aws

Guia de instalação automatizada

Wickr Enterprise

Wickr Enterprise: Guia de instalação automatizada

Table of Contents

O que é o Wickr Enterprise? 1
Conceitos básicos 2
Requisitos 2
Instale as dependências. 2
Configurar 3
Bootstrap 6
Implantar 6
Gere KOTS Config 7
Conectar-se ao Kubernetes 8
Conexões de proxy através do bastion 8
Instalar o Wickr Enterprise 10
Instalando o Wickr Enterprise manualmente 10
Instalando o Wickr Enterprise com o Lambda 10
Pós-instalação 11
Console de administração do KOTS 11
Wickr Admin Console 12
Valores do contexto 13
Como destruir recursos 18
Solução de problemas 19
Excluir o namespace Wickr 19
Redefinir a senha do KOTS Admin Console 19
Problemas na conexão com o cluster EKS com o bastion 19
Instalação personalizada 21
Requisitos 21
Requisitos de hardware 21
Requisitos de software 24
Requisitos de rede 24
Arquitetura 25
Instalação 26
Console de administração do KOTS 26
Configurações de entrada 27
Configurações do banco de dados 28
Configurações do banco de dados externo 28
Configurações internas do banco de dados 28

Armazenamento de arquivos S3 29
Configurações persistentes de reivindicação de volume 31
Configurações do certificado TLS 31
Let's Encrypt 31
Certificado fixado 32
Provedores de certificados 32
Gerando um certificado autoassinado 32
Configurações de chamada 33
Autoescalador de cluster Kubernetes (opcional) 34
AWS 34
Nuvem do Google 35
Azure 36
Backups 37
Instalação usando a documentação do Velero 38
Instalação do Airgap 38
Notificação móvel para instalações de airgap 39
Console de administração do Wickr 39
Perguntas frequentes 39
Instalação de cluster Incorporado 41
Conceitos básicos 41
Requisitos 41
Instalação padrão 42
Configuração do console de administração do KOTS 26
Requisitos de instalação adicionais 45
Histórico do documentos 49
..... I

O que é o Wickr Enterprise?

O Wickr Enterprise é um serviço end-to-end criptografado e auto-hospedado que ajuda organizações e agências governamentais a se comunicarem com segurança por meio one-to-one de mensagens em grupo, chamadas de voz e vídeo, compartilhamento de arquivos e compartilhamento de tela. Os clientes podem usar o Wickr Enterprise para superar as obrigações de retenção de dados associadas a aplicativos de mensagens para consumidores e facilitar a colaboração com segurança. Controles avançados de administração e segurança ajudam as organizações a atender aos requisitos legais e regulamentares e a criar soluções personalizadas para os desafios de segurança de dados.

As informações podem ser registradas em um armazenamento de dados privado controlado pelo cliente para fins de retenção e auditoria. Os clientes têm controle administrativo abrangente sobre os dados, o que inclui definir permissões, configurar opções de mensagens efêmeras e definir grupos de segurança. Os administradores também podem automatizar fluxos de trabalho com segurança com bots do Wickr. O Wickr Enterprise se integra a serviços adicionais, como o Active Directory e autenticação única (SSO) com OpenID Connect (OIDC). Para começar a configurar o Wickr Enterprise, consulte Introdução ao Wickr Enterprise.

> **Note**
>
> Se você ainda não tem o pacote de implantação do Wickr Enterprise, consulte o Fale conosco para questões comerciais.

Conceitos básicos do Wickr Enterprise

Tópicos

- Requisitos
- Instale as dependências.
- Configurar
- Bootstrap
- Implantar
- Gere KOTS Config

Requisitos

Antes de começar, verifique se os seguintes requisitos foram atendidos:

- Baixar o Node.js 16+
- AWS CLI configurado com credenciais para sua conta.

  Elas serão obtidas no seu arquivo de configuração em `~/.aws/config` ou usando as variáveis de ambiente `AWS_`.
- Instale o kubectl. Para obter mais informações, consulte Instalando ou atualizando o kubectl no Amazon EKSUser Guide.
- Instale a CLI do KOTS. Para obter mais informações, consulte Instalando a CLI do KOTS.
- Portas para a lista de permissões: 443/TCP para tráfego de chamadas HTTPS e TCP; 16384-19999/UDP para tráfego de chamadas UDP; TCP/8443

Arquitetura

Instale as dependências.

É possível adicionar todas as dependências ao pacote padrão com o seguinte comando:

```
npm install
```

Configurar

AWS Cloud Development Kit (AWS CDK) usa valores de contexto para controlar a configuração do aplicativo. O Wickr Enterprise usa valores de contexto do CDK para que você tenha controle sobre as configurações, como o nome de domínio da sua instalação do Wickr Enterprise ou o número de dias para reter os backups do RDS. Para obter mais informações, consulte Contexto do runtime no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK) .

Há várias maneiras de definir valores de contexto, mas recomendamos editar os valores em `cdk.context.json` de acordo com seu caso de uso específico. Somente os valores de contexto que começam com `wickr/` estão relacionados à implantação do Wickr Enterprise; o restante são valores de contexto específicos do CDK. Para manter as mesmas configurações na próxima vez que você fizer uma atualização por meio do CDK, salve esse arquivo.

No mínimo, você deve definir `wickr/licensePathwickr/domainName`, `wickr/route53:hostedZoneId` e `wickr/acm:certificateArn` ou `wickr/route53:hostedZoneName` e.

Com uma zona hospedada pública

Se você tiver uma zona hospedada pública do Route 53 em sua Conta da AWS, recomendamos usar as seguintes configurações para configurar seu contexto de CDK:

- `wickr/domainName`: o nome de domínio a ser usado para essa implantação do Wickr Enterprise. Se você usar uma zona hospedada pública do Route 53, os registros DNS e certificados ACM para esse nome de domínio serão criados automaticamente.
- `wickr/route53:hostedZoneName`: nome da zona hospedada do Route 53 na qual criar registros DNS.
- `wickr/route53:hostedZoneId`: ID da zona hospedada do Route 53 na qual criar registros DNS.

Esse método cria um certificado do ACM em seu nome, junto com os registros DNS que apontam seu nome de domínio para o balanceador de carga na frente da implantação do Wickr Enterprise.

Sem uma zona hospedada pública

Se você não tiver uma zona hospedada pública do Route 53 em sua conta, um certificado do ACM deverá ser criado manualmente e importado para o CDK usando o valor de contexto `wickr/acm:certificateArn`.

- `wickr/domainName`: o nome de domínio a ser usado para essa implantação do Wickr Enterprise. Se você usar uma zona hospedada pública do Route 53, os registros DNS e certificados ACM para esse nome de domínio serão criados automaticamente.
- `wickr/acm:certificateArn` - o ARN de um certificado do ACM a ser usado no balanceador de carga. Esse valor deve ser informado se não houver nenhuma zona hospedada pública do Route 53 disponível em sua conta.

Importando um certificado para o ACM

É possível importar um certificado obtido externamente com o seguinte 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
```

A saída será o ARN do certificado, que deve ser usado para o valor da configuração de contexto `wickr/acm:certificateArn`. É importante que o certificado carregado seja válido para o `wickr/domainName`, caso contrário, as conexões HTTPS não poderão ser validadas. Para obter mais informações, consulte Importando um certificado no Manual do usuário do AWS Certificate Manager .

Criar registros DNS

Como não há uma zona hospedada pública disponível, os registros DNS devem ser criados manualmente após a conclusão da implantação para apontar para o balanceador de carga na frente da implantação do Wickr Enterprise.

Implantar em uma VPC existente

Se você precisar usar uma VPC existente, poderá usar uma. No entanto, o VPC deve ser configurado para atender às especificações necessárias para o EKS. Para obter mais informações, consulte Veja os requisitos de rede do Amazon EKS para VPC e sub-redes no Guia do usuário do Amazon EKS e garanta que a VPC a ser usada atenda a esses requisitos.

Além disso, é altamente recomendável garantir que você tenha VPC endpoints para os seguintes serviços:

- CLOUDWATCH
- CLOUDWATCH_LOGS

- EC2
- EC2_MENSAGENS
- ECR
- ECR_DOCKER
- BALANCEAMENTO DE CARGA ELÁSTICO
- KMS
- GERENCIADOR_DE_SEGREDOS
- SSM
- MENSAGENS_SSM

Para implantar recursos em uma VPC existente, defina os seguintes valores de contexto:

- `wickr/vpc:id`- O ID da VPC no qual implantar recursos (por exemplo, `vpc-412beef`).
- `wickr/vpc:cidr`- O IPv4 CIDR da VPC (`172.16.0.0/16`por exemplo).
- `wickr/vpc:publicSubnetIds`- Uma lista separada por vírgulas das sub-redes públicas na VPC. O Application Load Balancer e os nós de processamento EKS de chamada serão implantados nessas sub-redes (por exemplo, `subnet-6ce9941,subnet-1785141,subnet-2e7dc10`).
- `wickr/vpc:privateSubnetIds`- Uma lista separada por vírgulas de sub-redes privadas na VPC. Os nós de processamento do EKS e o servidor bastion serão implantados nessas sub-redes (por exemplo, `subnet-f448ea8,subnet-3eb0da4,subnet-ad800b5`).
- `wickr/vpc:isolatedSubnetIds`- Uma lista separada por vírgulas das sub-redes isoladas na VPC. O banco de dados do RDS será implantado nessas sub-redes (por exemplo, `subnet-d1273a2,subnet-33504ae,subnet-0bc83ac`).
- `wickr/vpc:availabilityZones`- Uma lista separada por vírgulas das zonas de disponibilidade para as sub-redes na VPC (por exemplo, `us-east-1a,us-east-1b,us-east-1c`).

Para obter mais informações sobre endpoints VPC de interface, consulte Acessar um AWS serviço usando um endpoint de VPC de interface.

Outras configurações

Para obter mais informações, consulte Valores de contexto.

Bootstrap

Se esta é a primeira vez que você usa o CDK nessa região específica Conta da AWS , você deve primeiro inicializar a conta para começar a usar o CDK.

```
npx cdk bootstrap
```

Implantar

Esse processo levará cerca de 45 minutos.

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

Depois de concluída, a infraestrutura foi criada e você pode começar a instalar o Wickr Enterprise.

Criar registros DNS

Essa etapa não é necessária se você tiver usado uma zona hospedada pública ao configurar o CDK.

A saída do processo de implantação incluirá um valor `WickrAlb.AlbDnsName`, que é o nome DNS do balanceador de carga. A saída será semelhante a:

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

Nesse caso, o nome DNS é `Wickr-Alb-1Q5IBPJR4ZVZR-409483305.us-west-2.elb.amazonaws.com`. Esse é o valor que deve ser usado ao criar um registro CNAME ou A/AAAA (ALIAS) para seu nome de domínio.

Se você não tiver a saída da implantação, execute um dos comandos a seguir para exibir o nome DNS do balanceador de carga:

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

Gere KOTS Config

> **Warning**
>
> Esse arquivo contém informações confidenciais sobre sua instalação. Não compartilhe nem salve publicamente.

O instalador do Wickr Enterprise requer vários valores de configuração relativos à infraestrutura para ser instalado com êxito. Você pode usar um script auxiliar para gerar os valores das configurações.

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

Se você importou um certificado externo para o ACM na primeira etapa, passe o sinalizador `--ca-file` para esse script, por exemplo:

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

Se você receber um erro dizendo que a pilha não existe, defina a variável de ambiente `AWS_REGION` (`export AWS_REGION=us-west-2`) para a região selecionada e tente novamente. Ou, se você definir o valor do contexto`wickr/stackSuffix`, passe o sufixo com o `--stack-suffix` sinalizador.

Conectar-se ao cluster Kubernetes

A API do Amazon EKS só pode ser acessada por meio de um bastion host criado como parte da implantação. Como resultado, todos os comandos `kubectl` devem ser executados no próprio bastion host ou ser enviados por proxy por meio do bastion host.

Conexões de proxy através do bastion

Na primeira vez em que você se conectar ao cluster, atualize seu arquivo kubeconfig local usando o comando `aws eks update-kubeconfig` e, em seguida, defina o `proxy-url` em sua configuração. Depois, sempre que quiser se conectar ao cluster, inicie uma sessão SSM com o bastion host para encaminhar ao proxy para acesso à API.

Configuração única

Há um valor de saída na `WickrEks` CloudFormation pilha com um nome que começa com`WickrEnterpriseConfigCommand`. O valor contém o comando completo necessário para gerar a configuração kubectl para seu cluster. Essa entrada pode ser exibida com o seguinte comando:

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

Isso deve gerar um comando que comece com `aws eks update-kubeconfig`. Execute este comando.

Em seguida, a configuração do Kubernetes deve ser modificada para solicitações de proxy por meio do bastion host. Faça isso usando o seguinte comando:

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

Se funcionou corretamente, você verá uma saída como `'Property "clusters.arn:aws:eks:us-west-2:012345678912:cluster/WickrEnterprise5B8BF472-1234a41c4ec48b7b615c6789d93dcce.proxy-url" set.'`

Porta para frente até o bastion

Para se conectar ao cluster Amazon EKS, é necessário iniciar uma sessão de SSM para transferir solicitações de encaminhamento para o proxy em execução no seu bastion host. O comando para fazer isso é fornecido como saída `BastionSSMProxyEKSCommand` na pilha `WickrEks`. Execute o comando a seguir para visualizar o valor da saída:

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

O comando que ele emitirá começará com `aws ssm start-session`. Execute esse comando para iniciar um proxy local em execução na porta 8888 por meio da qual você pode se conectar ao cluster Amazon EKS. Se o encaminhamento da porta funcionou corretamente, a saída deverá dizer 'Aguardando conexões... '. Mantenha esse processo em execução durante todo o tempo necessário para acessar o cluster do Amazon EKS.

Se tudo estiver configurado corretamente, você poderá executar `kubectl get nodes` em outro terminal para listar os nós de trabalho no cluster Amazon EKS:

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

Instalar o Wickr Enterprise

Depois que sua conexão com o cluster Kubernetes for estabelecida, você poderá começar a instalar o Wickr Enterprise usando o suplemento `kubectl kots`. Você precisará do seu arquivo de licença do KOTS (um arquivo `.yaml` fornecido pelo Wickr) e do arquivo de valores de configuração, que foram salvos no arquivo `wickr-config.json` na seção Gerar configuração do KOTS. Para obter mais informações sobre Gerar configuração KOTS, consulte Gerar configuração KOTS.

Instalando o Wickr Enterprise manualmente

O comando a seguir iniciará a instalação do Wickr Enterprise:

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

Você será solicitado a digitar uma senha para o KOTS Admin Console. Salve essa senha porque você precisará dela para atualizar ou alterar a configuração da sua instalação do Wickr Enterprise no futuro.

Quando a instalação estiver concluída, `kubectl kots` abrirá uma porta local (geralmente `http://localhost:8080`), que fornece acesso ao KOTS Admin Console. Você pode alterar ou monitorar o status da instalação do Wickr Enterprise neste site ou começar a configurar o Wickr ao visitar o nome de domínio configurado para a instalação no navegador.

Instalando o Wickr Enterprise com o Lambda

Durante a implantação do CDK, um Lambda é criado e invocado para concluir automaticamente a instalação do Wickr Enterprise em seu nome. Para invocá-lo manualmente, abra o AWS console e encontre a função `WickrLambda-func*` lambda, na guia teste, selecione`test`, a entrada é irrelevante.

Pós-instalação

Há dois consoles web disponíveis para gerenciar sua instalação do Wickr Enterprise: o KOTS Admin Console e o Wickr Admin Console.

> **Note**
>
> Faça as alterações necessárias para refletir as políticas de backup e registro de sua organização (configurações do Amazon S3, registros de acesso do Elastic Load Balancing, registros de fluxo Amazon Virtual Private Cloud).

Console de administração do KOTS

Essa interface é usada para gerenciar a versão implantada do Wickr Enterprise. Você pode ver o status da instalação, modificar as configurações ou realizar atualizações. O KOTS Admin Console pode ser acessado somente por meio de um encaminhamento de porta Kubernetes, que pode ser aberto usando o seguinte comando:

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

> **Note**
>
> Primeiro, é necessário configurar sua conexão bastion conforme descrito na seção porta de encaminhamento para o bastion. Para obter mais informações sobre o encaminhamento de portas para o bastion, consulte Como fazer proxy de conexões por meio do bastion.

Quando a porta de encaminhamento for configurada com êxito, o comando anterior exibirá o seguinte:

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

Use o URL fornecido para acessar o KOTS Admin Console. A senha para fazer login é aquela que você escolheu ao executar `kubectl kots install` durante a instalação. Se você precisar redefinir sua senha, consulte Redefinir a senha do KOTS Admin Console.

Wickr Admin Console

Essa interface é usada para configurar sua instalação do Wickr Enterprise para configurar redes, usuários e federação. Ele pode ser acessado por HTTPS no nome DNS que você configurou para apontar para seu Load Balancer (Balanceador de carga). Se o DNS foi configurado automaticamente com uma zona hospedada pública, o nome do domínio é o valor do `wickr/domainName` contexto.

O nome de usuário padrão é `admin` e a senha é `Password123`. Você deverá alterar essa senha no primeiro login.

Valores do contexto

Os valores de contexto são pares de chave-valor que podem ser associados a um aplicativo, pilha ou estrutura. Eles podem ser fornecidos ao seu aplicativo a partir de um arquivo (geralmente `cdk.json` ou `cdk.context.json` no diretório do projeto) ou na linha de comando. O CDK usa valores de contexto para controlar a configuração do aplicativo. O Wickr Enterprise usa valores de contexto do CDK para que você tenha controle sobre as configurações, como o nome de domínio da sua instalação do Wickr Enterprise ou o número de dias para reter os backups do RDS.

Há várias maneiras de definir valores de contexto, mas recomendamos editar os valores em `cdk.context.json` de acordo com seu caso de uso específico. Somente os valores de contexto que começam com `wickr/` estão relacionados à implantação do Wickr Enterprise.

Nome	Descrição	Padrão
`wickr/licensePath`	O caminho para sua licença KOTS (um arquivo `.yaml` fornecido pelo Wickr).	nulo
`wickr/domainName`	: o nome de domínio a ser usado para essa implantaç ão do Wickr Enterprise. Se você estiver usando uma zona hospedada pública do Route 53, os registros de DNS e certificados do ACM para esse nome de domínio serão criados automaticamente.	nulo
`wickr/route53:host edZoneId`	: ID da zona hospedada do Route 53 na qual criar registros do DNS.	nulo
`wickr/route53:host edZoneName`	Nome da zona hospedada do Route 53 na qual criar registros do DNS.	nulo

Nome	Descrição	Padrão
`wickr/acm:certificateArn`	ARN de um certificado do ACM para usar no balanceador de carga. Esse valor deve ser informado se não houver nenhuma zona hospedada pública do Route 53 disponível em sua conta.	nulo
`wickr/caPath`	Caminho do certificado, necessário somente ao usar certificados autoassinados.	nulo
`wickr/vpc:id`	O ID do VPC em que o recurso será implantado. Necessário somente ao implantar em uma VPC existente. Se não estiver definido, uma nova VPC será criada.	nulo
`wickr/vpc:cidr`	IPv4 CIDR para associar à VPC criada. Se estiver implantando em uma VPC existente, defina isso como o CIDR da VPC existente.	172.16.0.0/16
`wickr/vpc:availabilityZones`	Lista separada por vírgulas das zonas de disponibilidade. Necessário somente ao implantar em uma VPC existente.	nulo

Nome	Descrição	Padrão
`wickr/vpc:publicSubnetIds`	Lista separada por vírgula da sub-rede pública. IDs Necessário somente ao implantar em uma VPC existente.	nulo
`wickr/vpc:privateSubnetIds`	Lista separada por vírgula de sub-rede privada. IDs Necessário somente ao implantar em uma VPC existente.	nulo
`wickr/vpc:isolatedSubnetIds`	Lista separada por vírgula de sub-rede isolada IDs para o banco de dados do RDS. Necessário somente ao implantar em uma VPC existente.	nulo
`wickr/rds:deletionProtection`	Habilite a proteção contra exclusão em instâncias do RDS.	true
`wickr/rds:removalPolicy`	Política de remoção para instâncias do RDS “snapshot”, “destruir” ou “reter”.	snapshot
`wickr/rds:readerCount`	Número de instâncias de leitura a serem criadas no cluster do RDS.	1
`wickr/rds:instanceType`	Tipo de instância a ser usado para instâncias do RDS.	r6g.xlarge
`wickr/rds:backupRetentionDays`	O número de dias para reter backups.	7

Nome	Descrição	Padrão
`wickr/eks:namespace`	Namespace padrão para serviços Wickr no EKS.	Wickr
`wickr/eks:defaultCapacity`	Número de nós de processamento do EKS para a infraestrutura de mensagens.	3
`wickr/eks:defaultCapacityCalling`	Número de nós de processamento do EKS para a infraestrutura de chamadas.	2
`wickr/eks:instanceTypes`	Lista separada por vírgulas dos tipos de instância a serem usados nos nós de processamento do EKS de mensagens.	m5.xlarge
`wickr/eks:instanceTypesCalling`	Lista separada por vírgulas dos tipos de instância a serem usados para chamar os nós de processamento do EKS.	c5n.large
`wickr/eks:enableAutoscaler`	Ativa a funcionalidade Cluster Autoscaler para EKS.	true
`wickr/s3:expireAfterDays`	Número de dias após os quais os uploads de arquivos serão removidos do bucket do S3.	1095
`wickr/eks:clusterVersion`	Versões de cluster, incluindo versão Kubernetes, versão KubectLayer, versão AlbController, versão e muito mais. nodeGroupRelease	1,27
`wickr/stackSuffix`	Um sufixo a ser aplicado aos nomes das CloudFormation pilhas.	"

Nome	Descrição	Padrão
`wickr/autoDeployWickr`	Implante automaticamente o aplicativo Wickr com lambda.	true

Como destruir recursos

Para excluir tudo o que foi criado por esse AWS CDK aplicativo, você deve excluir a `WickrRds` pilha antes de todas as outras pilhas.

Para que os recursos do Amazon RDS sejam excluídos adequadamente, a proteção contra exclusão deve ser desabilitada e a política de remoção deve ser definida como `snapshot` ou `destroy`. Se essas não forem as configurações atuais, modifique os valores `wickr/rds:deletionProtection` e `wickr/rds:removalPolicy` em seu contexto AWS CDK e reimplante a pilha do Amazon RDS executando `npx cdk deploy -e WickrRds`.

Depois que a política de proteção e remoção de exclusão estiver definida corretamente, execute `cdk destroy` para a pilha `WickrRds`:

```
npx cdk destroy WickrRds
```

Quando a `WickrRds` pilha terminar de ser destruída, as CloudFormation pilhas restantes poderão ser destruídas com o seguinte comando:

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

Solução de problemas

Excluir o namespace Wickr

Se você precisar excluir o namespace `wickr` para começar de novo, é importante primeiro fazer backup de todas as contas de serviço criadas pelo CDK dentro desse namespace. Essas contas de serviço permitem que os serviços do Wickr se comuniquem por AWS APIs meio de funções do IAM. Sem elas, tarefas como upload de arquivos por meio do Amazon Simple Storage Service (Amazon S3) não funcionarão mais.

Use o comando a seguir para fazer backup das contas de serviço e excluir e recriar o namespace `wickr` e as contas de serviço apropriadas:

```
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
```

Redefinir a senha do KOTS Admin Console

Você pode redefinir sua senha do KOTS Admin Console com o seguinte comando:

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

Ao alterar essa senha, você também pode querer atualizar o segredo do `wickr/kots` Secrets Manager, embora geralmente não seja usado novamente por nenhuma automação.

Problemas na conexão com o cluster EKS com o bastion

Se sua conexão com o cluster EKS por meio do bastion parecer lenta ou estiver expirando ocasionalmente, você poderá ver o seguinte erro ao executar `kubectl` comandos:

net/http: solicitação cancelada enquanto aguardava a conexão (Client.Timeout excedido enquanto aguardava os cabeçalhos)

Esse problema geralmente pode ser solucionado fazendo login no bastion host via SSM (veja o `BastionSSMCommand` na WickrEks pilha) e reiniciando o serviço: `tinyproxy`

```
sudo systemctl restart tinyproxy
```

Instalação personalizada

Na seção Instalação personalizada, você aprenderá como instalar o Wickr Enterprise.

Tópicos

- Requisitos
- Arquitetura
- Instalação
- Configurações de entrada
- Configurações do banco de dados
- Armazenamento de arquivos S3
- Configurações persistentes de reivindicação de volume
- Configurações do certificado TLS
- Configurações de chamada
- Autoescalador de cluster Kubernetes (opcional)
- Backups
- Instalação do Airgap
- Console de administração do Wickr
- Perguntas frequentes

Requisitos

Antes de começar a instalar o Wickr Enterprise, verifique se os seguintes requisitos foram atendidos.

Requisitos de hardware

O Wickr Enterprise requer um cluster Kubernetes para operar. É possível operar em um único nó com o Modo de poucos recursos ativado, mas isso não é recomendado para uso geral na produção. Em uma implantação de produção, recomendamos um mínimo de três nós de trabalho de mensagens, bem como um mínimo de dois nós de trabalho de chamada.

Um nó de trabalho deve ter as seguintes especificações mínimas.

- 2 a 4 núcleos de CPU

- 8 GB de RAM
- 200 GB de espaço em disco

Requisitos mínimos de hardware

Um cluster de um único nó de trabalho executado no modo de poucos recursos requer um mínimo de 3.000 m de CPU e 5846 milhões de RAM. Isso não inclui os pods do sistema kube.

Requisitos de recursos por pod

Nome do pod	Proprietário	CPU	Memória
API de administração	Wickr	100m	256 milhões
directory	Wickr	100m	128 mi
expirador	Wickr	100m	128 mi
proxy de arquivo	Wickr	100m	256 milhões
oidc	Wickr	100m	128 mi
opensearch	Wickr	500m	100 milhas
Orville	Wickr	50m	128 mi
orville-redis	Wickr	50m	128 mi
dispositivo push	Wickr	100m	128 mi
rabbitmq	Wickr	50m	256 milhões
reagir	Wickr	100m	64 milhas
recibos	Wickr	250 m	128 mi
redis	Wickr	50m	128 mi
API de servidor	Wickr	250 m	256 milhões
quadro de distribuição	Wickr	250 m	512 mi

Nome do pod	Proprietário	CPU	Memória
kotsadm	NÓS	50m	50 mi
kotsadm-mini	NÓS	100m	512 mi
kotsadm-rqlite	NÓS	200 m	1 Gi
minioperadora	S3 interno	200 m	256 milhões
miniinquilino	S3 interno	100m	256 milhões
mysql-primary	MySQL interno	100m	512 mi
mysql-secundário	MySQL interno	100m	512 mi

Requisitos de armazenamento

O Wickr Enterprise exige um padrão StorageClass para ser utilizado ao criar declarações de volume persistentes. Ao implantar em um ambiente isolado ou no local, talvez seja necessário configurar um para seu cluster. Uma opção disponível é o Longhorn. Os requisitos recomendados de espaço em disco variam com base no uso da opção Internal S3 e da opção Internal Mysql e na quantidade de espaço que você deseja ter disponível para upload de arquivos.

- Cache interno de imagens: ~ 60 GB
- RabbitMQ: 24 Gi Default/8 Gi no modo de poucos recursos
- Redis: 24 Gi Default/8 Gi no modo de poucos recursos
- OpenSearch: 24 Gi Default/8 Gi no modo de poucos recursos
- Mysql interno: 80 Gi Default/20Gi no modo de poucos recursos
- S3 interno: 160 Gi Default/2Gi no modo de poucos recursos
- KOTS Mini: 4 GB
- KOTS Ralite: 1 GB

Tamanho mínimo de armazenamento

- 377 Gi Default com S3 interno e Mysql interno
- 111 Gi no modo de poucos recursos

Requisitos de versão do Kubernetes

O Wickr Enterprise depende do KOTS replicado. A Replicated, uma plataforma comercial de distribuição de software, fornece uma lista das versões atualmente suportadas do Kubernetes. Para obter mais informações, consulte Compatibilidade de versão do Kubernetes.

Requisitos de software

O Wickr Enterprise requer um cluster Kubernetes e KOTS para operar. Consulte a documentação do KOTS para ver as versões compatíveis do sistema operacional e do Kubernetes. Para obter mais informações, consulte Requisitos mínimos do sistema.

Sistema Host para Desenvolvedores

Sistema operacional — Os comandos nesta documentação foram projetados para funcionar em Linux, macOS ou Windows com o WSL (Windows Subsystem for Linux) instalado.

Serviços internos de estado

O Wickr Enterprise pode fornecer serviços internos para banco de dados MySQL e armazenamento compatível com S3. No entanto, para uso geral de produção, é recomendável que você forneça esses serviços externamente ao cluster Kubernetes.

- Banco de dados MySQL 5.7
  - Banco de dados Amazon RDS MySQL 5.7 ou MySQL 5.7 (externo)
  - Gráfico do Mysql Bitnami Helm (interno)
  - Armazenamento de arquivos
    - Provedor de armazenamento compatível com Amazon S3 ou S3 (externo)
    - Gráfico do leme do operador Minio (interno)

Requisitos de rede

O Wickr Enterprise requer um FQDN, certificados SSL e portas TCP e UDP abertas específicas.

- FQDN: Um domínio ou subdomínio a ser usado pela implantação do Wickr Enterprise.
- Certificado SSL: um par de chaves de certificado SSL assinado por uma CA pública ou um par de chaves de certificado autoassinado. O certificado deve listar o FQDN no nome comum e também como uma entrada SAN DNS. O certificado também deve habilitar a extensão ServerAuth extendedKeyUsage .

- As instalações on-line precisarão de acesso de saída a recursos replicados e de terceiros. A Replicated mantém uma lista de seus endereços IP. Para obter mais informações, consulte Endereços IP replicados. O Replicated também mantém uma lista dos recursos de terceiros necessários. Para obter mais informações, consulte Aberturas de firewall para instalações online.
- Instalações isoladas exigem acesso a um registro privado de contêineres.

Nodos de mensagens

Os nós de mensagens não exigem um IPV4 endereço público e devem estar localizados em uma sub-rede privada. O tráfego de mensagens entrará no cluster por meio do LoadBalancer ou Ingress.

Chamando nós

Os nós de chamada exigem um IPV4 endereço público, portanto, devem estar em uma sub-rede pública. A mídia da chamada é transferida via UDP por padrão. Quando a chamada TCP estiver habilitada, o Proxy TCP aceitará conexões no TCP 443 e as enviará por proxy para o serviço Orville.

- TCP: 443 Chamando o proxy TCP
- UDP: 16384-16484 streams Audio/Video

Acesso à instalação e configuração

O acesso ao KOTS Admin Console para instalação e configuração é feito por meio de um encaminhamento de porta do Kubernetes.

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

Requisitos de licença

A instalação exigirá um arquivo de licença no formato.yaml, que será fornecido a você pelo Wickr Support.

Arquitetura

Arquitetura de produção recomendada

O diagrama abaixo mostra o Wickr Enterprise configurado conforme recomendado para produção, com os serviços MySQL e Object Storage situados fora do cluster Kubernetes.

Arquitetura interna ou de teste

O diagrama abaixo mostra a configuração do Wickr Enterprise, utilizando os serviços internos de MYSQL e Object Storage. Embora possa satisfazer as necessidades específicas de determinadas implantações, não é recomendado para uso geral na produção.

Instalação

1. Instale o kubectl e a CLI do kots.
2. Conecte-se ao cluster Kubernetes.
3. Obtenha o arquivo de licença do Wickr Enterprise do Wickr Support.
4. Instale o Wickr Enterprise usando o comando a seguir.

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

> **Note**
>
> license.yaml representa o arquivo de licença fornecido.

Após a instalação inicial, o KOTS Admin Console fornecerá opções de gerenciamento e configuração em nível de cluster.

Console de administração do KOTS

Essa interface é usada para gerenciar a versão implantada do Wickr Enterprise. Você pode ver o status da instalação, modificar configurações ou realizar atualizações do Wickr Enterprise. O KOTS Admin Console pode ser acessado somente por meio de um encaminhamento de porta Kubernetes, que pode ser aberto usando o seguinte comando:

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

Configurações de entrada

Controlador de entrada

O Wickr Enterprise suporta quatro tipos de controladores de entrada:

- LoadBalancer (Padrão)
  - O objeto loadbalancer pode exigir configuração explícita em instalações totalmente locais, embora geralmente seja fornecido por provedores de nuvem.
  - Implanta o serviço do controlador de entrada (ingress-nginx) com o tipo de serviço. LoadBalancer Isso exige que o cluster Kubernetes esteja sendo executado em uma plataforma que ofereça suporte a balanceadores de carga externos.
- ALB existente
  - Conecta o controlador de entrada a um ALB existente.
  - Você precisará fornecer o ARN existente do grupo alvo do Application Load Balancer.
- NodePort
  - O controlador de entrada (ingress-nginx) será configurado para usar o tipo de NodePort serviço, que abre uma porta em todos os nós do cluster Kubernetes e encaminha o tráfego para a entrada. O tráfego do cliente pode então ser direcionado para esses nós por meio do DNS ou de algum balanceador de carga externo.
  - Você pode escolher um intervalo de portas de 1 a 65535 ou uma porta aleatória de 30000 a 32767 será usada.
- Ingress
  - Traga seu próprio controlador de entrada. Essa configuração aceitará um nome de classe de entrada que os serviços usarão em seus manifestos de entrada. Isso implica que o controlador de entrada já tem alguma conectividade externa configurada por meio de algum outro mecanismo de balanceamento de carga.
  - Atualmente, somente o controlador ingress-nginx é suportado.

Nome de host Wildcard

Por padrão, as rotas do Ingress serão definidas com um valor de host de `*`. Desative essa configuração para usar o nome de host definido para o Wickr Enterprise Server. O nome de host curinga é necessário para nomes de host baseados em IP.

Configurações do banco de dados

O Wickr Enterprise requer um banco de dados MySQL 5.7. Recomendamos usar um banco de dados externo ao seu cluster Kubernetes, como o Amazon RDS, mas você também tem a opção de implantar um banco de dados MySQL interno dentro do cluster Kubernetes como parte da instalação.

Configurações do banco de dados externo

- Nome do host: nome do host ou endereço IP do servidor do banco de dados.
- Nome do host do leitor: nome do host ou endereço IP de um endpoint somente para leitura do servidor de banco de dados (se disponível).
- Porta: A porta na qual o MySQL será acessado.
- Nome do banco de dados: O nome do banco de dados criado no servidor.
- Nome de usuário: O usuário que tem permissões para acessar o banco de dados.
- Senha: A senha desse usuário.
- Certificado CA: um certificado PEM para conexão com o banco de dados via TLS.

> **Note**
>
> Certifique-se de que sua instalação do MySQL 5.7 esteja usando o conjunto de caracteres latin1 padrão com o agrupamento latin1_swedish_ci. Isso pode ser feito verificando se seu servidor MySQL foi iniciado com os seguintes sinalizadores: `"--character-set-server latin1", "--collation-server latin1_swedish_ci"`

Configurações internas do banco de dados

O tipo de banco de dados interno implantará dois StatefulSets em seu cluster para um MySQL primário e secundário com replicação binária. O secundário não recebe nenhum tráfego e está disponível somente para recuperação de desastres e backups.

Tamanho de armazenamento: tamanho (em gibibytes) dos volumes persistentes dos pods do banco de dados.

Aumentando o tamanho do armazenamento do MySQL

> **Note**
>
> O tipo de volume do seu StorageClass deve suportar a expansão do volume para aumentar o tamanho do armazenamento. Para obter mais informações, consulte Expansão de volume.

Os serviços MySQL usados no Wickr Enterprise são implantados como StatefulSet recursos no Kubernetes. StatefulSets torne imutáveis muitas propriedades do recurso, incluindo os modelos de Declaração de Volume Persistente. Como solução alternativa para a imutabilidade do StatefulSets, as seguintes ações devem ser executadas para aumentar o tamanho dos volumes usados pelo MySQL.

1. Edite as declarações de volume persistentes para `data-mysql-primary-0` `data-mysql-secondary-0` e.
   1. `kubectl -n wickr edit pvc data-mysql-primary-0.` Set `spec.resources.requests.storage`até o tamanho de armazenamento desejado.
   2. `kubectl -n wickr edit pvc data-mysql-secondary-0.` Set `spec.resources.requests.storage`até o tamanho de armazenamento desejado.
2. Exclua o existente StatefulSets, mas saia dos pods passando a `--cascade=orphan` bandeira.

   `kubectl -n wickr delete statefulset --cascade=orphan mysql-primary mysql-secondary`.
3. Na interface do usuário do KOTS, atualize a configuração do tamanho do armazenamento para corresponder ao valor definido na Etapa 1. Salve e implante essa configuração.
4. Reinicie o StatefulSets para expandir os volumes e colocar os serviços MySQL novamente on-line.

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

Armazenamento de arquivos S3

O Wickr Enterprise requer um serviço de armazenamento compatível com S3. Recomendamos usar um serviço S3 externo ao seu cluster Kubernetes, como o Amazon S3, mas você também tem a opção de implantar um serviço S3 interno dentro do cluster Kubernetes como parte da instalação.

Configurações externas do S3

- Nome do bucket: o nome do bucket do S3 em que os uploads de arquivos serão armazenados.
- Região: a AWS região do bucket do S3.
- Ponto final: defina o endpoint que o Wickr usará para interagir com a API do S3. O padrão é o endpoint de serviço S3 da região.
- Nome da conta do serviço Fileproxy: somente Amazon S3. O nome de uma conta de serviço Kubernetes existente a ser usada para autenticação no S3 usando funções do IAM para contas de serviço.
- Chave de acesso externa do S3: Esta é sua chave de acesso do S3 existente.
- Chave secreta externa do S3: Esta é a sua chave secreta do S3 existente.

Configurações internas do S3

O tipo S3 interno implantará um padrão de 4 pods de servidor MinIO, cada um contendo 4 declarações de volume persistentes. A configuração padrão utiliza a Codificação de Eliminação do MinIO para aumentar a tolerância a falhas.

- Contagem interna de servidores S3: o número de pods de servidores MinIO a serem criados, o padrão é 4 para uma implantação tolerante a falhas. Esse valor pode ser definido como tão baixo quanto 1 para uma development/test implantação.
- Contagem de volumes S3 internos: o número de volumes MinIO a serem criados em cada pod de servidor MinIO; o padrão é 4 para uma implantação tolerante a falhas. Esse valor pode ser definido como tão baixo quanto 1 para uma development/test implantação.
- Tamanho do volume S3 interno: o tamanho em GB dos volumes MinIO criados nos pods do servidor MinIO, o padrão é 10 GB.
- Uma implantação interna padrão do S3 usará 4 servidores com 4 PVCs. Cada PVC tem 10 Gi, produzindo 160 Gi de armazenamento bruto com 120 Gi de armazenamento codificado de eliminação disponível para os usuários.
- A calculadora Minio Erasure Coding está disponível. Para obter mais informações, consulte Calculadora de código de exclusão.

Configurações persistentes de reivindicação de volume

O Wickr Enterprise exige declarações de volume persistentes para armazenar dados com estado. Essa configuração permite que você especifique o nome da Classe de Armazenamento que você gostaria de usar. Se deixado em branco, o Wickr tentará usar a Classe de Armazenamento padrão. Não há suporte para alterar a classe de armazenamento após a implantação do Wickr.

Um padrão StorageClass para reivindicações de volume persistentes geralmente é fornecido por provedores de nuvem, no entanto, em instalações totalmente locais, pode ser necessária uma configuração explícita usando um serviço de terceiros, como o Longhorn.

Configurações do certificado TLS

Faça upload de um certificado PEM e uma chave privada para encerrar o TLS. O nome alternativo do assunto no certificado deve corresponder ao nome do host definido nas configurações da sua implantação do Wickr Enterprise.

Para o campo da cadeia de certificados, concatene todos os certificados intermediários (se necessário) com o certificado CA raiz antes de fazer o upload.

Let’s Encrypt

Selecione essa opção para gerar automaticamente um certificado usando o Let's Encrypt. Os certificados são emitidos usando o desafio HTTP-01 por meio do operador cert-manager.

O desafio HTTP-01 exige que o nome DNS desejado seja resolvido no ponto de entrada do seu cluster (geralmente um Load Balancer) e que o tráfego para a porta TCP 80 seja aberto ao público. Esses certificados duram pouco e serão renovados regularmente. É necessário manter a porta 80 aberta para permitir que os certificados sejam renovados automaticamente.

> **Note**
>
> Esta seção se refere explicitamente ao certificado usado pelo próprio aplicativo Wickr Enterprise.

Certificado fixado

O Wickr Enterprise exige a fixação de certificados ao usar certificados autoassinados ou certificados não confiáveis para dispositivos clientes. Se o certificado apresentado pelo seu Load Balancer for autoassinado ou assinado por uma CA diferente da instalação do Wickr Enterprise, carregue o certificado CA aqui para que os clientes o fixem.

Na maioria das situações, essa configuração não é necessária.

Provedores de certificados

Se você planeja comprar um certificado para uso com o Wickr Enterprise, veja abaixo uma lista de provedores cujos certificados funcionam corretamente por padrão. Se um provedor estiver listado abaixo, seus certificados foram validados explicitamente com o software.

- Digicert
- SSL rápido

Gerando um certificado autoassinado

Se você quiser criar seu próprio certificado autoassinado para uso com o Wickr Enterprise, o comando de exemplo abaixo contém todos os sinalizadores necessários para geração.

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

Se você quiser criar um certificado autoassinado baseado em IP, use o comando a seguir. Para usar o certificado baseado em IP, certifique-se de que o campo Wildcard Hostname esteja ativado nas configurações de entrada. Para obter mais informações, consulte Configurações de entrada.

```
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**
>
> Substitua $YOUR_DOMAIN no exemplo pelo nome de domínio ou endereço IP que você pretende usar.

Configurações de chamada

- Exigir nós de chamada: quando essa configuração está ativada, os serviços de chamada do Wickr são implantados somente nos nós do Kubernetes com o rótulo. `role=calling` Desative essa configuração para implantar serviços de chamadas e mensagens nos mesmos nós ou para implantações de um único nó.

  Geralmente, você também desejará desativar o Proxy TCP de chamada quando essa configuração estiver desativada, porque o serviço de Proxy TCP é executado na porta 443.
- Ativar proxy TCP: essa configuração controla se o serviço para o modo de fallback TCP nas chamadas é implantado ou não. Desative essa configuração se você tiver outros serviços em execução em 443/tcp ou não precisar do modo de fallback TCP para chamadas.
- Descubra automaticamente os endereços IP públicos do servidor: quando essa configuração estiver ativada, os serviços de chamada descobrirão seu endereço IP público fazendo solicitações HTTPS para https://ipv4.icanhazip.com/ https://ipv6.icanhazip.com/ e. Quando desativada, você deve ativar a configuração “Usar endereço IP primário do host para tráfego de chamada” ou “Substituição do nome do host”, caso contrário, os serviços de chamada não serão iniciados.
- Use o endereço IP primário do host para tráfego de chamadas: use o endereço IP primário dos nós do Kubernetes para chamar serviços. Isso significa que todos os clientes do Wickr podem se conectar aos seus nós do Kubernetes no endereço IP principal do nó, conforme apresentado na `status.hostIP` API Downward.
- Substituição de nome de host: forneça um nome de host ou endereço IP para retornar como ponto de conectividade para serviços de chamada. Essa configuração só deve ser usada ao executar um único servidor de chamada, porque o mesmo valor é retornado para todas as réplicas do serviço. Quando uma substituição de nome de host é definida e a configuração “usar endereço IP primário do host” está ativada, a configuração do endereço IP primário do host tem precedência.

Autoescalador de cluster Kubernetes (opcional)

O Kubernetes Cluster Autoscaler é um valor de configuração opcional para a instalação do Wickr Enterprise. Isso ajudará a escalar seus grupos de nós do Kubernetes no caso de aumento de tráfego ou outras restrições de recursos que possam levar a um desempenho ruim.

A instalação do Wickr Enterprise oferece suporte a três integrações de provedores de nuvem: AWS Google Cloud e Azure. Cada provedor de nuvem tem requisitos diferentes para essa integração. Siga as instruções abaixo para seu provedor de nuvem específico para ativar esse recurso.

AWS

Se você não usou o WickrEnterprise CDK para instalar seu ambiente Wickr AWS, precisará realizar algumas etapas adicionais para habilitar o autoescalador de cluster.

1. Adicione as seguintes tags aos seus grupos de nós. Isso permite que o autoescalador de cluster descubra automaticamente os nós apropriados.
    1. `k8s.io/cluster-autoscaler/`**`clusterName`** `= owned`onde clusterName é o nome do seu cluster Kubernetes
    2. `k8s.io/cluster-autoscaler-enabled = true`
2. Adicione uma conta de serviço do Kubernetes no namespace kube-system e associe-a a uma política do IAM que permite escalonamento automático e ações ec2. Para obter mais informações e instruções detalhadas, consulte Como configurar uma conta de serviço do Kubernetes para assumir uma função do IAM no Guia do usuário do Amazon EKS.
    1. Você precisará usar o namespace 'kube-system' ao configurar a conta de serviço
    2. A política a seguir pode ser usada para a Conta de Serviço:

       JSON

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

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

Na interface replicada, ao configurar o autoescalador de cluster, selecione AWScomo seu provedor de nuvem e forneça o nome da conta de serviço que você criou acima para instruir o autoescalador de cluster a utilizar essa conta de serviço.

Nuvem do Google

É altamente recomendável usar os recursos integrados de escalonamento automático do GKE tanto para o piloto automático quanto para os clusters padrão. No entanto, se você quiser continuar com essa integração, os seguintes requisitos devem ser atendidos antes de continuar.

Requisitos:

1. Os grupos de instâncias gerenciadas (MIG) devem ser criados com o escopo de segurança, incluindo no mínimo “leitura/gravação” nos recursos do Compute Engine. Atualmente, isso não pode ser adicionado ao MIG posteriormente.
2. O cluster deve ter a Federação de Identidade de Carga de Trabalho ativada. Você pode habilitar isso em um cluster existente executando: `gcloud container clusters update ${CLUSTER_NAME} --workload-pool=${PROJECT_ID}.svc.id.goog`
3. Uma conta de serviço do Google Cloud Platform (GCP) com acesso à função `roles/compute.instanceAdmin.v1`. Isso pode ser criado usando estas instruções:

```
# 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

O Azure Kubernetes Service (AKS) fornece escalonamento automático de cluster integrado para a maioria das implantações e é altamente recomendável utilizar esses métodos para escalonamento automático de clusters. No entanto, se seus requisitos forem tais que esses métodos não funcionem, fornecemos uma integração do Kubernetes Cluster Autoscaler para o Azure Kubernetes Service. Para utilizar essa integração, você precisará reunir as seguintes informações e colocá-las na configuração do painel de administração do KOTS em Cluster Autoscaler depois de selecionar o Azure como seu provedor de nuvem.

Autenticação do

ID da assinatura: A ID da assinatura pode ser obtida por meio do portal do Azure seguindo a documentação oficial. Para obter mais informações, consulte Obter assinatura e inquilino IDs no portal do Azure.

Os parâmetros a seguir podem ser obtidos criando um AD Service Principal usando o utilitário de linha de comando az.

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

ID do aplicativo:

Senha do cliente:

ID do inquilino:

Configuração do escalonador automático de cluster do Azure

Além dos requisitos de autenticação, os campos a seguir são necessários para o funcionamento adequado do autoescalador do cluster. Os comandos para obter essas informações foram fornecidos por conveniência, no entanto, eles podem exigir algumas modificações, dependendo da configuração específica do AKS.

Grupo de Recursos de Nó Gerenciado do Azure: Esse valor é o Grupo de Recursos Gerenciados criado pelo Azure quando você estabeleceu o Cluster AKS e não o Grupo de Recursos que você definiu. Para obter esse valor, você precisa do CLUSTER_NAME e RESOURCE_GROUP de quando você criou o cluster. Depois de ter esses valores, você pode obtê-los executando:

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

Nome do VMSS do Application Node Pool: Esse é o nome do Virtual Machine Scaling Set (VMSS) associado ao seu AKS Nodepool para o aplicativo Wickr. Esse é o recurso que será ampliado ou reduzido com base nas necessidades do seu cluster. Para obter esse valor, você pode executar o seguinte comando az:

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

ACalling Nome do VMSS do Node Pool (opcional): Esse é o nome do VMSS associado à sua chamada Nodepool, se você tiver um. Para obter esse valor, você pode executar uma versão modificada do comando para Application Node Pool VMSS Name trocando o valor CLUSTER_NODEPOOL_NAME pelo nome do nodepool de sua chamada nodepool.

Backups

O Wickr Enterprise utiliza o Velero para fins de Backup. O Velero fornece as ferramentas necessárias para fazer backup e restaurar recursos de cluster e volumes persistentes do Kubernetes, seja operando em um provedor de nuvem ou no local.

Backups do Velero com o Minio: Atualmente, os backups do Velero estão habilitados apenas para o Minio no modo de poucos recursos.

Instalação usando a documentação do Velero

- Instale o Velero CLI. Para obter mais informações, consulte Instalando a CLI do Velero.
- Instale o Velero em seu cluster e configure o armazenamento com base em seu provedor:
  - AWS.
  - GCP.
  - Azure.
  - Outros fornecedores.

Instalação do Airgap

Tanto o Wickr Enterprise quanto o KOTS oferecem suporte à implantação em um cluster Kubernetes totalmente isolado. Você deve fornecer acesso a um registro privado de imagens do Docker que possa ser acessado a partir do cluster Kubernetes isolado. O Private Docker Image Registry fornecido ao KOTS deve ser protegido com username/password autenticação para funcionar corretamente para essa finalidade. O KOTS utilizará o Private Docker Image Registry para hospedar todas as imagens do Wickr Enterprise.

- Wickr Enterprise license.yaml com airgap ativado (entre em contato com a equipe de vendas ou suporte ao cliente da Wickr)
- Pacote de arquivos Wickr Enterprise wickr.airgap (entre em contato com a equipe de vendas ou suporte ao cliente da Wickr)
- Acesso a um registro privado de imagens do Docker.
- Acesso a um cluster Kubernetes implantado no ambiente airgap.
- Kubectl instalado.
- KOTS CLI instalado.
- kotsadm.tar.gz baixado.

Execute os comandos a seguir para implantar o KOTS e o Wickr Enterprise em seu cluster kubernetes com airgap. Esses comandos carregam as imagens administrativas do KOTS e as imagens do Wickr Enterprise para o Private Docker Image Registry. Após a conclusão dos comandos, você será solicitado a acessar o KOTS Admin Console para concluir a instalação do Wickr Enterprise conforme descrito acima.

```
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
```

Notificação móvel para instalações de airgap

Listas adicionais de permissão de rede são necessárias para notificações push do back-end do servidor para clientes móveis. Esse requisito se deve à forma como o Apple iOS e o Google Android implementam esse recurso para dispositivos off-line e em segundo plano. Consulte a documentação desses serviços e liste as portas e endereços IP especificados.

- iOS
- Android

Console de administração do Wickr

A interface do Wickr Admin Console é usada para administrar o próprio aplicativo Wickr Enterprise. Ele pode ser usado para configurar redes, usuários, federação e muito mais. Ele pode ser acessado por HTTPS no nome DNS que você configurou para apontar para seu Load Balancer (Balanceador de carga). O nome de usuário padrão é admin, com a senha Password123. Você deverá alterar essa senha no primeiro login.

Perguntas frequentes

P: Minha implantação falha com o seguinte erro no helm stderr:

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

R: Isso pode acontecer quando o registro de depuração está ativado. Desative o registro de depuração, exclua os trabalhos problemáticos e tente novamente.

Cluster incorporado para Wickr Enterprise

A opção de instalação de cluster incorporado para o Wickr Enterprise fornece uma oferta de instalação pequena e eficiente para o produto Wickr Enterprise. Ele aproveita o cluster incorporado replicado para fornecer uma pequena instalação do Kubernetes usando k0s na qual o Wickr Enterprise pode ser instalado. O uso desse método de instalação minimiza os requisitos de habilidades técnicas, bem como os requisitos gerais de hardware para uma instalação do Wickr Enterprise, fornecendo uma solução “all-in-one” ao custo de resiliência e alta disponibilidade.

Tópicos

- Conceitos básicos do cluster Incorporado do Wickr Enterprise
- Requisitos de cluster incorporado do Wickr Enterprise
- Instalação do cluster incorporado Wickr Enterprise (padrão)
- Configuração do console de administração do KOTS
- Requisitos comuns adicionais de instalação

Conceitos básicos do cluster Incorporado do Wickr Enterprise

Para começar a usar a opção de cluster incorporado do Wickr Enterprise, entre em contato com o suporte para receber uma licença. Se você tiver uma licença existente e quiser utilizar essa opção, entre em contato com o suporte para obter ajuda na atualização da licença existente e instruções adicionais de instalação.

Requisitos de cluster incorporado do Wickr Enterprise

Antes de começar a instalar o cluster Incorporado do Wickr Enterprise, verifique se os requisitos a seguir são atendidos.

Requisitos de rede

Você precisará permitir a entrada no seu servidor Wickr nas seguintes portas:

- 443/TCP para tráfego de chamadas HTTPS e TCP
- 16384-19999/UDP para tráfego de chamadas UDP
- Somente LAN - 30000/TCP para acessar o console de administração do KOTS

Requisitos do sistema

Antes da instalação, verifique se você tem uma VM (máquina virtual) ou uma máquina física executando um sistema operacional (SO) baseado em Linux com os seguintes recursos mínimos disponíveis:

- 8 Núcleos de CPU
- 12 gigabytes (GB) de RAM
- 100 gigabytes (GB) de armazenamento em disco na partição/(raiz)

O cluster incorporado Wickr Enterprise foi testado nos seguintes sistemas operacionais Linux, mas outras opções de sistema operacional baseadas em Linux também podem ser adequadas:

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

Instalação do cluster incorporado Wickr Enterprise (padrão)

Depois de obter as instruções de download, baixe o pacote Wickr Enterprise na máquina de destino e descompacte-o.

```
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
```

Agora você deve ter dois arquivos, `wickr-enterprise-ha` `license.yaml` e. O `wickr-enterprise-ha` arquivo é um arquivo binário que inclui todas as peças necessárias para a instalação do Embedded Cluster, enquanto `license.yaml` é sua licença do Wickr que será usada para validar sua instalação.

Uma instalação básica pode ser executada nesse estágio executando o `wickr-enterprise-ha` arquivo:

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

Depois que o processo de instalação começar, você será solicitado a inserir uma senha do Admin Console. Insira uma senha segura e certifique-se de salvá-la conforme necessário ao acessar o KOTS Admin Console para continuar configurando sua instalação.

Depois que a instalação for concluída, a saída poderá ser semelhante ao seguinte:

```
   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
```

Após a instalação padrão, vá para o URL do console de administração do KOTS fornecido na saída usando um navegador da web. Neste exemplo, o URL é`http://192.168.1.100:30000`. No entanto, seu URL será diferente com base na sua configuração de rede.

Configuração do console de administração do KOTS

O console de administração do KOTS usa inicialmente um certificado autoassinado, que você precisará permitir como exceção em seu navegador. Depois de aceitar essa exceção, você será recebido pelo Assistente de Configuração do console administrativo do KOTS. Esse assistente orienta você nas etapas adicionais de configuração para configurar o comportamento do KOTS Admin Console, incluindo a opção de adicionar um certificado personalizado, se necessário.

Depois que a configuração inicial do console de administração do KOTS for concluída, você será solicitado a inserir a senha do console de administração que você criou durante o processo de instalação. No primeiro login, você precisa configurar o cluster.

Escolha Continuar para prosseguir para o console de administração do KOTS para Wickr.

> **Note**
>
> As instalações de vários nós estão atualmente na versão beta e o Wickr não as suporta.

Uma vez no console de administração do KOTS, configure sua instalação de acordo com suas necessidades. Ao utilizar a oferta de cluster incorporado, há algumas configurações importantes que devem ser definidas para garantir a funcionalidade adequada da instalação do Wickr Enterprise.

- Nome do host - Esse é o nome do host que você usa ao se comunicar com a instalação do Wickr. Certifique-se de criar registros DNS apropriados para esse domínio para apontar para sua instalação do Wickr Enterprise.
- Em Opções avançadas, marque a opção [] Configure Ingress Controller para expor um bloco de configuração para configurar o Kubernetes Ingress. No bloco de configuração do Ingress, selecione Single Node Embedded Cluster e, em seguida, insira o IP “público” associado ao seu servidor Wickr na caixa de texto chamada Loadbalancer External IP (Somente). IPv4

  Se não tiver certeza do que é esse IP, você pode executar o seguinte comando na linha de comando no servidor Wickr para determinar esse valor: `ip route get 1.1.1.1|awk '{print $7}'`
- Em Opções avançadas, marque a opção Ativar modo de poucos recursos.
- Em Chamada, verifique se a opção Exigir nós de chamada está desativada.
- Se você quiser uma solução completa que não utilize um banco de dados externo ou armazenamento compatível com S3 para compartilhamento de arquivos, selecione as opções internas para as seguintes configurações:
  - Banco de dados
  - Local de armazenamento S3

    O local de armazenamento interno do S3 fornece opções adicionais para configurar a capacidade de armazenamento. É recomendável começar aos poucos e expandir conforme necessário, pois reduzir a escala não é uma opção após o provisionamento.

Depois de configurar todos os recursos necessários, role até a parte inferior da página de configuração e escolha Salvar configuração. Isso iniciará algumas verificações prévias do anfitrião. Quando as verificações de pré-voo estiverem concluídas, escolha Deploy para iniciar a instalação do Wickr Enterprise.

Agora você está pronto para começar a configurar sua instalação do Wickr Enterprise. Para obter mais informações sobre como configurar o Wickr Enterprise, consulte O que é o Wickr Enterprise? .

Requisitos comuns adicionais de instalação

Instalações de nome de host IP

Se sua instalação exigir um nome de host baseado em IP, há algumas opções adicionais de configuração. Essas instruções são específicas para nomes de host baseados em IP, e é recomendável seguir as outras instruções para a configuração básica listadas acima.

No painel de administração do KOTS, conclua as etapas a seguir.

1. Defina o nome do host para o IP que você usará.
2. Em Certificados, selecione Carregar um certificado. Em seguida, gere um certificado autoassinado seguindo as instruções para um certificado baseado em IP. Para obter mais informações, consulte Gerar um certificado autoassinado.
3. Carregue o `.crt` arquivo para o certificado e o `.key` arquivo para a chave privada
4. Para a cadeia de certificados, faça o upload do `.crt` arquivo novamente.
5. Marque a caixa de seleção Definir um certificado fixo.
6. Faça o upload do `.crt` para o certificado fixado.
7. Em Chamada, desmarque as caixas de seleção Descobrir automaticamente endereços IP públicos do servidor e Usar endereço IP primário do host para tráfego de chamadas.
8. Em Chamada, coloque o endereço IP do nome do host na caixa de texto Substituição do nome do host.
9. Em Opções avançadas, marque a caixa de seleção Configurar controlador de entrada. Uma nova seção de configuração chamada Ingress aparece abaixo.
10. Em Entrada, selecione Cluster incorporado de nó único.
11. Em Entrada, insira o IP da interface 'pública' no servidor Wickr. Isso pode ser diferente do IP usado como nome de host. Veja informações adicionais sobre esse valor nas etapas básicas de configuração.
12. Em Entrada, marque Usar nome de host curinga.

SELinux Modo de imposição

Se você precisar usar SELinux no modo obrigatório, modifique o diretório de dados padrão usado para instalar o cluster incorporado. É recomendável usá-lo, `/opt` pois foi testado para funcionar com a maioria das SELinux políticas para esse caso de uso.

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

As verificações de pré-voo de instalação padrão dos clusters incorporados replicados tentarão validar se SELinux está no modo permissivo e falharão se estiver em Aplicação. SELinux Para contornar isso, é necessário usar o argumento da linha de `--ignore-host-preflights` comando. Ao usar a opção de linha de comando, há um prompt semelhante ao abaixo. Digite Sim quando solicitado.

```
#  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 instalações

A opção de instalação de cluster incorporado para o Wickr Enterprise oferece suporte a instalações isoladas. São necessárias configurações e habilitações adicionais para sua licença. Entre em contato com o suporte se você estiver interessado em usar o cluster incorporado do Wickr Enterprise em um ambiente isolado.

Ao realizar uma instalação de airgap, as instruções de download diferem do método de instalação padrão. Eles devem se parecer com os seguintes:

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

Faça o download do pacote para uma máquina que tenha acesso à Internet e, em seguida, transfira-o para seu ambiente isolado usando seu método de transporte de dados preferido. Depois que o pacote for transferido, extraia-o como você faria com qualquer pacote de instalação padrão. Um terceiro arquivo`wickr-enterprise-ha.airgap`, contendo todas as imagens do serviço do aplicativo Wickr Enterprise associadas, será incluído.

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

Durante a instalação, é necessário definir o argumento da linha de `--airgap-bundle` comando após a extração; caso contrário, o processo segue o procedimento de instalação padrão.

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

Atualização de um cluster incorporado AirGapped

Para atualizar um cluster AirGapped Incorporado, conclua as etapas a seguir.

1. Faça o download do novo pacote de cluster incorporado da Replicated e transfira-o para a máquina host usando seus métodos padrão de transferência de dados para seu ambiente isolado. Depois que o novo pacote estiver na máquina host, extraia o tarball:

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

2. Execute a atualização usando o novo pacote binário e airgap:

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

3. Inicie o console de administração do KOTS e faça login no URL fornecido usando seus métodos padrão de acesso ao console de administração do KOTS

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

4. Depois de fazer login no KOTS Admin Console, encontre a última atualização disponível à esquerda, em Versão, e pressione o botão Ir para o histórico da versão.
5. Escolha Implantar para a nova versão em Atualizações disponíveis. Percorra as telas:
    1. Altere as opções de configuração, role para baixo e escolha Avançar.
    2. Verifique se nenhuma verificação de pré-voo falhou, escolha Avançar: Confirmar e implantar.
    3. Escolha Implantar.

Notas adicionais sobre o cluster incorporado Wickr Enterprise

- NAMESPACE: Diferentemente da maioria das instalações do Wickr Enterprise, a instalação do cluster incorporado instala os ativos do Wickr no namespace kotsadm no kubernetes e não no wickr. Modifique todos os scripts ou comandos que você salvou e que use `-n wickr` para o kubectl, helm ou qualquer outro utilitário para usar em vez disso. `-n kotsadm`
- Interagindo com o cluster Kubernetes: na máquina host, use o `./wickr-enterprise-ha` binário para criar um shell com as variáveis apropriadas definidas para interagir com a instalação do Kubernetes em execução. ./wickr-enterprise-ha shell Isso fornecerá o utilitário kubectl dentro do PATH do shell e definirá a configuração apropriada do kube para a instalação local.

Histórico do documento

A tabela a seguir descreve as versões da documentação do Wickr Enterprise Automated Install Guide.

Alteração	Descrição	Data
Opções de implantação automática	Opções de implantação automática foram adicionadas. Para obter mais informações, consulte Instalando o Wickr Enterprise.	23 de fevereiro de 2024
Portas para a lista de permissões	A porta TCP/8443 foi adicionada à lista de permissões. Para obter mais informações, consulte Requisitos.	12 de fevereiro de 2024
Destruindo recursos e portas para a lista de permissões	Instruções sobre como destruir recursos foram adicionadas. Para obter mais informações, consulte Destruindo recursos. Além disso, portas à lista de permissões foram adicionad as. Para obter mais informaçõ es, consulte Requisitos.	17 de agosto de 2023
Lançamento inicial	Versão inicial do Guia de Instalação Automatizada do Wickr Enterprise	4 de agosto de 2023

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.