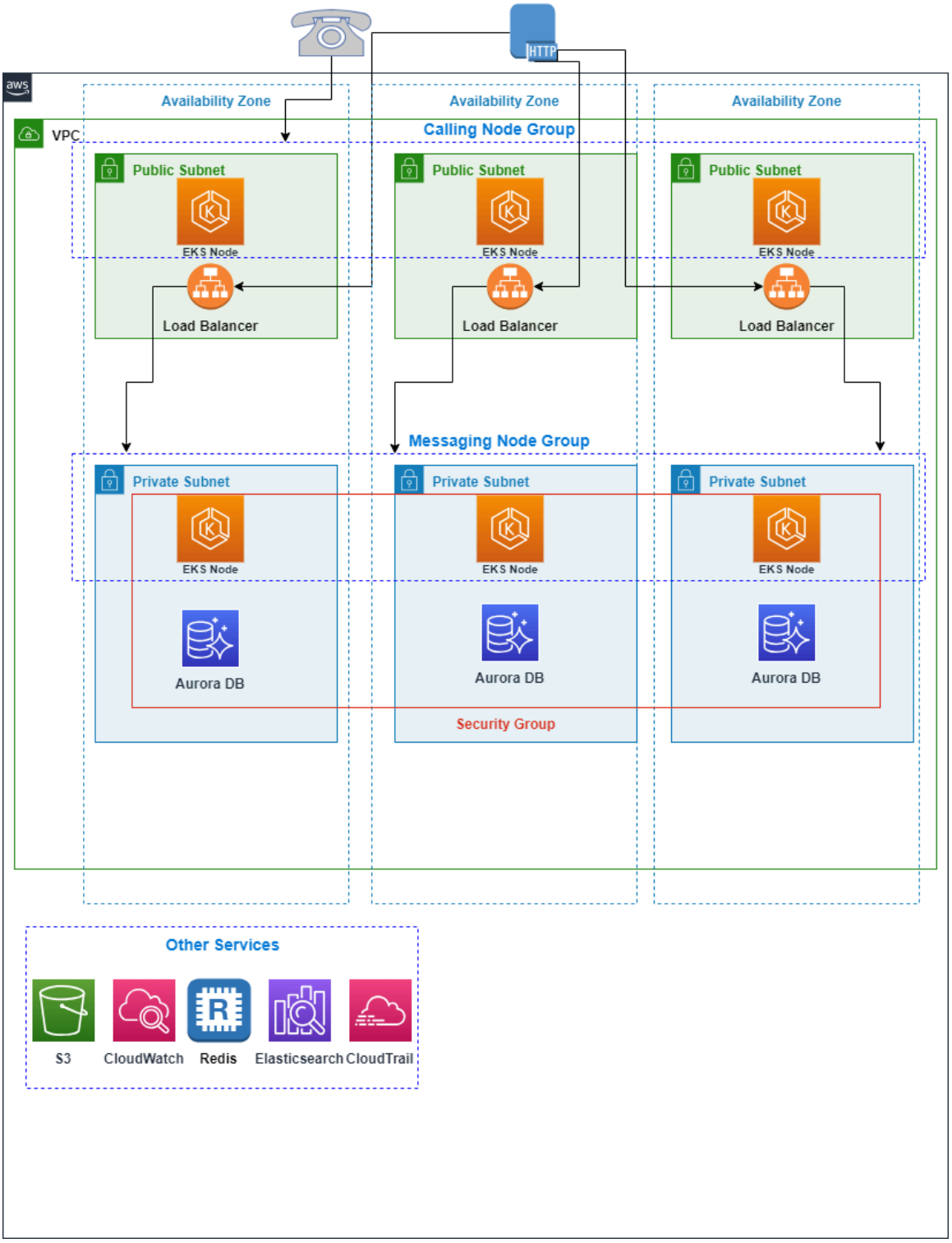
Requisitos

Antes de comenzar, compruebe que se cumplen los siguientes requisitos.

- Descargar Node.js 16+
- AWS CLI configurado con las credenciales de su cuenta.

  Se obtendrán de su archivo de configuración de `~/.aws/config` o utilizando las variables de entorno de `AWS_`.
- Instale kubectl. Para obtener más información, consulta Instalar o actualizar kubectl en la Guía de Amazon. EKSUser
- Instale kots CLI. Para obtener más información, consulte Instalación de los nudos CLI.
- Lista de puertos permitidos: 443/TCP para el tráfico de llamadas HTTPS y TCP; 16384-19999/UDP para el tráfico de llamadas UDP; TCP/8443

Arquitectura



Cómo instalar las dependencias

Puede añadir todas las dependencias al paquete predeterminado con el siguiente comando:

```
npm install
```

Configuración

AWS Cloud Development Kit (AWS CDK) usa valores de contexto para controlar la configuración de la aplicación. Wickr Enterprise utiliza los valores de contexto de CDK para controlar ajustes como el nombre de dominio de la instalación de Wickr Enterprise o el número de días que se retienen las copias de seguridad de RDS. Para más información, consulte Contexto del tiempo de ejecución en la Guía para desarrolladores de AWS Cloud Development Kit (AWS CDK) .

Hay varias formas de establecer los valores de contexto, pero le recomendamos que edite los valores en `cdk.context.json` para que se adapten a su caso de uso concreto. Solo los valores de contexto que comienzan por `wickr/` están relacionados con la implementación de Wickr Enterprise, el resto son valores de contexto de CDK específico. Para mantener la misma configuración la próxima vez que realice una actualización a través del CDK, guarde este archivo.

Como mínimo, debe establecer `wickr/licensePathwickr/domainName`, y `wickr/acm:certificateArn` o `wickr/route53:hostedZoneId` y`wickr/route53:hostedZoneName`.

Con una zona alojada pública

Si tiene una zona alojada pública de Route 53 Cuenta de AWS, le recomendamos que utilice los siguientes ajustes para configurar el contexto de CDK:

- `wickr/domainName`: el nombre de dominio que se utilizará para esta implementación de Wickr Enterprise. Si utiliza una zona alojada pública de Route 53, los registros DNS y los certificados ACM para este nombre de dominio se crearán automáticamente.
- `wickr/route53:hostedZoneName`: nombre de la zona alojada de Route 53 en la que crear registros DNS.
- `wickr/route53:hostedZoneId`: el ID de zona alojada de Route 53 en la que crear registros DNS.

Este método crea un certificado de ACM en su nombre, junto con los registros de DNS que apuntan su nombre de dominio al equilibrador de carga antes de su implementación de Wickr Enterprise.

Sin una zona alojada pública

Si no tiene una zona alojada pública de Route 53 en su cuenta, debe crear un certificado de ACM manualmente e importarlo al CDK con el valor de contexto `wickr/acm:certificateArn`.

- `wickr/domainName`: el nombre de dominio que se utilizará para esta implementación de Wickr Enterprise. Si utiliza una zona alojada pública de Route 53, los registros DNS y los certificados ACM para este nombre de dominio se crearán automáticamente.
- `wickr/acm:certificateArn`: el ARN de un certificado ACM para usar en el equilibrador de carga. Este valor debe proporcionarse si una zona alojada pública de Route 53 no está disponible en su cuenta.

Importación de un certificado a AMC

Puede importar un certificado obtenido externamente con el siguiente 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
```

El resultado será el ARN del certificado, que debe usarse para el valor de la configuración de contexto `wickr/acm:certificateArn`. Es importante que el certificado cargado sea válido para `wickr/domainName` o las conexiones HTTPS no se podrán validar. Para obtener más información, consulte Importación de certificado en la Guía del usuario de AWS Certificate Manager .

Cómo crear registros DNS

Como no hay ninguna zona alojada pública disponible, los registros de DNS deben crearse manualmente una vez finalizada la implementación para que apunten al equilibrador de carga delante de la implementación de Wickr Enterprise.

Implementación en una VPC existente

Si necesita usar una VPC existente, puede usar una. Sin embargo, la VPC debe configurarse para cumplir con las especificaciones necesarias para EKS. Para obtener más información, consulte Ver los requisitos de red de Amazon EKS para la VPC y las subredes en la Guía del usuario de Amazon EKS y asegúrese de que la VPC que se va a utilizar cumpla estos requisitos.

Además, se recomienda encarecidamente asegurarse de tener puntos de enlace de VPC para los siguientes servicios:

- CLOUDWATCH
- CLOUDWATCH_LOGS
- EC2
- EC2_MENSAJES
- ECR
- ECR_DOCKER
- ELASTIC_LOAD_BALANCING
- KMS
- ADMINISTRADOR DE SECRETOS
- SSM
- SSM_MESSAGES

Para implementar recursos en una VPC existente, defina los siguientes valores de contexto:

- `wickr/vpc:id`: el ID de VPC en el que se van a implementar los recursos (por ejemplo, `vpc-412beef`).
- `wickr/vpc:cidr`- El IPv4 CIDR de la VPC (`172.16.0.0/16`por ejemplo).
- `wickr/vpc:publicSubnetIds`: una lista separada por comas de subredes públicas en la VPC. El equilibrador de carga de aplicación y los nodos de trabajo de EKS de llamada se implementarán en estas subredes (por ejemplo, `subnet-6ce9941,subnet-1785141,subnet-2e7dc10`).
- `wickr/vpc:privateSubnetIds`: una lista separada por comas de subredes privadas en la VPC. Los nodos de trabajo y el servidor bastión de EKS se implementarán en estas subredes (por ejemplo, `subnet-f448ea8,subnet-3eb0da4,subnet-ad800b5`).
- `wickr/vpc:isolatedSubnetIds`: una lista separada por comas de subredes aisladas en la VPC. La base de datos de RDS se implementará en estas subredes (por ejemplo, `subnet-d1273a2,subnet-33504ae,subnet-0bc83ac`).
- `wickr/vpc:availabilityZones`: una lista de zonas de disponibilidad separadas por comas para las subredes de la VPC (por ejemplo, `us-east-1a,us-east-1b,us-east-1c`).

Para obtener más información sobre los puntos de enlace de VPC de interfaz, consulte Acceder a un AWS servicio mediante un punto de enlace de VPC de interfaz.

Otra configuración

Para obtener más información, consulte Valores de contexto.

Arranque

Si es la primera vez que usa la CDK en esta región Cuenta de AWS y en particular, primero debe iniciar la cuenta para empezar a usar la CDK.

```
npx cdk bootstrap
```

Implementación

Este proceso tardará unos 45 minutos.

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

Cuando se haya completado, se habrá creado la infraestructura y podrá empezar a instalar Wickr Enterprise.

Cómo crear registros DNS

Este paso no es necesario si utilizó una zona alojada pública al configurar el CDK.

La salida del proceso de implementación incluirá un valor `WickrAlb.AlbDnsName`, que es el nombre de DNS del equilibrador de carga. La salida tendrá este aspecto:

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

En este caso, el nombre del DNS es `Wickr-Alb-1Q5IBPJR4ZVZR-409483305.us-west-2.elb.amazonaws.com`. Ese es el valor que debe utilizarse al crear un registro CNAME o A/AAAA (ALIAS) para su nombre de dominio.

Si no dispone de la salida de la implementación, ejecute el siguiente comando para ver el nombre de DNS del equilibrador de carga:

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

Cómo generar la configuración de KOTS

> **Warning**
>
> Este archivo contiene información confidencial sobre su instalación. No lo comparta ni lo guarde públicamente.

El instalador de Wickr Enterprise requiere una serie de valores de configuración sobre la infraestructura para poder instalarlo correctamente. Puede usar un script auxiliar para generar los valores de configuración.

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

Si importó un certificado externo a ACM en el primer paso, transfiera la marca `--ca-file` a este script, por ejemplo:

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

Si recibe un error que indica que la pila no existe, establezca la variable de entorno `AWS_REGION` (`export AWS_REGION=us-west-2`) en la región seleccionada e inténtelo de nuevo. O bien, si estableces el valor de contexto`wickr/stackSuffix`, pasa el sufijo con la `--stack-suffix` bandera.

Conexión al clúster de Kubernetes

Solo se puede acceder a la API de Amazon EKS a través de un host bastión que se crea como parte de la implementación. Como resultado, todos los comandos de `kubectl` deben ejecutarse en el propio host bastión o enviarse mediante un proxy a través del host bastión.

Conexiones mediante proxy a través del bastión

La primera vez que se conecte al clúster, debe actualizar su archivo kubeconfig local mediante el comando `aws eks update-kubeconfig` y, a continuación, ajustar el `proxy-url` en su configuración. A continuación, cada vez que quiera conectarse al clúster, iniciará una sesión de SSM con el host bastión para redirigirla al proxy y poder acceder a la API.

Configuración única

Hay un valor de salida en la `WickrEks` CloudFormation pila con un nombre que comienza por`WickrEnterpriseConfigCommand`. El valor contiene el comando completo necesario para generar la configuración de kubectl para el clúster. Puede verse este resultado con el siguiente comando:

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

Esto debería generar un comando que comience por `aws eks update-kubeconfig`. Ejecute este comando.

A continuación, se debe modificar la configuración de Kubernetes para que las solicitudes se envíen por proxy a través del host bastión. Se puede hacer con los siguientes comandos:

```
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 funcionó correctamente, verá un resultado como `'Property "clusters.arn:aws:eks:us-west-2:012345678912:cluster/WickrEnterprise5B8BF472-1234a41c4ec48b7b615c6789d93dcce.proxy-url" set.'`

Avanzar hacia el bastión

Para conectarse al clúster de Amazon EKS, debe iniciar una sesión de SSM para reenviar las solicitudes al proxy que se ejecuta en su host bastión. El comando para hacerlo se proporciona como resultado `BastionSSMProxyEKSCommand` en la pila `WickrEks`. Ejecute el siguiente comando para ver el valor del resultado:

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

El comando que se genere empezará por `aws ssm start-session`. Ejecute este comando para iniciar un proxy local que se ejecute en el puerto 8888 a través del cual podrá conectarse al clúster de Amazon EKS. Si el reenvío de puertos funcionó correctamente, el resultado debería decir «Esperando conexiones...». Mantenga este proceso en ejecución todo el tiempo que necesite acceder al clúster de Amazon EKS.

Si todo está configurado correctamente, podrá ejecutar `kubectl get nodes` en otro terminal para enumerar los nodos de trabajo del clúster de 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
```

Instalación de Wickr Enterprise

Una vez realizada la conexión al clúster de Kubernetes, puede empezar a instalar Wickr Enterprise usando el complemento `kubectl kots`. Necesitará su archivo de licencia de KOTS (un archivo `.yaml` proporcionado por Wickr) y su archivo Config Values (de valores de configuración), que se guardaron en el archivo `wickr-config.json` de la sección Generación de la configuración de KOTS. Para obtener más información sobre cómo se genera la configuración de KOTS, consulte Generación de la configuración de KOTS.

Instalación manual de Wickr Enterprise

El siguiente comando iniciará la instalación de Wickr Enterprise:

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

Se le pedirá que indique una contraseña para la consola administración de KOTS. Consérvela: la necesitará para actualizar o cambiar la configuración de su instalación de Wickr Enterprise en el futuro.

Cuando se complete la instalación, `kubectl kots` abrirá un puerto local (normalmente `http://localhost:8080`), que da acceso a la consola de administración de KOTS. Puede cambiar o supervisar el estado de su instalación de Wickr Enterprise en este sitio. También puede comenzar a configurar Wickr accediendo al nombre de dominio que configuró para su instalación en su navegador.

Instalación de Wickr Enterprise con Lambda

Durante la implementación de la CDK, se crea e invoca una Lambda para completar automáticamente la instalación de Wickr Enterprise en su nombre. Para invocarla manualmente, abra la AWS consola y busque la función `WickrLambda-func*` lambda; en la pestaña de prueba, seleccione`test`, la entrada es irrelevante.

Después de la instalación

Hay dos consolas web disponibles para gestionar su instalación de Wickr Enterprise: la consola de administración de KOTS y la consola de administración de Wickr.

> **Note**
>
> Aplique los cambios necesarios para reflejar las políticas de backup y registro de su organización (configuración de Amazon S3, registros de acceso de Elastic Load Balancing, registros de flujo de Amazon Virtual Private Cloud).

Consola de administración de KOTS

Esta interfaz se usa para administrar la versión implementada de Wickr Enterprise. Puede ver el estado de la instalación, modificar las configuraciones o realizar actualizaciones. Solo se puede acceder a la consola de administración de KOTS a través de un puerto de reenvío de Kubernetes, que se puede abrir con el comando siguiente:

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

> **Note**
>
> Primero, configure su conexión con el bastión tal y como se describe en la sección de reenvío de puertos al bastión. Para obtener más información sobre el reenvío de puertos al bastión, consulte Cómo conectar mediante el bastión.

Cuando el reenvío de puertos se haya configurado correctamente, el comando anterior mostrará el resultado siguiente:

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

Utilice la URL proporcionada para acceder a la consola de administración de KOTS. La contraseña para iniciar sesión es la que eligió cuando se ejecutó `kubectl kots install` durante la

instalación. Si necesita restablecer la contraseña, consulte Cómo restablecer la contraseña de la consola de administración de KOTS.

Consola de administración de Wickr

Esta interfaz se utiliza para configurar la instalación de Wickr Enterprise y establecer redes, usuarios y federaciones. Se puede acceder a ella a través de HTTPS con el nombre de DNS que configuró para que apunte a su equilibrador de carga. Si el DNS se configuró automáticamente con una zona alojada pública, el nombre de dominio es el valor del contexto de `wickr/domainName`.

El nombre de usuario y la contraseña predeterminados son `admin` y `Password123` respectivamente. Se le pedirá que cambie esta contraseña la primera vez que inicie sesión.

Valores de contexto

Los valores de contexto son pares clave-valor que pueden asociarse a una aplicación, pila o constructo. Se pueden proporcionar a la aplicación desde un archivo (normalmente, `cdk.json` o `cdk.context.json` en el directorio del proyecto) o en la línea de comandos. CDK usa valores de contexto para controlar la configuración de la aplicación. Wickr Enterprise utiliza los valores de contexto de CDK para controlar ajustes como el nombre de dominio de la instalación de Wickr Enterprise o el número de días que se retienen las copias de seguridad de RDS.

Hay varias formas de establecer los valores de contexto, pero le recomendamos que edite los valores en `cdk.context.json` para que se adapten a su caso de uso concreto. Solo los valores de contexto que comienzan por `wickr/` están relacionados con la implementación de Wickr Enterprise.

Name	Descripción	Predeterminado
`wickr/licensePath`	La ruta a su licencia de KOTS (un archivo `.yaml` proporcionado por Wickr).	null
`wickr/domainName`	El nombre de dominio que se utilizará para esta implementación de Wickr Enterprise. Si utiliza una zona alojada pública de Route 53, los registros DNS y los certificados ACM para este nombre de dominio se crearán automáticamente.	null
`wickr/route53:hostedZoneId`	El ID de zona alojada de Route 53 en la que crear registros DNS.	null
`wickr/route53:hostedZoneName`	Nombre de la zona alojada de Route 53 en la que crear registros DNS.	null

Name	Descripción	Predeterminado
`wickr/acm:certificateArn`	ARN de un certificado ACM para usar en el equilibrador de carga. Este valor debe proporcionarse si una zona alojada pública de Route 53 no está disponible en su cuenta.	null
`wickr/caPath`	Ruta del certificado, solo necesaria cuando se utilizan certificados autofirmados.	null
`wickr/vpc:id`	Es el ID de la VPC en la que se deben implementar los recursos. Solo se requiere cuando se implementa en una VPC existente. Si no se establece, se creará una nueva VPC.	null
`wickr/vpc:cidr`	IPv4 CIDR para asociarlo a la VPC creada. Si se implementa en una VPC existente, establezca la configuración en el CIDR de la VPC existente.	172.16.0.0/16
`wickr/vpc:availabilityZones`	Lista separada por comas de zonas de disponibilidad. Solo se requiere cuando se implementa en una VPC existente.	null

Name	Descripción	Predeterminado
`wickr/vpc:publicSubnetIds`	Lista de subredes públicas separadas por comas. IDs Solo se requiere cuando se implementa en una VPC existente.	null
`wickr/vpc:privateSubnetIds`	Lista de subredes privadas separadas por comas. IDs Solo se requiere cuando se implementa en una VPC existente.	null
`wickr/vpc:isolatedSubnetIds`	Lista separada por comas de subredes aisladas para la base de datos de RDS IDs . Solo se requiere cuando se implementa en una VPC existente.	null
`wickr/rds:deletionProtection`	Habilite la protección contra la eliminación en las instancias de RDS.	true
`wickr/rds:removalPolicy`	Política de eliminación para las instancias de RDS: “instantánea”, “destruir” o “retener”.	instantánea
`wickr/rds:readerCount`	Número de instancias de lectura que se van a crear en el clúster de RDS.	1
`wickr/rds:instanceType`	Tipo de instancia que se utilizará en las instancias de RDS.	r6g.xlarge

Name	Descripción	Predeterminado
`wickr/rds:backupRetentionDays`	Número de días que se conservan las copias de seguridad.	7
`wickr/eks:namespace`	Espacio de nombres predeterminado para los servicios de Wickr en EKS.	wickr
`wickr/eks:defaultCapacity`	Número de nodos de trabajo de EKS para la infraestructura de mensajería.	3
`wickr/eks:defaultCapacityCalling`	Número de nodos de trabajo de EKS para la infraestructura de llamadas.	2
`wickr/eks:instanceTypes`	Lista separada por comas de los tipos de instancias que se utilizan en los nodos de trabajo de EKS para mensajería.	m5.xlarge
`wickr/eks:instanceTypesCalling`	Lista separada por comas de los tipos de instancias que se utilizan para llamar a los nodos de trabajo de EKS.	c5n.large
`wickr/eks:enableAutoscaler`	Activa la funcionalidad de escalado automático de clústeres para EKS.	true
`wickr/s3:expireAfterDays`	Número de días después de los cuales las cargas de archivos se eliminarán del bucket de S3.	1095

Name	Descripción	Predeterminado
`wickr/eks:clusterVersion`	Versiones de clúster, incluidas la versión de Kubernetes, la versión de KubectLayer, la versión de AlbController, la versión y más. nodeGroup Release	1.27
`wickr/stackSuffix`	Un sufijo para aplicar a los nombres de las CloudFormation pilas.	"
`wickr/autoDeployWickr`	Implemente automáticamente la aplicación Wickr con lambda.	true

Destrucción de recursos

Para eliminar todo lo creado por esta AWS CDK aplicación, debe eliminar la `WickrRds` pila antes que todas las demás pilas.

Para que los recursos de Amazon RDS se eliminen correctamente, debe estar deshabilitada la protección contra la eliminación y estar configurada la política de eliminación en una de las dos opciones `snapshot` o `destroy`. Si esta no es la configuración actual, modifique los valores `wickr/rds:deletionProtection` y `wickr/rds:removalPolicy` en su contexto de AWS CDK y vuelva a implementar la pila de Amazon RDS ejecutando `npx cdk deploy -e WickrRds`.

Una vez que la política de eliminación y la protección contra eliminaciones estén correctamente establecidas, ejecute la operación `cdk destroy` para la pila `WickrRds`:

```
npx cdk destroy WickrRds
```

Cuando la `WickrRds` pila haya terminado de destruirse, las CloudFormation acumulaciones restantes se pueden destruir con el siguiente comando:

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

Resolución de problemas

Eliminación del espacio de nombres de Wickr

Si necesita eliminar el espacio de nombres de `wickr` para volver a empezar, es importante que primero haga una copia de seguridad de las cuentas de servicio que CDK haya creado en ese espacio de nombres. Estas cuentas de servicio permiten que los servicios de Wickr se comuniquen a AWS APIs través de las funciones de IAM. Sin ellas, tareas como cargar archivos mediante Amazon Simple Storage Service (Amazon S3) dejarían de funcionar.

Utilice el siguiente comando para hacer una copia de seguridad de las cuentas de servicio y eliminar y volver a crear el espacio de nombres de `wickr` y las cuentas de servicio correspondientes:

```
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
```

Restablecimiento de la contraseña de la consola de KOTS Admin

Puede restablecer la contraseña de la consola de KOTS Admin con el siguiente comando:

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

Cuando cambies esta contraseña, es posible que también desees actualizar el secreto de `wickr/kots` Secrets Manager, aunque por lo general no lo volverá a utilizar ninguna automatización.

Problemas para conectarse al clúster EKS con Bastion

Si la conexión al clúster EKS a través del bastión parece lenta o se agota el tiempo de espera de vez en cuando, es posible que aparezca el siguiente error al ejecutar `kubectl` los comandos:

net/http: solicitud cancelada mientras se esperaba la conexión (se ha superado el tiempo de espera de Client.Timeout mientras se esperaban los encabezados)

Este problema suele solucionarse iniciando sesión en el host del bastión mediante SSM (consulta el que aparece en la pila) y reiniciando `BastionSSMCommand` el servicio: WickrEks `tinyproxy`

```
sudo systemctl restart tinyproxy
```

Instalación personalizada

En la sección Instalación personalizada, aprenderás cómo instalar Wickr Enterprise.

Temas

Requisitos

Antes de empezar a instalar Wickr Enterprise, compruebe que se cumplen los siguientes requisitos.

Requisitos de hardware

Wickr Enterprise requiere un clúster de Kubernetes para funcionar. Es posible operar en un solo nodo con el modo de bajo recurso activado, pero no se recomienda para un uso general de producción. En una implementación de producción, recomendamos un mínimo de tres nodos de trabajo de mensajería, así como un mínimo de dos nodos de trabajo de llamada.

Un nodo trabajador debe tener las siguientes especificaciones mínimas.

- 2 a 4 núcleos de CPU
- 8 GB de RAM
- 200 GB de espacio en disco

Requisitos mínimos de hardware

Un clúster de un solo nodo de trabajo que se ejecute en modo de bajo recurso requiere un mínimo de 3000 M de CPU y 5846 MB de RAM. Esto no incluye los módulos del sistema kube.

Requisitos de recursos por pod

Nombre de pod	Propietario	CPU	Memoria
admin-api	Wickr	100 m	256Mi
directory	Wickr	100 m	128 Mi
expirador	Wickr	100 m	128 Mi
proxy de archivos	Wickr	100 m	256Mi
oidc	Wickr	100 m	128 Mi
opensearch	Wickr	500 m	100 millas
Corville	Wickr	50 m	128 Mi
Morville-redis	Wickr	50 m	128 Mi
dispositivo de empuje	Wickr	100 m	128 Mi
rabbitmq	Wickr	50 m	256Mi
reaccionar	Wickr	100 m	64 Mi
recibos	Wickr	250 m	128 Mi
redis	Wickr	50 m	128 Mi
servidor-api	Wickr	250 m	256Mi

Nombre de pod	Propietario	CPU	Memoria
conmutador	Wickr	250 m	512 Mi
kotsadm	LOTES	50 m	50 millas
kotsadm-minio	LOTES	100 m	512 Mi
kotsadm-qlite	LOTES	200m	1 Gi
minioperadora	S3 interno	200m	256Mi
miniinquilino	S3 interno	100 m	256Mi
mysql-primary	MySQL interno	100 m	512 Mi
mysql-secondary	MySQL interno	100 m	512 Mi

Requisitos de almacenamiento

Al crear reclamaciones por volumen persistentes, Wickr Enterprise requiere una configuración predeterminada StorageClass . Cuando se despliega en un entorno aislado o local, es posible que deba configurar uno para su clúster. Una opción disponible es Longhorn. Los requisitos de espacio en disco recomendados variarán en función del uso de las opciones S3 interno y Mysql interno y de la cantidad de espacio disponible para cargar archivos.

- Almacenamiento interno de imágenes: ~60 Gi
- RabbitMQ: 24 Gi por defecto, 8 Gi en modo de bajos recursos
- Redis: 24 Gi por defecto/8 Gi en modo de bajos recursos
- OpenSearch: 24 Gi Default/8 Gi en modo de bajos recursos
- Mysql interno: 80 Gi por defecto/20 Gi en modo de bajos recursos
- S3 interno: 160 Gi por defecto /2Gi en modo de bajos recursos
- KOTS Minio: 4 Gi
- KOTS Relite: 1 GB

Tamaño mínimo de almacenamiento

- 377 Gi predeterminado con S3 interno y Mysql interno
- 111 Gi en modo de bajos recursos

Requisitos de versión de Kubernetes

Wickr Enterprise confía en los KOTS replicados. Replicated, una plataforma de distribución de software comercial, proporciona una lista de las versiones de Kubernetes compatibles actualmente. Para obtener más información, consulte Compatibilidad de versiones de Kubernetes.

Requisitos de software

Wickr Enterprise requiere un clúster de Kubernetes y un KOTS para funcionar. Consulte la documentación de KOTS para ver las versiones de OS y Kubernetes compatibles. Para obtener más información, consulta los requisitos mínimos del sistema.

Sistema anfitrión para desarrolladores

Sistema operativo: los comandos de esta documentación están diseñados para funcionar en Linux, macOS o Windows con WSL (subsistema de Windows para Linux) instalado.

Servicios internos de estado

Wickr Enterprise puede proporcionar servicios internos tanto para la base de datos MySQL como para el almacenamiento compatible con S3; sin embargo, para un uso de producción general, se recomienda proporcionar estos servicios de forma externa al clúster de Kubernetes.

- Base de datos MySQL 5.7
  - Base de datos MySQL 5.7 o MySQL 5.7 de Amazon RDS (externa)
  - Gráfico Bitnami Helm de Mysql (interno)
  - Almacenamiento de archivos
    - Proveedor de almacenamiento compatible con Amazon S3 o S3 (externo)
    - Gráfico Minio Operator Helm (interno)

Requisitos de red

Wickr Enterprise requiere un FQDN, certificados SSL y puertos TCP y UDP abiertos específicos.

- FQDN: dominio o subdominio que utilizará la implementación de Wickr Enterprise.

- Certificado SSL: un par de claves de certificado SSL firmado por una CA pública o un par de claves de certificado autofirmado. El certificado debe incluir el FQDN en el nombre común y también como una entrada de DNS de SAN. El certificado también debe habilitar la extensión ServerAuth extendedKeyUsage .
- Las instalaciones en línea necesitarán acceso de salida a los recursos replicados y de terceros. Replicated mantiene una lista de sus direcciones IP. Para obtener más información, consulte Direcciones IP replicadas. Replicated también mantiene una lista de los recursos de terceros necesarios. Para obtener más información, consulte Aperturas de firewall para instalaciones en línea.
- Las instalaciones aisladas requieren acceso a un registro de contenedores privado.

Nodos de mensajería

Los nodos de mensajería no requieren una IPV4 dirección pública y deben estar ubicados en una subred privada. El tráfico de mensajes ingresará al clúster a través de la entrada LoadBalancer o.

Nodos de llamada

Los nodos de llamada requieren una IPV4 dirección pública, por lo que deben estar en una subred pública. El contenido multimedia de las llamadas se transfiere mediante UDP de forma predeterminada. Cuando las llamadas TCP están habilitadas, el proxy TCP acepta las conexiones en el TCP 443 y las envía por proxy al servicio Orville.

- TCP: 443 Llamando al proxy TCP
- UDP: Audio/Video 16384-16484 transmisiones

Acceso a la instalación y configuración

El acceso a la consola de administración de KOTS para la instalación y la configuración se realiza a través de un reenvío de puertos de Kubernetes.

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

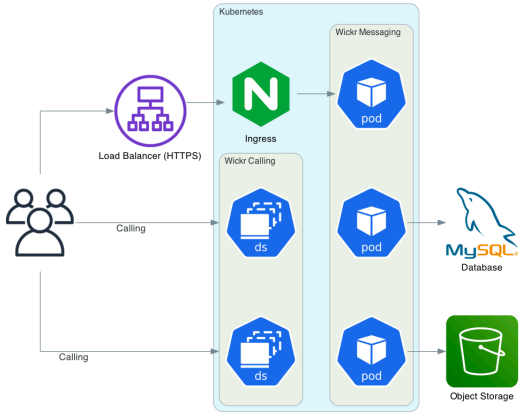
Requisitos de licencia

La instalación requerirá un archivo de licencia en formato.yaml, que se lo proporcionará Wickr Support.

Arquitectura

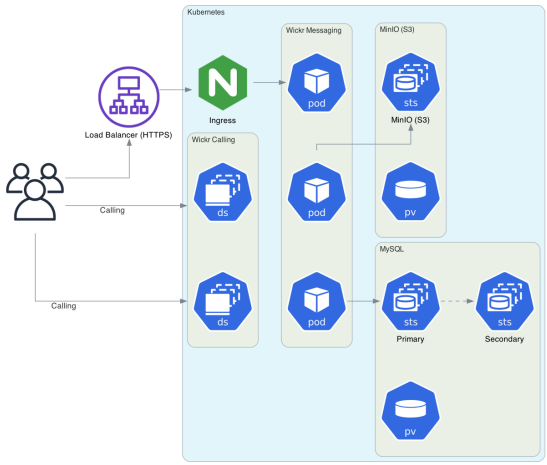
Arquitectura de producción recomendada

El siguiente diagrama muestra Wickr Enterprise configurado según lo recomendado para producción, con los servicios MySQL y Object Storage situados fuera del clúster de Kubernetes.



Arquitectura interna o de prueba

El siguiente diagrama muestra la configuración de Wickr Enterprise, utilizando los servicios internos de MYSQL y Object Storage. Si bien puede satisfacer las necesidades específicas de ciertas implementaciones, no se recomienda su uso en producción general.



Instalación

1. Instale kubectl y kots CLI.
2. Conéctese al clúster de Kubernetes.
3. Obtenga el archivo de licencia de Wickr Enterprise en Wickr Support.
4. Instala Wickr Enterprise con el siguiente comando.

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

> **Note**
>
> license.yaml representa el archivo de licencia que has proporcionado.

Tras la instalación inicial, la consola de administración de KOTS proporcionará opciones de administración y configuración a nivel de clúster.

Consola de administración de KOTS

Esta interfaz se usa para administrar la versión implementada de Wickr Enterprise. Puedes ver el estado de la instalación, modificar las configuraciones o realizar actualizaciones de Wickr Enterprise. Solo se puede acceder a la consola de administración de KOTS a través de un puerto de reenvío de Kubernetes, que se puede abrir con el comando siguiente:

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

Configuración de ingreso

Controlador de ingreso

Wickr Enterprise admite cuatro tipos de controladores de ingreso:

- LoadBalancer (Predeterminado)
  - El objeto loadbalancer puede requerir una configuración explícita en instalaciones totalmente locales, aunque a menudo lo proporcionan los proveedores de nube.
  - Implementa el servicio del controlador de entrada (ingress-nginx) con el tipo de servicio. LoadBalancer Esto requiere que el clúster de Kubernetes se ejecute en una plataforma que admita balanceadores de carga externos.
- ALB existente
  - Conecta el controlador de ingreso a un ALB existente.
  - Deberá proporcionar el ARN del grupo objetivo de Application Load Balancer existente.
- NLB existente
  - Conecta el controlador de ingreso a un NLB existente.
  - Deberá proporcionar el ARN del grupo objetivo de Network Load Balancer existente.
- NodePort
  - El controlador de entrada (ingress-nginx) se configurará para usar el tipo de NodePort servicio, que abre un puerto en todos los nodos del clúster de Kubernetes y reenvía el tráfico a la entrada. Luego, el tráfico de los clientes se puede dirigir a estos nodos a través del DNS o de algún balanceador de carga externo.
  - Puede elegir un rango de puertos entre 1 y 65535 o usar un puerto aleatorio entre 30000-32767.
- Ingreso

- Traiga su propio controlador de entrada. Esta configuración aceptará un nombre de clase de entrada que los servicios utilizarán después en sus manifiestos de entrada. Esto implica que el controlador de entrada tiene alguna conectividad externa ya configurada mediante algún otro mecanismo de equilibrio de carga.
- Actualmente, solo se admite el controlador ingress-nginx.

Nombre de host comodín

De forma predeterminada, las rutas de entrada se definirán con un valor de host de `*`. Desactive esta configuración para usar el nombre de host definido para el servidor empresarial de Wickr. Se requiere un nombre de host comodín para los nombres de host basados en IP.

Configuración de base de datos

Wickr Enterprise requiere una base de datos MySQL 8.0. Si utiliza MySQL 5.7, consulte Actualización a MySQL 8.0 para actualizar. Recomendamos usar una base de datos externa al clúster de Kubernetes, como Amazon RDS, pero también tiene la opción de implementar una base de datos MySQL interna dentro del clúster de Kubernetes como parte de la instalación.

Configuración de la base de datos externa

- Nombre de host: nombre de host o dirección IP del servidor de base de datos.
- Nombre de host del lector: nombre de host o dirección IP de un punto final de solo lectura para el servidor de base de datos (si está disponible).
- Puerto: el puerto desde el que se accederá a MySQL.
- Nombre de la base de datos: el nombre de la base de datos creada en el servidor.
- Nombre de usuario: el usuario que tiene permisos para acceder a la base de datos.
- Contraseña: la contraseña de ese usuario.
- Certificado de CA: certificado PEM para conectarse a la base de datos a través de TLS.

> **Note**
>
> Asegúrese de que su instalación de MySQL utilice el conjunto de caracteres latin1 predeterminado con la intercalación latin1_swedish_ci. Esto se puede lograr verificando que su servidor MySQL se inicie con los siguientes indicadores:

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

Configuración interna de la base de datos

El tipo de base de datos interna desplegará dos StatefulSets en su clúster para una MySQL principal y una secundaria con replicación binaria. La secundaria no recibe ningún tráfico y solo está disponible para la recuperación ante desastres y las copias de seguridad.

Tamaño de almacenamiento: tamaño (en gibibytes) de los volúmenes persistentes de los pods de bases de datos.

Aumentar el tamaño de almacenamiento de MySQL

> **Note**
>
> El tipo de volumen StorageClass que utilice debe admitir la expansión del volumen para aumentar el tamaño del almacenamiento. Para obtener más información, consulte Expansión de volumen.

Los servicios MySQL utilizados en Wickr Enterprise se implementan como StatefulSet recursos en Kubernetes. StatefulSets hacen que muchas propiedades del recurso sean inmutables, incluidas las plantillas Persistent Volume Claim. Como solución alternativa a la inmutabilidad de StatefulSets, se deben realizar las siguientes acciones para aumentar el tamaño de los volúmenes utilizados por MySQL.

1. Edite las notificaciones de volumen persistentes para y. `data-mysql-primary-0` `data-mysql-secondary-0`
    1. `kubectl -n wickr edit pvc data-mysql-primary-0.` Set `spec.resources.requests.storage`al tamaño de almacenamiento deseado.
    2. `kubectl -n wickr edit pvc data-mysql-secondary-0.` Set `spec.resources.requests.storage`al tamaño de almacenamiento deseado.
2. Elimina los existentes StatefulSets, pero abandona los Pods pasando la `--cascade=orphan` bandera.

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

3. En la interfaz de usuario de KOTS, actualiza la configuración del tamaño de almacenamiento para que coincida con el valor que estableciste en el paso 1. Guarde e implemente esta configuración.
4. Reinicie el StatefulSets para ampliar los volúmenes y volver a poner en línea los servicios MySQL.

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

Actualización a MySQL 8.0

Base de datos externa (RDS)

Para desconectar Wickr Backend, sigue estos pasos.

1. Encuentra el espacio de nombres de Ingress `kubectl get deployments --all-namespaces`

   En el siguiente ejemplo, el espacio de nombres es Wickr y las réplicas son 3.

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

2. Reduce el ingreso `kubectl scale deployment/ingress-nginx-controller --replicas=0 -n wickr`
3. Realice una instantánea para hacer una copia de seguridad de la base de datos. Para obtener más información, consulte Gestión de copias de seguridad manuales en la Guía del usuario de Amazon Relational Database Service.
4. Actualice la versión del motor a MySQL 8.0.x (MySQL 8.4 no es compatible). Para obtener más información, consulte Actualización de una versión de motor de instancias de base de datos en la Guía del usuario de Amazon Relational Database Service.

Para poner Wickr Backend en línea, reduzca el ingreso `kubectl scale deployment/ingress-nginx-controller --replicas=3 -n wickr`

Base de datos interna

Para obtener más información, consulte Backup and Restore MySQL.

Almacenamiento de archivos S3

Wickr Enterprise requiere un servicio de almacenamiento compatible con S3. Te recomendamos usar un servicio S3 externo al clúster de Kubernetes, como Amazon S3, pero también tienes la opción de implementar un servicio S3 interno dentro del clúster de Kubernetes como parte de la instalación.

Configuración externa de S3

- Nombre del depósito: el nombre del depósito de S3 en el que se almacenarán las cargas de archivos.
- Región: la AWS región del depósito de S3.
- Punto final: establece el punto final que utilizará Wickr para interactuar con la API de S3. El valor predeterminado es el punto final del servicio S3 de la región.
- Nombre de la cuenta del servicio Fileproxy: Amazon S3 únicamente. El nombre de una cuenta de servicio de Kubernetes existente que se utilizará para autenticarse en S3 mediante las funciones de IAM para las cuentas de servicio.
- Clave de acceso S3 externa: es tu clave de acceso S3 existente.
- Clave secreta de S3 externa: es su clave secreta de S3 existente.

Configuración interna de S3

El tipo S3 interno implementará por defecto 4 módulos de servidor MinIO, cada uno de los cuales contiene 4 notificaciones de volumen persistentes. La configuración predeterminada utiliza la codificación de borrado de MinIO para aumentar la tolerancia a los errores.

- Recuento de servidores S3 internos: el número de módulos de servidores MinIO que se van a crear; el valor predeterminado es 4 para una implementación tolerante a errores. Este valor se puede establecer en tan solo 1 para una development/test implementación.
- Recuento de volúmenes S3 internos: el número de volúmenes MinIO que se van a crear en cada módulo de servidor MinIO; el valor predeterminado es 4 para una implementación tolerante a errores. Este valor se puede establecer en tan solo 1 para una development/test implementación.

- Tamaño del volumen S3 interno: el tamaño en GB de los volúmenes de MinIO creados en los módulos de servidores MinIO, el tamaño predeterminado es de 10 GB.
- Una implementación interna de S3 predeterminada utilizará 4 servidores con 4 PVCs. Cada PVC es de 10 Gi, lo que proporciona un almacenamiento sin procesar de 160 Gi y un almacenamiento codificado de borrado de 120 Gi disponible para los usuarios.
- La calculadora de codificación Minio Erasure está disponible. Para obtener más información, consulte la calculadora de códigos de borrado.

Configuración de reclamaciones por volumen persistente

Wickr Enterprise requiere que Persistent Volume Claims almacene datos con estado. Esta configuración le permite especificar el nombre de la clase de almacenamiento que desea utilizar. Si se deja en blanco, Wickr intentará usar la clase de almacenamiento predeterminada. No se admite el cambio de la clase de almacenamiento después de implementar Wickr.

Los proveedores de servicios en la nube suelen proporcionar un valor predeterminado StorageClass para las reclamaciones por volumen persistentes; sin embargo, en las instalaciones totalmente locales, es posible que sea necesaria una configuración explícita mediante un servicio de terceros, como Longhorn.

Configuración del certificado TLS

Cargue un certificado PEM y una clave privada para finalizar el TLS. El nombre alternativo del sujeto del certificado debe coincidir con el nombre de host configurado en la configuración de su implementación de Wickr Enterprise.

Para el campo de cadena de certificados, concatene todos los certificados intermedios (si es necesario) con el certificado de CA raíz antes de cargarlos.

Let's Encrypt

Seleccione esta opción para generar automáticamente un certificado mediante Let's Encrypt. Los certificados se emiten mediante el desafío HTTP-01 a través del operador cert-manager.

El desafío HTTP-01 requiere que el nombre de DNS deseado se dirija al punto de entrada del clúster (normalmente un Load Balancer) y que el tráfico al puerto TCP 80 esté abierto al público. Estos certificados son de corta duración y se renovarán periódicamente. Es necesario mantener abierto el puerto 80 para permitir que los certificados se renueven automáticamente.

> **Note**
>
> Esta sección hace referencia explícita al certificado utilizado por la propia aplicación Wickr Enterprise.

Certificado anclado

Wickr Enterprise requiere la fijación de certificados cuando se utilizan certificados autofirmados o certificados en los que los dispositivos cliente no confían. Si el certificado presentado por tu Load Balancer es autofirmado o lo ha firmado una entidad emisora de certificados diferente a la de la instalación de Wickr Enterprise, sube el certificado de CA aquí para que los clientes lo coloquen en su lugar.

En la mayoría de los casos, esta configuración no es obligatoria.

Proveedores de certificados

Si planeas comprar un certificado para usarlo con Wickr Enterprise, consulta a continuación una lista de proveedores cuyos certificados se sabe que funcionan correctamente de forma predeterminada. Si un proveedor aparece en la lista siguiente, sus certificados se han validado con el software de forma explícita.

- Digicert
- RapidSSL

Generar un certificado autofirmado

Si quieres crear tu propio certificado autofirmado para usarlo con Wickr Enterprise, el siguiente comando de ejemplo contiene todos los indicadores necesarios para su generación.

```
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 deseas crear un certificado autofirmado basado en IP, usa el siguiente comando en su lugar. Para utilizar el certificado basado en IP, asegúrese de que el campo Wildcard Hostname esté activado en la configuración de Ingress. Para obtener más información, consulte Configuración de ingreso.

```
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**
>
> Sustituya $YOUR_DOMAIN en el ejemplo por el nombre de dominio o la dirección IP que piensa usar.

Configuración de llamadas

- Requerir nodos de llamada: cuando esta configuración está habilitada, los servicios de llamadas de Wickr solo se implementan en los nodos de Kubernetes con la etiqueta. `role=calling` Inhabilita esta configuración para implementar los servicios de llamadas y mensajería en los mismos nodos o para despliegues de un solo nodo.

  Por lo general, también querrá deshabilitar el proxy TCP de llamada cuando esta configuración esté deshabilitada, ya que el servicio de proxy TCP se ejecuta en el puerto 443.
- Habilitar el proxy TCP: esta configuración controla si se implementa o no el servicio para el modo alternativo TCP en las llamadas. Deshabilite esta configuración si tiene otros servicios que se ejecutan en 443/tcp o si no necesita el modo alternativo de TCP para las llamadas. Esta opción debe estar habilitada para las implementaciones que planean usar Wickr Open Access.
- Detectar automáticamente las direcciones IP públicas del servidor: cuando esta configuración está habilitada, los servicios de llamadas descubrirán su dirección IP pública realizando solicitudes HTTPS a https://ipv4.icanhazip.com/ y. https://ipv6.icanhazip.com/ Si está deshabilitada, debe habilitar la configuración «Usar la dirección IP principal del host para el tráfico de llamadas» o «Anular el nombre de host»; de lo contrario, los servicios de llamadas no se iniciarán.
- Use la dirección IP principal del host para el tráfico de llamadas: use la dirección IP principal de los nodos de Kubernetes para los servicios de llamadas. Esto implica que todos los clientes de Wickr pueden conectarse a tus nodos de Kubernetes con la dirección IP principal del nodo, tal como se muestra en la API Downward. `status.hostIP`
- Anulación del nombre de host: proporciona un nombre de host o una dirección IP para utilizarlos como punto de conectividad para los servicios de llamadas. Esta configuración solo debe usarse cuando se ejecuta un único servidor de llamadas, ya que se devuelve el mismo valor para todas las réplicas del servicio. Cuando se establece una anulación del nombre de host y se habilita

la opción «usar la dirección IP principal del host», prevalece la configuración de la dirección IP principal del host.

- Red host de llamadas habilitada: de forma predeterminada, los pods de llamadas utilizan la red host de los nodos para conectarse. Desactívela para exponer un NodePort servicio al tráfico de llamadas. Si la entrada de llamadas está habilitada, asegúrese de que el servicio adecuado esté configurado para permitir el tráfico de entrada. Esto debe estar deshabilitado para cumplir con el STIG.

Configuración de ingreso de llamadas

Wickr admite una configuración de entrada de llamadas, lo que permite al cliente conectarse a cualquier nodo de llamada del clúster y disponer de la ruta de llamada al servidor de llamada correcto. Wickr admite cuatro tipos de ingreso de llamadas:

- LoadBalancer (predeterminado)
  - Lo LoadBalancer aprovisionará el proveedor de la nube (las instalaciones totalmente locales requerirán una configuración adicional). Una vez aprovisionada, se debe volver a actualizar la configuración de KOTS para proporcionar el nombre de host o las direcciones IP del balanceador de cargas. LoadBalancer
- NodePort
  - Expone un NodePort servicio en cada nodo de llamada que servirá como punto de entrada para el tráfico de llamadas. Se debe proporcionar un nombre de host que se dirija a uno o más nodos o una dirección IP de uno o más nodos. Puede elegir un rango de puertos entre 30000-32767 para el tráfico UDP y, opcionalmente, el TCP.
- NLB existente
  - Adjunta el servicio de entrada de llamadas a un NLB existente. Deberá proporcionar el ARN del grupo de destino para el tráfico UDP y, opcionalmente, el TCP.
- Sin servicio
  - Seleccione esta opción si no necesita un servicio de Kubernetes adicional para permitir la entrada de tráfico. Por lo general, se usa con la configuración de la red del host para enrutar el tráfico entrante de llamadas directamente a los nodos de llamada.

Consideraciones

- Para garantizar la compatibilidad con versiones anteriores de clientes y redes federadas sin entrada de llamadas, si la entrada de llamadas está habilitada, el modo de llamada anterior sigue estando disponible (conexión directa a los servidores de llamadas). Si cambias algún puerto predeterminado, asegúrate de que no haya colisiones de puertos en los nodos de llamada.
- La pila doble NLBs que sirve el tráfico UDP debe tener objetivos de IPv6 back-end. Para obtener más información, consulte Grupos objetivo de Network Load Balancer.
- Si necesita cumplir con las normas STIG, debe deshabilitar la opción de red host para realizar llamadas. Si los nodos están configurados en modo de doble pila, pero el clúster no, es posible que pierda la IPv6 conectividad (suponiendo que se trate de un IPv4 clúster).
- La entrada de llamadas requiere nombres de host o direcciones IP predefinidos. Para escalar los nodos o proporcionar un enrutamiento personalizado, es posible que sea necesario modificar la configuración.
- Los puertos de entrada de llamadas predeterminados son 8443 para TCP y 16384 para UDP. Asegúrese de que los firewalls y los grupos de seguridad permitan el tráfico en estos puertos o en puertos alternativos si se anulan los valores predeterminados.

Arquitecturas de referencia

Entrada con balanceador de carga

Esta opción expone un único balanceador de cargas como punto de entrada para todo el tráfico de llamadas.

1. Para el tipo de ingreso de llamadas, elija Load Balancer o Existing NLB. Para obtener más información sobre el NLB existente, consulte la pila de NLB del ejemplo del CDK de Wickr Enterprise en. GitHub
2. Realice una de las siguientes acciones, según el tipo de entrada de llamadas:
    - En el caso del NLB existente, indique el grupo ARNs objetivo del tráfico UDP y TCP y el nombre de host del NLB.
    - En el caso de Load Balancer, proporciona el nombre de host una vez que Kubernetes lo haya aprovisionado.

Como alternativa, para cualquier tipo de entrada de llamadas, puedes proporcionar las direcciones IP del balanceador de cargas o un nombre de host personalizado que apunte al balanceador de cargas.

3. (Opcional) Para combinar el tráfico de mensajería y llamadas en un único NLB, selecciona el NLB existente en la sección de ingreso y proporciona un grupo objetivo HTTPS.

Ingresa con NodePort

Esta opción resulta útil si la red del host está deshabilitada y no quieres exponer un balanceador de carga adicional.

> **Note**
>
> Asegúrese de que sus firewalls y grupos de seguridad permitan el tráfico de. NodePorts

1. Para el tipo de ingreso de llamadas, elija. NodePort
2. Agregue los nombres de host o las direcciones IP del nodo de llamada.
3. Deshabilite la red de host de llamadas.

Entrada directa con HostNetwork

Esta opción no expone ningún servicio de Kubernetes adicional y permite que el tráfico de entrada de llamadas se conecte directamente a través de la red host de los nodos que llaman. Se prefiere este enfoque si se requiere conectividad. IPv6

1. Para el tipo de ingreso de llamadas, selecciona Sin servicio.
2. Agregue los nombres de host o las direcciones IP del nodo de llamada.
3. Habilite la red de hosts de llamadas.

Escalador automático de clústeres de Kubernetes (opcional)

El escalador automático de clústeres de Kubernetes es un valor de configuración opcional para la instalación de Wickr Enterprise. Ayudará a escalar los grupos de nodos de Kubernetes en caso

de que aumente el tráfico u otras restricciones de recursos que puedan provocar un rendimiento deficiente.

La instalación de Wickr Enterprise admite tres integraciones de proveedores de nube: Google Cloud AWS y Azure. Cada proveedor de nube tiene requisitos diferentes para esta integración. Siga las instrucciones para su proveedor de nube específico que aparecen a continuación para habilitar esta función.

AWS

Si no utilizaste el WickrEnterprise CDK para instalar tu entorno de Wickr AWS, tendrás que tomar algunas medidas adicionales para habilitar el escalador automático de clústeres.

1. Añada las siguientes etiquetas a sus grupos de nodos. Esto permite que el escalador automático de clústeres descubra automáticamente los nodos adecuados.
    1. `k8s.io/cluster-autoscaler/`**`clusterName`** `= owned`donde ClusterName es el nombre de tu clúster de Kubernetes
    2. `k8s.io/cluster-autoscaler-enabled = true`
2. Añada una cuenta de servicio de Kubernetes en el espacio de nombres del sistema kube y asóciela a una política de IAM que permita el escalado automático y las acciones de ec2. Para obtener más información e instrucciones detalladas, consulte Configuración de una cuenta de servicio de Kubernetes para asumir una función de IAM en la Guía del usuario de Amazon EKS.
    1. Deberá utilizar el espacio de nombres «kube-system» al configurar la cuenta de servicio
    2. Se puede usar la siguiente política para la cuenta de servicio:

       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"
        }
    ]
}
```

En la interfaz de usuario replicada, al configurar el escalador automático de clústeres, seleccione AWSsu proveedor de nube e indique el nombre de la cuenta de servicio que creó anteriormente para indicar al escalador automático de clústeres que utilice esa cuenta de servicio.

Nube de Google

Se recomienda encarecidamente utilizar las funciones de escalado automático integradas de GKE tanto para el piloto automático como para los clústeres estándar. Sin embargo, si deseas continuar con esta integración, debes cumplir los siguientes requisitos antes de continuar.

Requisitos:

1. Los grupos de instancias administrados (MIG) deben crearse con un ámbito de seguridad que incluya, como mínimo, la función de lectura y escritura de los recursos de Compute Engine. Actualmente, esto no se puede agregar al MIG más adelante.
2. El clúster debe tener habilitada la federación de identidades de carga de trabajo. Para habilitarlo en un clúster existente, ejecute: `gcloud container clusters update ${CLUSTER_NAME} --workload-pool=${PROJECT_ID}.svc.id.goog`
3. Una cuenta de servicio de Google Cloud Platform (GCP) con acceso al rol `roles/compute.instanceAdmin.v1`. Se puede crear siguiendo estas instrucciones:

```
# 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) proporciona un escalado automático de clústeres integrado para la mayoría de las implementaciones y se recomienda encarecidamente utilizar esos métodos para el escalado automático de clústeres. Sin embargo, si sus requisitos hacen que esos métodos no funcionen, le ofrecemos una integración con el escalador automático de clústeres de Kubernetes para Azure Kubernetes Service. Para utilizar esta integración, tendrá que recopilar la siguiente información y ponerla en la configuración del panel de administración de KOTS, en Cluster Autoscaler, después de seleccionar Azure como su proveedor de nube.

Autenticación de

ID de suscripción: el ID de suscripción se puede obtener a través del portal de Azure siguiendo la documentación oficial. Para obtener más información, consulte Obtener una suscripción y un inquilino IDs en el portal de Azure.

Los siguientes parámetros se pueden obtener creando un director de servicio de AD mediante la utilidad de línea de comandos az.

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

ID de aplicación:

Contraseña del cliente:

ID de inquilino:

Configuración del escalador automático de clústeres de Azure

Además de los requisitos de autenticación, los siguientes campos son necesarios para el correcto funcionamiento del escalador automático del clúster. Los comandos para obtener esta información

se proporcionan por comodidad; sin embargo, es posible que requieran algunas modificaciones en función de la configuración específica de AKS.

Grupo de recursos de nodos administrados de Azure: este valor es el grupo de recursos administrados que creó Azure al establecer el clúster de AKS y no el grupo de recursos que definió. Para obtener este valor, necesita los valores CLUSTER_NAME y RESOURCE_GROUP de cuando creó el clúster. Una vez que tenga esos valores, puede obtenerlos ejecutando:

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

Nombre del VMSS del grupo de nodos de la aplicación: es el nombre del conjunto de escalado de máquinas virtuales (VMSS) asociado a su grupo de nodos de AKS para la aplicación Wickr. Este es el recurso que se ampliará o reducirá en función de las necesidades del clúster. Para obtener este valor, puede ejecutar el siguiente comando az:

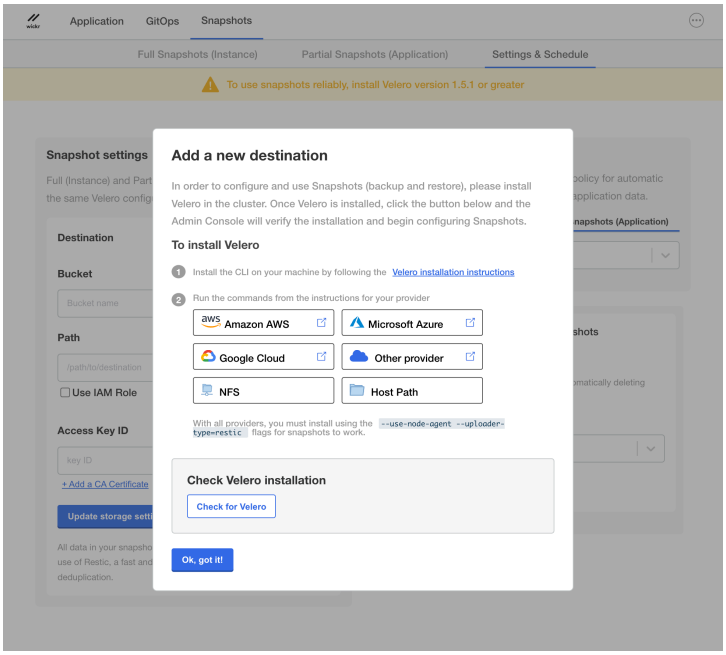
```
CLUSTER_NODEPOOL_NAME="(Your-NodePool-Name)"
CLUSTER_RESOURCE_GROUP="(Your-Managed-Node-Resource-Group-As-Defined-Above>)"
az vmss list -g ${CLUSTER_RESOURCE_GROUP} --query '[?tags."aks-managed-
poolName"==`'''${CLUSTER_NODEPOOL_NAME}'''`].{VMSS_name:name}' -o tsv
```

ACalling Nombre del VMSS del grupo de nodos (opcional): es el nombre del VMSS asociado al grupo de nodos al que llamas, si lo tienes. Para obtener este valor, puede ejecutar una versión modificada del comando para el nombre VMSS del grupo de nodos de la aplicación, sustituyendo el valor CLUSTER_NODEPOOL_NAME por el nombre del grupo de nodos del grupo de nodos al que llama.

Copias de seguridad

Wickr Enterprise utiliza Velero para fines de Backup. Velero proporciona las herramientas necesarias para realizar copias de seguridad y restaurar los recursos del clúster y los volúmenes persistentes de Kubernetes, ya sea que se opere en un proveedor de nube o de forma local.

Copias de seguridad de Velero con Minio: Actualmente, las copias de seguridad de Velero solo están habilitadas para Minio en el modo de bajos recursos.



Instalación mediante la documentación de Velero

- Instale la CLI de Velero. Para obtener más información, consulte Instalación de la CLI de Velero.
- Instale Velero en su clúster y configure el almacenamiento según su proveedor:
  - AWS.
  - GCP.
  - Azure.
  - Otros proveedores.

Limitación

De forma predeterminada, no se incluye ningún volumen en la copia de seguridad. Si algún módulo monta un volumen del que deba hacerse una copia de seguridad, debe configurar la copia de seguridad con una anotación que enumere los volúmenes específicos que se van a incluir en la copia de seguridad.

Para cada volumen que requiera una copia de seguridad, agrega el archivo backup.velero. io/ backup-volumes annotation. The annotation name is backup.velero.io/backup-volumes y el valor es una lista de volúmenes separados por comas para incluirlos en la copia de seguridad. Para obtener más información, consulte Configurar instantáneas.

Instalación de Airgap

Tanto Wickr Enterprise como KOTS admiten el despliegue en un clúster de Kubernetes totalmente integrado. Debes proporcionar acceso a un registro de imágenes de Docker privado al que se pueda acceder desde el clúster de Kubernetes reducido. El registro privado de imágenes de Docker suministrado a KOTS debe estar protegido con autenticación para que funcione correctamente con este fin. username/password KOTS utilizará el registro privado de imágenes de Docker para alojar todas las imágenes de Wickr Enterprise.

- Wickr Enterprise license.yaml con airgap activado (póngase en contacto con el equipo de ventas o soporte al cliente de Wickr)
- Paquete de archivos wickr.airgap de Wickr Enterprise (póngase en contacto con el equipo de ventas o atención al cliente de Wickr)
- Acceso a un registro privado de imágenes de Docker.
- Acceso a un clúster de Kubernetes implementado en el entorno airgap.
- Kubectl instalado.
- CLI de KOTS instalada.
- kotsadm.tar.gz descargado.

Ejecuta los siguientes comandos para implementar KOTS y Wickr Enterprise en tu clúster de kubernetes reducido. Estos comandos cargan las imágenes de administrador de KOTS y las imágenes de Wickr Enterprise al registro privado de imágenes de Docker. Una vez finalizados los comandos, se te pedirá que accedas a la consola de administración de KOTS para completar la instalación de Wickr Enterprise, tal y como se indica arriba.

```
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
```

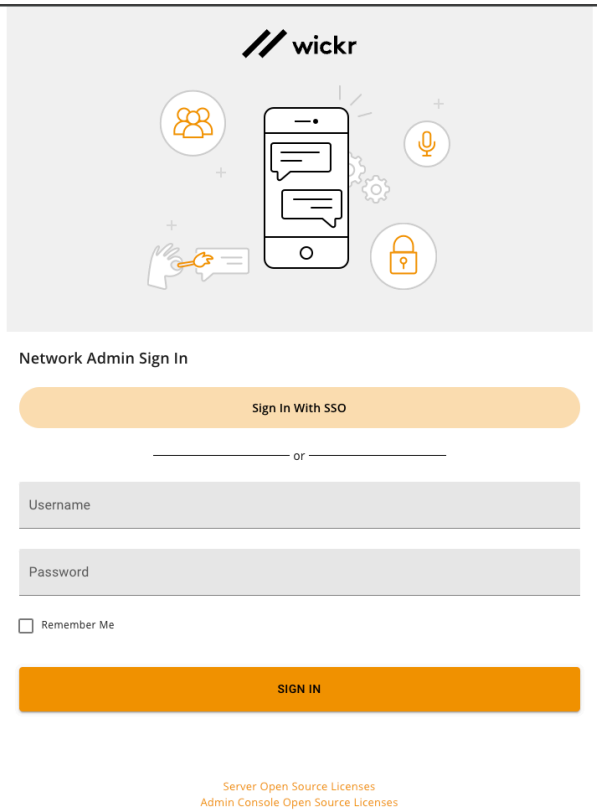
Notificación móvil para las instalaciones de airgap

Se necesitan listas adicionales de redes permitidas para enviar notificaciones push desde el servidor a los clientes móviles. Este requisito se debe a la forma en que Apple iOS y Google Android implementan esta función para dispositivos fuera de línea y en segundo plano. Consulte la documentación de estos servicios para obtener una lista de las direcciones IP y los puertos especificados.

- iOS
- Android

Consola de administración de Wickr

La interfaz de la consola de administración de Wickr se utiliza para administrar la propia aplicación Wickr Enterprise. Se puede usar para configurar redes, usuarios, federaciones y más. Se puede acceder a ella a través de HTTPS con el nombre de DNS que configuró para que apunte a su equilibrador de carga. El nombre de usuario predeterminado es admin, con la contraseña Password123. Se le pedirá que cambie esta contraseña la primera vez que inicie sesión.



Configuración de seguridad

AWS Wickr Enterprise proporciona opciones de configuración para aplicar un contexto de seguridad mejorado para su implementación. Este estándar de seguridad más elevado se aplica a nivel de módulo y contenedor, y es obligatorio para cumplir con la Guía de implementación técnica de seguridad (STIG).

Establezca los siguientes parámetros de configuración para aplicar el contexto de seguridad mejorado:

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

> **Warning**
>
> En el caso de Opensearch, esta configuración de seguridad desactiva el `fsgroup-volume` InitContainer que actualiza los permisos del almacenamiento persistente, lo que puede provocar problemas de compatibilidad relacionados con los permisos.

Preguntas frecuentes

P: Mi implementación falla y aparece el siguiente error en helm stderr:

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

R: Esto puede suceder cuando el registro de depuración está activado. Desactive el registro de depuración, elimine los trabajos problemáticos e inténtelo de nuevo.

Clúster integrado para Wickr Enterprise

La opción de instalación de clústeres integrados para Wickr Enterprise ofrece una oferta de instalación pequeña y eficiente para el producto Wickr Enterprise. Aprovecha el clúster integrado replicado para proporcionar una instalación pequeña de Kubernetes mediante k0s en la que se puede instalar Wickr Enterprise. El uso de este método de instalación minimiza los requisitos de habilidades técnicas y los requisitos generales de hardware para una instalación de Wickr Enterprise, ya que proporciona una solución «all-in-one» a costa de la resiliencia y la alta disponibilidad.

Temas

Cómo empezar con el clúster integrado de Wickr Enterprise

Para empezar a utilizar la opción de clúster integrado de Wickr Enterprise, ponte en contacto con el servicio de asistencia para recibir una licencia. Si ya tienes una licencia y deseas utilizar esta opción, ponte en contacto con el servicio de asistencia para que te ayuden a actualizar tu licencia actual e instrucciones de instalación adicionales.

Requisitos del clúster integrado de Wickr Enterprise

Antes de empezar a instalar el clúster integrado de Wickr Enterprise, compruebe que se cumplen los siguientes requisitos.

Requisitos de red

Deberás permitir la entrada a tu servidor Wickr en los siguientes puertos:

- 443/TCP para HTTPS

- Solo llamadas al proxy TCP: el puerto proxy TCP configurado para el tráfico de llamadas TCP en KOTS
- 16384-19999/UDP para el tráfico de llamadas UDP
- Solo LAN: 30000/TCP para acceder a la consola de administración de KOTS

Requisitos del sistema

Antes de la instalación, asegúrese de tener una VM (máquina virtual) o una máquina física que ejecute un sistema operativo (SO) basado en Linux con los siguientes recursos mínimos disponibles:

- 8 núcleos de CPU
- 12 gigabytes (GB) de RAM
- 100 gigabytes (GB) de almacenamiento en disco en la partición/(raíz)

El clúster integrado Wickr Enterprise se ha probado en los siguientes sistemas operativos Linux, pero también pueden ser adecuadas otras opciones de sistemas operativos basados en Linux:

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

Instalación del clúster integrado de Wickr Enterprise (estándar)

Una vez que tengas las instrucciones de descarga, descarga el paquete Wickr Enterprise en la máquina de destino y desempaquétalo.

```
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
```

Ahora deberías tener dos archivos, y`wickr-enterprise-ha`. `license.yaml` El `wickr-enterprise-ha` archivo es un archivo binario que incluye todas las piezas necesarias para la instalación del clúster integrado, mientras que `license.yaml` es su licencia de Wickr la que se utilizará para validar la instalación.

En esta etapa, se puede realizar una instalación básica ejecutando el `wickr-enterprise-ha` archivo:

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

Una vez que comience el proceso de instalación, se le solicitará que introduzca una contraseña para la Admin Console. Introduzca una contraseña segura y asegúrese de guardarla, ya que la necesitará al acceder a la consola de administración de KOTS para continuar configurando la instalación.

Una vez completada la instalación, el resultado es similar al siguiente:

```
   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
```

Tras la instalación estándar, diríjase a la URL de la consola de administración de KOTS que aparece en el resultado mediante un navegador web. Para este ejemplo, la URL es`http://192.168.1.100:30000`. Sin embargo, la URL diferirá en función de la configuración de red.

Instalación de varios nodos

Las instalaciones multinodo de Wickr Enterprise Embedded Cluster ofrecen a los usuarios de clústeres integrados la opción de separar las cargas de trabajo de Wickr Calling y Wickr Messaging en diferentes máquinas físicas. Para ello, Wickr Enterprise aprovecha las herramientas de varios nodos del clúster integrado replicado.

Requisitos de los puertos

Los siguientes puertos deben estar abiertos en todos los miembros del clúster para que la funcionalidad de varios nodos funcione correctamente. Estos solo deben estar abiertos entre los propios nodos y no estar abiertos a Internet en general.

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

Requisitos de licencia

Las opciones de configuración de varios nodos del clúster integrado de Wickr requieren privilegios de licencia adicionales. Póngase en contacto con nosotros Soporte para asegurarse de que su licencia sea compatible con esta función.

Crear un nodo adicional durante la configuración inicial

Al configurar inicialmente el clúster integrado de Wickr Enterprise, puede crear un nodo de llamada adicional durante el proceso de configuración. Comience por seguir el procedimiento descrito en Instalación del clúster integrado de Wickr Enterprise (estándar). Cuando vaya al panel de administración de KOTS, se le pedirá que cree nodos adicionales.

> **Note**
>
> Actualmente, el clúster integrado de varios nodos solo admite 1 nodo de llamada y 1 nodo. messaging/controller

Para empezar, deseleccione la opción de función de controlador y seleccione la opción de función de llamada. Esto rellena conjuntos de instrucciones adicionales para configurar el nuevo nodo. Ejecute estas instrucciones en el nuevo nodo para configurarlo para que se una al clúster como nodo de llamada.

Ejecute instrucciones similares a las de los ejemplos siguientes en el nuevo nodo:

1. Descargue el binario en el nuevo nodo:

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

2. Extraiga el binario:

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

3. Une el nodo al clúster:

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

Cuando el comando de unión se complete correctamente, el nuevo nodo aparecerá en la página Configurar el clúster con la función de llamada asignada. Selecciona Continuar para ir a la página de configuración de Wickr Enterprise. Siga las instrucciones para ver las opciones de configuración de nodos integrados que se describen en la configuración de la consola de administración de KOTS.

Añadir un nodo adicional a una instalación de clúster integrada existente

Para añadir un nodo de llamada a una instalación de Wickr Enterprise Embedded Cluster existente, navegue hasta la consola de administración de KOTS. Para ello, inicie sesión en el nodo mediante ssh u otro mecanismo y navegue hasta el directorio de instalación que contiene el `wickr-enterprise-ha` binario utilizado para la instalación. Ejecute `./wickr-enterprise-ha admin-console` para iniciar la consola de administración de KOTS. Si este comando no devuelve ningún

resultado, significa que la consola de administración de KOTS ya se está ejecutando y se puede acceder a ella desde el puerto 30000 de la IP del nodo en un navegador web, por ejemplo:. `https://127.0.0.1:30000/`

Introduzca la contraseña de administrador de KOTS cuando se le solicite y, a continuación, realice el siguiente procedimiento para crear un nodo adicional:

1. Una vez que haya iniciado sesión, vaya a la página de administración de clústeres en la parte superior izquierda de la consola de administración de KOTS.
2. Elija Add node (Agregar nodo).
3. Deselecciona Controlador en la parte inferior. `Roles`
4. Selecciona Llamar en `Roles`
5. Siga las instrucciones proporcionadas para ejecutar los comandos en el nuevo nodo que desee agregar.
6. Cuando haya terminado, elija Cerrar
7. El nuevo nodo aparece en la lista de nodos con la función de llamada.
8. Ve a la página de la aplicación en la parte superior izquierda de la consola de administración de KOTS
9. Selecciona Config en la barra de navegación de la parte superior de la página.
10. Ve a la sección de llamadas en el panel de navegación izquierdo.
11. Seleccione Requerir nodos de llamada para permitir el uso del nodo de llamada.
12. Desplázate hasta la parte inferior de la página y selecciona Guardar configuración.
13. Aparece una ventana emergente que indica que la Config se ha actualizado. Selecciona Ir a la versión actualizada.
14. En la página de la versión actualizada, se muestra la versión actualmente instalada. Aparece un nuevo elemento de línea en las versiones instaladas con la designación Config Change. Elija Deploy para implementar esta nueva versión y habilitar el nuevo nodo de llamada.

Configuración de la consola de administración de KOTS

La consola de administración de KOTS utiliza inicialmente un certificado autofirmado, que tendrás que permitir como excepción en tu navegador. Una vez que aceptes esta excepción, el asistente de configuración de la consola de administración de KOTS te dará la bienvenida. Este asistente te

guía por los pasos de configuración adicionales para configurar el comportamiento de la consola de administración de KOTS, incluida la opción de añadir un certificado personalizado si es necesario.

Una vez completada la configuración inicial de la consola de administración de KOTS, se le solicitará que introduzca la contraseña de la consola de administración que creó durante el proceso de instalación. Al iniciar sesión por primera vez, debe configurar el clúster.

Selecciona Continuar para ir a la consola de administración de KOTS para Wickr.

Para un clúster integrado de un solo nodo, selecciona Continuar para ir a la consola de administración de KOTS para Wickr. Para instalaciones con varios nodos, consulta Instalación con varios nodos.

Una vez en la consola de administración de KOTS, configure la instalación según sus necesidades. Al utilizar la oferta de clústeres integrados, hay algunos ajustes de configuración clave que deben configurarse para garantizar el correcto funcionamiento de la instalación de Wickr Enterprise.

- Nombre de servidor: es el nombre de servidor que utilizas cuando te comunicas con la instalación de Wickr. Asegúrese de crear los registros DNS adecuados para que este dominio apunte a su instalación de Wickr Enterprise.
- En Opciones avanzadas, selecciona la opción [] Configurar el controlador de entrada para ver un bloque de configuración que permite configurar la entrada de Kubernetes. En el bloque de configuración de Ingress, selecciona Clúster integrado de nodo único y, a continuación, introduce la IP «pública» asociada a tu servidor Wickr en el cuadro de texto denominado Loadbalancer External IP (solo). IPv4

  Si no estás seguro de cuál es esta IP, puedes ejecutar el siguiente comando desde la línea de comandos del servidor Wickr para determinar este valor: `ip route get 1.1.1.1|awk '{print $7}'`
- En Opciones avanzadas, marca la opción Habilitar el modo de bajos recursos.
- En Llamar, si utiliza un clúster integrado de un solo nodo, asegúrese de que la opción Requerir nodos de llamada no esté seleccionada. De lo contrario, si ha agregado un nodo de llamada durante la configuración inicial, asegúrese de que esté seleccionada la opción Requerir nodos de llamada.
- Si desea una solución integral que no utilice una base de datos externa o un almacenamiento compatible con S3 para compartir archivos, seleccione las opciones internas para los siguientes ajustes:
  - Base de datos

- Ubicación de almacenamiento de S3

  La ubicación de almacenamiento S3 interna ofrece opciones adicionales para configurar la capacidad de almacenamiento. Se recomienda empezar con algo pequeño y ampliarlo según sea necesario, ya que la reducción no es una opción después del aprovisionamiento.

Una vez que haya configurado todas las funciones necesarias, desplácese hasta la parte inferior de la página de configuración y elija Save Config. Esto iniciará algunas comprobaciones previas a la verificación del host. Una vez finalizadas las comprobaciones previas, selecciona Deploy para iniciar la instalación de Wickr Enterprise.

Ahora está listo para comenzar a configurar su instalación de Wickr Enterprise. Para obtener más información sobre la configuración de Wickr Enterprise, consulta ¿Qué es Wickr Enterprise? .

Requisitos de instalación comunes adicionales

Instalaciones de nombres de host IP

Si su instalación requiere un nombre de host basado en IP, hay algunas opciones de configuración adicionales. Estas instrucciones son específicas para los nombres de host basados en IP y se recomienda seguir las demás instrucciones para la configuración básica que se indican anteriormente.

En el panel de administración de KOTS, complete los siguientes pasos.

1. Establezca el nombre de host en la IP que utilizará.
2. En Certificados, selecciona Cargar un certificado. A continuación, genere un certificado autofirmado siguiendo las instrucciones de un certificado basado en IP. Para obtener más información, consulte Generar un certificado autofirmado.
3. Cargue el `.crt` archivo del certificado y el `.key` archivo de la clave privada
4. Para la cadena de certificados, vuelva a cargar el `.crt` archivo.
5. Marque la casilla de verificación Establecer un certificado fijo.
6. Cargue el `.crt` para el certificado anclado.
7. En Llamar, desactive las casillas Detectar automáticamente las direcciones IP públicas del servidor y Usar la dirección IP principal del host para el tráfico de llamadas.
8. En Llamar, coloque la dirección IP del nombre de host en el cuadro de texto Reemplazar el nombre de host.

9. En Opciones avanzadas, active la casilla de verificación Configurar el controlador de entrada. A continuación, aparece una nueva sección de configuración llamada Ingress.
10. En Ingress, seleccione Clúster integrado de nodo único.
11. En Ingress, introduce la IP de la interfaz «pública» del servidor Wickr. Puede ser diferente de la IP utilizada como nombre de servidor. Consulte información adicional sobre este valor en los pasos básicos de configuración.
12. En Ingress, active Usar un nombre de host comodín.

SELinux Modo de aplicación

Si necesita usarlo SELinux en modo obligatorio, modifique el directorio de datos predeterminado utilizado para instalar el clúster integrado. Se recomienda su uso, `/opt` ya que se ha probado que funciona con la mayoría de SELinux las políticas para este caso de uso.

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

Las comprobaciones previas a la instalación predeterminada de los clústeres integrados replicados intentarán validar si SELinux se encuentra en modo permisivo y fallarán si SELinux se encuentra en modo obligatorio. Para evitar esto, es necesario utilizar el argumento de la línea de comandos. `--ignore-host-preflights` Al utilizar la opción de línea de comandos, aparece un mensaje similar al que se muestra a continuación. Introduzca Sí cuando se le pida.

```
#  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 instalaciones

La opción de instalación de clústeres integrados para Wickr Enterprise admite instalaciones aisladas. Se requieren configuraciones y habilitaciones adicionales para su licencia. Póngase en contacto con

el servicio de asistencia si está interesado en utilizar el clúster integrado de Wickr Enterprise en un entorno aislado.

Al realizar una instalación de airgap, las instrucciones de descarga difieren del método de instalación estándar. Deberían parecerse a lo siguiente:

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

Descargue el paquete en una máquina que tenga acceso a Internet y, a continuación, transfiéralo a su entorno aislado utilizando el método de transporte de datos que prefiera. Una vez transferido el paquete, extráigalo como lo haría con cualquier paquete de instalación estándar. Se incluirá un tercer archivo`wickr-enterprise-ha.airgap`, que contiene todas las imágenes del servicio de aplicaciones de Wickr Enterprise asociadas.

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

Durante la instalación, es necesario establecer el argumento de la línea de `--airgap-bundle` comandos después de la extracción; de lo contrario, el proceso sigue el procedimiento de instalación estándar.

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

Actualización de un clúster integrado con AirGapped

Para actualizar un clúster AirGapped integrado, complete los siguientes pasos.

1. Descargue el nuevo paquete de clústeres integrados de Replicated y transfiéralo a la máquina host mediante los métodos de transferencia de datos estándar para su entorno fragmentado. Una vez que el nuevo paquete esté en la máquina host, extraiga el archivo tar:

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

2. Ejecute la actualización con el nuevo paquete binario y airgap:

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

3. Inicie la consola de administración de KOTS e inicie sesión en la URL proporcionada utilizando sus métodos estándar de acceso a la consola de administración de KOTS

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

4. Una vez que hayas iniciado sesión en la Consola de administración de KOTS, busca la última actualización disponible a la izquierda, debajo de la sección Versión, y luego presiona el botón Ir al historial de versiones.
5. Selecciona Implementar para la nueva versión en Actualizaciones disponibles. Recorre las pantallas:
    1. Cambie las opciones de configuración, desplácese hacia abajo y, a continuación, seleccione Siguiente.
    2. Compruebe que no se haya producido ningún error en las comprobaciones previas y seleccione Siguiente: confirmar e implementar.
    3. Elija Implementar.

Notas adicionales sobre el clúster integrado de Wickr Enterprise

- ESPACIO DE NOMBRES: A diferencia de la mayoría de las instalaciones de Wickr Enterprise, la instalación del clúster integrado instala los activos de Wickr en el espacio de nombres kotsadm de Kubernetes y no en wickr. Modifica cualquier script o comando que hayas guardado y que utilices para utilizarlos en su lugar con kubectl, helm o cualquier otra utilidad. `-n wickr` `-n kotsadm`
- Interacción con el clúster de Kubernetes: desde la máquina host, usa el `./wickr-enterprise-ha` binario para crear un shell con las variables adecuadas configuradas para interactuar con la instalación de Kubernetes mediante la ejecución. ./wickr-enterprise-ha shell Esto proporcionará la utilidad kubectl dentro de la ruta del shell y establecerá la configuración de kube adecuada para la instalación local.

Solución de problemas en instalaciones de clústeres integrados de Wickr

En todas las instancias de estos pasos de solución de problemas se supone que tienes acceso desde el shell a la instancia que ejecuta la instalación del clúster integrado de Wickr y que has ejecutado el `./wickr-enterprise-ha shell` comando para poder interactuar directamente con la instalación de Kubernetes.

Problemas generales

Falta el botón Añadir nodo en la pantalla de administración del clúster

Instalaciones aisladas

Si estás en una instalación de airgap, ponte en contacto con el equipo de soporte de Wickr para que te ayude a corregir este comportamiento.

Instalaciones estándar

Si su licencia incluye la licencia Embedded Cluster Multi-Node, sincronice la licencia para obtener la versión más reciente. Si no estás seguro o no tienes este derecho, ponte en contacto con Wickr Support.

Para realizar una sincronización de licencias, sigue estos pasos.

1. Navegue hasta el panel de control de KOTS.
2. En la página del panel de control, localice la sección de licencias en el área superior derecha de la página.
3. Dentro de esta sección, en la esquina superior derecha, debería ver un hipervínculo de sincronización de licencias. Seleccione el hipervínculo.
4. Una vez sincronizada la licencia, la interfaz de usuario se actualiza y aparece Última sincronización hace unos segundos.
5. Selecciona Reimplementar en la sección de versiones de la página del panel de control de KOTS.
6. Una vez que finalice la reimplementación, vuelve a la administración de clústeres y podrás añadir nodos.

Problemas de actualización

La actualización se bloquea al actualizar el clúster

Si la actualización se bloquea al actualizar el clúster, es probable que algunos pods no se estén finalizando correctamente. Inicia sesión en la instancia y usa el `./wickr-enteprise-ha shell` comando para ingresar al entorno de shell para administrar la instalación de Kubernetes.

1. Identifica los pods que aún se están ejecutando:

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

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

   > **Note**
   >
   > Si uno de los pods en ejecución es `embedded-cluster-upgrade-XXXXXXXXXXXXXX-xxxxx` `kotsadm-xxxxxxx` o similar, no lo elimines, ya que estos pods son necesarios para realizar la actualización.

3. Comprueba que no queden pods en ejecución.

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

Este procedimiento debería permitir que la actualización del clúster continúe con la actualización de Wickr.

La aplicación no se actualizó durante la actualización del clúster y no puede implementar una nueva versión

Si la aplicación permanece en la versión anterior después de la actualización, es posible que la nueva versión se encuentre en un estado incoherente.

Comprueba los registros de instalación de Kubernetes:

1. Abre el shell de Kubernetes desde el instalador.

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

2. Ejecuta el siguiente comando kubectl:

   ```
   kubectl get installations
   ```

3. El resultado tendrá un aspecto similar al siguiente:

```
      [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. Elimine la instalación fallida.

```
kubectl delete installation 20251113180133
```

5. Intente ejecutar la actualización nuevamente a través del panel de administración de KOTS.

El pod RabbitMQ está fallando con las líneas de registro **Error while waiting for Mnesia tables: {timeout_waiting_for_tables}**

El secreto y el almacenamiento de RabbitMQ no están sincronizados. Esto suele ocurrir cuando se ejecutan varias instancias de RabbitMQ y se produce un error de quórum o de selección de líderes. Para solucionar este problema, elimine el servicio RabbitMQ y sus volúmenes de almacenamiento y, a continuación, vuelva a implementarlo.

Para eliminar el RabbitMQ defectuoso, complete los siguientes pasos.

1. Elimine el conjunto de estados de RabbitMQ.

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

2. Elimine los pods de RabbitMQ restantes. Si hay varios pods de RabbitMQ-x en ejecución, ejecute este comando varias veces actualizando el valor de RabbitMQ-x para que se corresponda con los nombres de los módulos adicionales.

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

3. PVCsElimine el correspondiente. Si hay varios pods en ejecución, ejecute este comando varias veces actualizándolos data-RabbitMQ-X para que se correspondan con los pods correspondientes.

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

4. Comprueba si hay algún pod restante; si se ejecuta correctamente, no se generará ningún resultado.

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

5. Comprueba si queda alguno PVCs, esto no debería generar nada si tiene éxito.

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

6. Vuelva a implementar a través del panel de administración de KOTS.

Para obtener más información sobre la solución de problemas, consulte Solución de problemas.

Historial del documento

En la siguiente tabla se describen las versiones de la documentación de la Guía de instalación automatizada de Wickr Enterprise.

Cambio	Descripción	Fecha
Configuración de seguridad	Se ha añadido la configuración de seguridad. Para obtener más información, consulte Configuración de seguridad.	26 de agosto de 2025
Instalación de varios nodos	Se ha agregado la instalación de varios nodos. Para obtener más información, consulte Instalación de varios nodos.	26 de agosto de 2025
Configuración de ingreso de llamadas	Se ha añadido la configura ción de ingreso de llamadas. Para obtener más información, consulta la sección Configura ción de ingreso de llamadas.	26 de agosto de 2025
Opciones de despliegue automático	Se han añadido opciones de despliegue automático. Para obtener más información, consulte Instalación de Wickr Enterprise.	23 de febrero de 2024
Lista de puertos permitidos	Se agregó el puerto TCP/8443 a la lista de permitidos. Para obtener más información, consulte Requisitos.	12 de febrero de 2024
Destruir los recursos y los puertos de la lista de permitido s	Se han agregado instrucci ones sobre cómo destruir los recursos. Para obtener más información, consulte Destruir	17 de agosto de 2023

	recursos. Además, se han agregado los puertos a la lista de permitidos. Para obtener más información, consulte Requisitos.	
Versión inicial	Versión inicial de la guía de instalación automatizada de Wickr Enterprise	4 de agosto de 2023

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.