aws

自动安装指南

Wickr Enterprise

Wickr Enterprise: 自动安装指南

Table of Contents

Wickr Enterprise 是什么？

Wickr Enterprise 是一项 end-to-end加密的自托管服务，可帮助组织和政府机构通过 one-to-one群组消息、语音和视频通话、文件共享和屏幕共享进行安全通信。客户可以使用 Wickr Enterprise 来克服与消费级消息传递应用程序相关的数据留存义务，并安全地促进协作。先进的安全和管理控制措施可帮助组织满足法律和监管要求，并针对数据安全挑战构建定制解决方案。

可以将信息记录到客户控制的私有数据存储中，以便保留和审计。客户可以对数据进行全面的管理控制，包括设置权限、配置临时消息选项和定义安全组。管理员还可以使用 Wickr 机器人安全地自动执行工作流程。Wickr Enterprise 与其他服务集成，例如 Active Directory 和通过 OpenID Connect (OIDC) 进行单点登录（SSO）。要开始配置 Wickr Enterprise，请参阅 Wickr Enterprise 入门。

> **Note**
>
> 如果您还没有 Wickr Enterprise 部署包，请参阅联系我们进行业务咨询。

Wickr Enterprise 入门

主题

要求

在您开始之前，确认您满足以下要求：

- 下载 Node.js 16+
- AWS CLI 为您的账户配置了证书。

  它们将来自您的配置文件 `~/.aws/config` 或者使用 `AWS_` 环境变量。
- 安装 kubectl。有关更多信息，请参阅亚马逊 EKSUser 指南中的安装或更新 kubectl。
- 安装 kots CLI。有关更多信息，请参阅安装 kots CLI。
- 允许名单的端口：443/TCP 用于 HTTPS 和 TCP 呼叫流量；16384-19999/UDP 用于 UDP 呼叫流量；TCP/8443

架构

安装依赖项。

您可以使用以下命令将所有依赖项添加到默认软件包：

```
npm install
```

配置

AWS Cloud Development Kit (AWS CDK) 使用上下文值来控制应用程序的配置。Wickr Enterprise 使用 CDK 上下文值来控制一些设置，例如 Wickr Enterprise 安装的域名或保留 RDS 备份的天数。有关更多信息，请参阅AWS Cloud Development Kit (AWS CDK) 开发人员指南中的运行时上下文。

设置上下文值的方法有多种，但我们建议编辑 `cdk.context.json` 中的值以适应您的特定用例。只有以 `wickr/` 开头的上下文值与 Wickr Enterprise 部署相关；其余的则是 CDK 特定的上下文值。要在下次通过 CDK 进行更新时保持相同的设置，请保存此文件。

您必须至少设置`wickr/licensePath`、`wickr/domainName`、和`wickr/acm:certificateArn`或`wickr/route53:hostedZoneId`和`wickr/route53:hostedZoneName`。

使用公有托管区域

如果您的 Route 53 公共托管区域中有 Route 53 AWS 账户，我们建议您使用以下设置来配置 CDK 上下文：

- `wickr/domainName`：用于此 Wickr Enterprise 部署的域名。如果您使用 Route 53 公有托管区域，则将自动为该域名创建 DNS 记录和 ACM 凭证。
- `wickr/route53:hostedZoneName`：用于创建 DNS 记录的 Route 53 托管区域名称。
- `wickr/route53:hostedZoneId`：用于创建 DNS 记录的 Route 53 托管区域 ID。

此方法代表您创建 ACM 凭证，以及将您的域名指向 Wickr Enterprise 部署前的负载均衡器的 DNS 记录。

没有公有托管区域

如果您的账户中没有 Route 53 公有托管区域，则必须手动创建 ACM 凭证，然后使用 `wickr/acm:certificateArn` 上下文值将其导入 CDK。

- `wickr/domainName`：用于此 Wickr Enterprise 部署的域名。如果您使用 Route 53 公有托管区域，则将自动为该域名创建 DNS 记录和 ACM 凭证。
- `wickr/acm:certificateArn`：要在负载均衡器上使用的 ACM 凭证的 ARN。如果您的账户中没有 Route 53 公有托管区域，则必须提供此值。

将证书导入到 ACM。

您可以使用以下命令导入外部获得的凭证：

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

输出将是凭证 ARN，应将其用作 `wickr/acm:certificateArn` 上下文设置的值。上传的凭证必须对 `wickr/domainName` 有效，否则 HTTPS 连接将无法验证。有关更多信息，请参阅 AWS Certificate Manager 用户指南中的导入证书。

创建 DNS 记录

由于没有可用的公有托管区域，因此必须在部署完成后手动创建 DNS 记录，以指向 Wickr Enterprise 部署前的负载均衡器。

部署到现有 VPC

如果您需要使用现有 VPC，则可以使用现有 VPC。但是，必须将 VPC 配置为满足 EKS 所需的规范。有关更多信息，请参阅 Amazon EKS 用户指南中的查看 VPC 和子网的 Amazon EKS 联网要求，并确保要使用的 VPC 符合这些要求。

此外，强烈建议您确保拥有用于以下服务的 VPC 终端节点：

- 云观察
- 云监视日志
- EC2
- EC2_消息
- ECR
- ECR_DOCKER
- 弹性负载平衡
- KMS
- 秘密经理
- SSM
- SSM_MESSAGES

要将资源部署到现有 VPC，请设置以下上下文值：

- `wickr/vpc:id` - 要将资源部署到其中的 VPC ID（例如 `vpc-412beef`）。
- `wickr/vpc:cidr`-VPC 的 IPv4 CIDR（例如`172.16.0.0/16`）。
- `wickr/vpc:publicSubnetIds` - VPC 中以逗号分隔的公有子网列表。应用程序负载均衡器和调用 EKS Worker 节点将部署在这些子网中（例如 `subnet-6ce9941,subnet-1785141,subnet-2e7dc10`）。
- `wickr/vpc:privateSubnetIds` - VPC 中以逗号分隔的私有子网列表。EKS Worker 节点和堡垒机服务器将部署在这些子网中（例如 `subnet-f448ea8,subnet-3eb0da4,subnet-ad800b5`）。
- `wickr/vpc:isolatedSubnetIds` - VPC 中以逗号分隔的隔离子网列表。RDS 数据库将部署在这些子网中（例如 `subnet-d1273a2,subnet-33504ae,subnet-0bc83ac`）。
- `wickr/vpc:availabilityZones` - VPC 中以逗号分隔的子网可用区列表（例如 `us-east-1a,us-east-1b,us-east-1c`）。

有关接口 VPC 终端节点的更多信息，请参阅使用接口 VPC 终端节点访问 AWS 服务。

其他设置

有关更多信息，请参阅上下文值。

引导

如果这是您第一次在此特定 AWS 账户 和地区使用 CDK，则必须先启动该帐户才能开始使用 CDK。

```
npx cdk bootstrap
```

部署

此过程大约需要 45 分钟。

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

完成后，基础架构已创建完毕，您可以开始安装 Wickr Enterprise 了。

创建 DNS 记录

如果您在配置 CDK 时使用公有托管区域，则不需要执行此步骤。

部署过程的输出将包含一个值 WickrAlb.AlbDnsName，即负载均衡器的 DNS 名称。输出如下所示：

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

在这种情况下，DNS 名为 Wickr-Alb-1Q5IBPJR4ZVZR-409483305.us-west-2.elb.amazonaws.com。这是为您的域名创建 CNAME 或 A/AAAA（别名）记录时应使用的值。

如果您没有部署输出，运行以下命令来显示负载均衡器 DNS 名称：

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

生成 KOTS Config

> **Warning**
>
> 此文件包含有关您的安装的敏感信息。请勿公开共享或保存。

Wickr Enterprise 安装程序需要一些有关基础架构的配置值才能成功安装。您可以使用帮助脚本来生成配置值。

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

如果您在第一步中将外部凭证导入 ACM，请将 --ca-file 标志传递给此脚本，例如：

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

如果您收到错误消息，提示堆栈不存在，请将 AWS_REGION 环境变量 (export AWS_REGION=us-west-2) 设置为所选区域，然后重试。或者，如果您设置了上下文值wickr/stackSuffix，请传递带有--stack-suffix标志的后缀。

连接到 Kubernetes 集群

Amazon EKS API 只能通过在部署过程中创建的堡垒主机进行访问。因此，所有 kubectl 命令都必须要么在堡垒主机上运行，要么通过堡垒主机进行代理。

通过堡垒机代理连接

首次连接到集群时，必须使用 aws eks update-kubeconfig 命令更新本地 kubeconfig 文件，然后在配置中进行 proxy-url 设置。然后，每次要连接到集群时，都要启动与堡垒主机的 SSM 会话，以转发到代理进行 API 访问。

一次性设置

WickrEks CloudFormation 堆栈上有一个名称以开头的输出值WickrEnterpriseConfigCommand。该值包含为您的集群生成 kubectl 配置所需的完整命令。输出可以通过以下命令查看：

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

这应该输出一个以 aws eks update-kubeconfig 开头的命令。运行以下命令。

接下来，必须将 Kubernetes 配置修改为通过堡垒主机的代理请求。可以使用下面的命令进行这项操作：

```
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
```

如果运行正常，您会看到类似 'Property "clusters.arn:aws:eks:us-west-2:012345678912:cluster/WickrEnterprise5B8BF472-1234a41c4ec48b7b615c6789d93dcce.proxy-url" set.' 的输出

端口转发到堡垒机

要连接到 Amazon EKS 集群，您必须启动 SSM 会话，将转发请求移植到堡垒主机上运行的代理。执行此操作的命令作为 WickrEks 堆栈上的输出 BastionSSMProxyEKSCommand 提供。运行以下命令查看输出值：

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

它输出的命令将以 aws ssm start-session 开头。运行此命令启动在端口 8888 上运行的本地代理，您可以通过该代理连接到 Amazon EKS 集群。如果端口转发工作正常，则输出应显示“正在等待连接...”。在您访问 Amazon EKS 集群所需的整个过程中，请保持此过程处于运行状态。

如果一切设置正确，您将能够kubectl get nodes在另一个终端中运行以列出 Amazon EKS 集群中的工作节点：

```
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
```

安装 Wickr Enterprise

与 Kubernetes 集群建立连接后，您可以使用 `kubectl kots` 插件开始安装 Wickr Enterprise。您需要 KOTS 许可文件（由 Wickr 提供的 `.yaml` 文件）和配置值文件，这些文件已保存到“生成 KOTS 配置”部分的文件 `wickr-config.json` 中。有关“生成 KOTS 配置”的更多信息，请参阅生成 KOTS 配置。

手动安装 Wickr 企业版

以下命令将开始安装 Wickr Enterprise：

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

系统将提示您输入 KOTS 管理控制台的密码。请保存此密码，因为您将来需要用它来升级或更改 Wickr Enterprise 安装的配置。

安装完成后，`kubectl kots` 将打开一个本地端口（通常为 `http://localhost:8080`），用于访问 KOTS 管理控制台。您可以在此站点更改或监视 Wickr Enterprise 安装的状态，也可以通过在浏览器中访问为安装配置的域名来开始设置 Wickr。

使用 Lambda 安装 Wickr Enterprise

在 CDK 部署期间，系统会创建并调用 Lambda 以自动代表您完成 Wickr Enterprise 的安装。要手动调用它，请打开 AWS 控制台并找到 `WickrLambda-func*` lambda 函数，在 “测试” 选项卡下`test`，选择 “输入无关紧要”。

安装后

有两个 Web 控制台可用于管理 Wickr Enterprise 的安装：KOTS 管理员控制台和 Wickr 管理员控制台。

> **Note**
>
> 进行任何必要的更改以反映贵组织的备份和日志记录策略（Amazon S3 设置、Elastic Load Balancing 访问日志、 Amazon Virtual Private Cloud 流日志）。

KOTS 管理员控制台

此接口用于管理 Wickr Enterprise 的已部署版本。您可以查看安装状态、修改配置或执行升级。KOTS 管理员控制台只能通过 Kubernetes 端口转发功能来访问，该端口可以使用以下命令打开：

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

> **Note**
>
> 您必须首先按照“端口转发到堡垒机”部分所述，设置堡垒机连接。有关端口转发到堡垒机的更多信息，请参阅通过堡垒机代理连接。

成功配置端口转发后，上一个命令将输出以下内容：

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

使用提供的 URL 访问 KOTS 管理员控制台。登录密码是您在安装过程中运行 `kubectl kots install` 时选择的密码。如果需要重置密码，请参阅重置 KOTS 管理员控制台密码。

Wickr 管理员控制台

此接口用于配置 Wickr Enterprise 安装以设置网络、用户和联合身份验证。它可以通过 HTTPS 访问，使用您配置为指向负载均衡器的 DNS 名称。如果使用公共托管区域自动配置 DNS，则域名就是 `wickr/domainName` 上下文值的值。

默认用户名为 admin，密码为 Password123。首次登录时，您需要更改此密码。

上下文值

上下文值是可以与应用程序、堆栈或构造相关联的键值对。它们可以从文件（通常位于项目目录中的 `cdk.json` 或 `cdk.context.json`）在命令行中提供给您的应用程序。CDK 使用上下文值来控制应用程序的配置。Wickr Enterprise 使用 CDK 上下文值来控制一些设置，例如 Wickr Enterprise 安装的域名或保留 RDS 备份的天数。

设置上下文值的方法有多种，但我们建议编辑 `cdk.context.json` 中的值以适应您的特定用例。只有以 `wickr/` 开头的上下文值才与 Wickr Enterprise 部署相关。

名称	描述	默认
`wickr/licensePath`	获取 KOTS 许可证的路径（Wickr 提供的 `.yaml` 文件）。	null
`wickr/domainName`	用于此 Wickr Enterprise 部署的域名。如果使用 Route 53 公共 托管区，将自动为该域名创建 DNS 记录和 ACM 证书。	null
`wickr/route53:hostedZoneId`	将在其中创建 DNS 记录的 Route 53 托管区 ID。	null
`wickr/route53:hostedZoneName`	将在其中创建 DNS 记录的 Route 53 托管区名称。	null
`wickr/acm:certificateArn`	将在负载均衡器上使用的 ACM 证书的 ARN。如果您的账户中没有 Route 53 公共托管区，则必须提供此值。	null
`wickr/caPath`	证书路径，仅在使用自签名证书时才需要。	null
`wickr/vpc:id`	要将资源部署到其中的 VPC ID。仅在部署到现有 VPC 时才需要。如果未设置，则将创建一个新的 VPC。	null

名称	描述	默认
wickr/vpc:cidr	IPv4 要与已创建的 VPC 关联的 CIDR。如果部署到现有 VPC，则将其设置为现有 VPC 的 CIDR。	172.16.0.0/16
wickr/vpc:availabilityZones	以逗号分隔的可用区列表。仅在部署到现有 VPC 时才需要。	null
wickr/vpc:publicSubnetIds	以逗号分隔的公有子网列表。IDs仅在部署到现有 VPC 时才需要。	null
wickr/vpc:privateSubnetIds	以逗号分隔的私有子网列表。IDs仅在部署到现有 VPC 时才需要。	null
wickr/vpc:isolatedSubnetIds	以逗号分隔的 RDS 数据库隔离子网 IDs 列表。仅在部署到现有 VPC 时才需要。	null
wickr/rds:deletionProtection	对 RDS 实例启用删除保护。	true
wickr/rds:removalPolicy	RDS 实例“snapshot”、“destroy”或“retain”的删除策略。	快照
wickr/rds:readerCount	要在 RDS 集群中创建的读取器实例的数量。	1
wickr/rds:instanceType	用于 RDS 实例的实例类型。	r6g.xlarge
wickr/rds:backupRetentionDays	保留备份的天数。	7
wickr/eks:namespace	EKS 中 Wickr 服务的默认命名空间。	wickr

名称	描述	默认
wickr/eks:defaultCapacity	消息传递基础架构用的 EKS 工作节点的数量。	3
wickr/eks:defaultCapacityCalling	呼叫基础设架用的 EKS 工作节点的数量。	2
wickr/eks:instanceTypes	用于消息传递 EKS 工作节点的实例类型的逗号分隔列表。	m5.xlarge
wickr/eks:instanceTypesCalling	用于呼叫 EKS 工作节点的实例类型的逗号分隔列表。	c5n.large
wickr/eks:enableAutoscaler	切换启用 EKS 的 Cluster Autoscaler 功能。	true
wickr/s3:expireAfterDays	此后文件上传将从 S3 存储桶移除的天数。	1095
wickr/eks:clusterVersion	集群版本，包括 Kubernetes 版本、KubectlLayer 版本、AlbController 版本、版本等。 nodeGroupRelease	1.27
wickr/stackSuffix	适用于 CloudFormation 堆栈名称的后缀。	"
wickr/autoDeployWickr	使用 lambda 自动部署 Wickr 应用程序。	true

摧毁资源

要删除此 AWS CDK 应用程序创建的所有内容，必须先删除WickrRds堆栈，然后再删除所有其他堆栈。

为了正确删除 Amazon RDS 资源，必须禁用删除保护，并且必须将删除策略设置为 snapshot 或 destroy。如果这些不是当前设置，请在您的 AWS CDK 上下文中修改 wickr/rds:deletionProtection 和 wickr/rds:removalPolicy 值，然后通过运行 npx cdk deploy -e WickrRds 重新部署 Amazon RDS 堆栈。

正确设置删除保护和删除策略后，对 WickrRds 堆栈运行 cdk destroy：

```
npx cdk destroy WickrRds
```

WickrRds堆栈销毁完毕后，可以使用以下命令销毁剩余的 CloudFormation 堆栈：

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

故障排除

删除 Wickr 命名空间

如果您需要删除 `wickr` 命名空间才能重新开始，请务必先备份 CDK 在该命名空间中创建的所有服务账户。这些服务账户允许 Wickr 服务 AWS APIs通过 IAM 角色与之通信。没有它们，诸如通过 Amazon Simple Storage Service (Amazon S3) 上传文件之类的任务将不再起作用。

使用以下命令备份服务账户，删除并重新创建 `wickr` 命名空间和相应的服务账户：

```
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
```

重置 KOTS 管理控制台密码

您可以使用以下命令重置 KOTS 管理控制台密码：

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

更改此密码时，您可能还需要更新 S `wickr/kots` ecrets Manager 密码，尽管任何自动化系统通常不会再次使用该密码。

使用堡垒连接到 EKS 集群时出现问题

如果您通过堡垒与 EKS 集群的连接似乎很慢或偶尔会超时，则在运行`kubectl`命令时可能会看到以下错误：

net/http：等待连接时请求已取消（等待标头时超出了 Client.Timeout）

这个问题通常可以通过通过 SSM 登录堡垒主机（参见 WickrEks 堆栈`BastionSSMCommand`上的）并重新启动服务来解决：`tinyproxy`

```
sudo systemctl restart tinyproxy
```

自定义安装

在 “自定义安装” 部分，您将学习如何安装 Wickr Enterprise。

主题

要求

在开始安装 Wickr Enterprise 之前，请确认满足以下要求。

硬件要求

Wickr Enterprise 需要一个 Kubernetes 集群才能运行。在启用了低资源模式的情况下，可以在单个节点上运行，但不建议将其用于一般生产用途。在生产部署中，我们建议至少有三个消息传递工作节点以及至少两个调用工作节点。

工作节点应具有以下最低规格。

- 2 到 4 个 CPU 内核

- 8 GB 的内存
- 200 GB 的磁盘空间

最低硬件要求

在低资源模式下运行的单个工作节点群集至少需要 3000m CPU 和 5846Mi 内存。这不包括 kube-system 吊舱。

按 Pod 划分的资源需求

Pod 名称	所有者	CPU	内存
admin-api	Wickr	100m	256Mi
directory	Wickr	100m	128 Mi
过期者	Wickr	100m	128 Mi
文件代理	Wickr	100m	256Mi
oidc	Wickr	100m	128 Mi
opensearch	Wickr	500 米	100Mi
奥维尔	Wickr	50m	128 Mi
orville-redis	Wickr	50m	128 Mi
推送设备	Wickr	100m	128 Mi
rabbitmq	Wickr	50m	256Mi
反应	Wickr	100m	64 英里
收据	Wickr	250 米	128 Mi
redis	Wickr	50m	128 Mi
服务器 api	Wickr	250 米	256Mi
总机	Wickr	250 米	512Mi

Pod 名称	所有者	CPU	内存
kotsadm	KOTS	50m	50Mi
kotsadm-minio	KOTS	100m	512Mi
kotsadm-rqlite	KOTS	200m	1Gi
minio 操作员	内部 S3	200m	256Mi
迷你租户	内部 S3	100m	256Mi
mysql 主要	内部 MySQL	100m	512Mi
mysql-secon	内部 MySQL	100m	512Mi

存储要求

Wickr Enterprise 要求在创建永久卷声明时使用默认 StorageClass 值。在气隙环境或本地部署时，可能需要为集群配置一个。一种可用的选项是 Longhorn。建议的磁盘空间要求将根据内部 S3 选项和内部 Mysql 选项的使用情况以及您希望可用于文件上传的空间量而有所不同。

- 内部图像缓存：~60 Gi
- RabbitMQ：低资源模式下默认 24 Gi /8 Gi
- Redis：低资源模式下默认 24 Gi /8 Gi
- OpenSearch: 低资源模式下默认 24 Gi /8 Gi
- 内部 Mysql：低资源模式下默认 80 Gi /20Gi
- 内部 S3：低资源模式下默认 160 Gi /2Gi
- KOTS Minio：4 Gi
- KOTS Railite：1 Gi

最小存储大小

- 377 Gi 默认，内部 S3 和内部 Mysql
- 资源不足模式下的 111 Gi

Kubernetes 版本要求

Wickr Enterprise 依赖复制的 KOTS。Replicated 是一个商业软件分发平台，它提供了当前支持的 Kubernetes 版本列表。有关更多信息，请参阅 Kubernetes 版本兼容性。

软件要求

Wickr Enterprise 需要 Kubernetes 集群和 KOTS 才能运行。有关支持的操作系统和 Kubernetes 版本，请参阅 KOTS 文档。有关更多信息，请参阅最低系统要求。

开发者主机系统

操作系统 — 本文档中的命令专为在安装了 WSL（适用于 Linux 的 Windows 子系统）的 Linux、macOS 或 Windows 上运行而设计。

内部状态服务

Wickr Enterprise 可以为 MySQL 数据库和 S3 兼容存储提供内部服务，但是对于一般生产用途，建议您从 Kubernetes 集群外部提供这些服务。

- MySQL 5.7 数据库
  - 亚马逊 RDS MySQL 5.7 或 MySQL 5.7 数据库（外部）
  - Mysql Bitnami Helm 图表（内部）
  - 文件存储
    - 兼容 Amazon S3 或 S3 的存储提供商（外部）
    - Minio 操作员头盔图（内部）

网络要求

Wickr Enterprise 需要 FQDN、SSL 证书以及特定的开放式 TCP 和 UDP 端口。

- FQDN：Wickr Enterprise 部署要使用的域或子域。
- SSL 证书：由公共 CA 签名的 SSL 证书密钥对或自签名证书密钥对。证书必须在公用名中列出 FQDN，也必须作为 SAN DNS 条目列出。证书还必须启用 ServerAuth 扩展。 extendedKeyUsage
- 在线安装需要对复制资源和第三方资源的出口访问权限。“已复制” 会保留其 IP 地址列表。有关更多信息，请参阅复制的 IP 地址。Replicated 还会保留所需的第三方资源列表。有关更多信息，请参阅联机安装的防火墙开口。
- Air-Gapped 安装需要访问私有容器注册表。

消息节点

消息节点不需要公有 IPV4 地址，应位于私有子网中。消息流量将通过 LoadBalancer 或 Ingress 进入集群。

调用节点

调用节点需要公有 IPV4 地址，因此它们必须位于公有子网中。默认情况下，呼叫媒体通过 UDP 传输。启用 TCP 呼叫后，TCP 代理将接受 TCP 443 上的连接，并将这些连接代理到 Orville 服务。

- TCP：443 正在调用 TCP 代理
- UDP：16384-16484 直播 Audio/Video

安装和配置访问权限

通过 Kubernetes 端口转发访问 KOTS 管理控制台进行安装和配置。

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

许可证要求

安装需要.yaml 格式的许可文件，该文件将由 Wickr Support 提供给您。

架构

推荐的制作架构

下图显示了按照建议为生产配置的 Wickr Enterprise，MySQL 和对象存储服务都位于 Kubernetes 集群之外。

内部架构或测试架构

下图显示了 Wickr Enterprise 的配置，使用内部 MYSQL 和对象存储服务。尽管它可以满足某些部署的特定需求，但不建议将其用于一般生产用途。

安装

1. 安装 kubectl 和 kot s CLI。

2. 连接到 Kubernetes 集群。
3. 从 Wickr Support 获取 Wickr 企业版许可证文件。
4. 使用以下命令安装 Wickr Enterprise。

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

> **Note**
>
> license.yaml 代表您提供的许可证文件。

初始安装后，KOTS 管理控制台将提供集群级别的管理和配置选项。

KOTS 管理员控制台

此接口用于管理 Wickr Enterprise 的已部署版本。您可以查看 Wickr Enterprise 的安装状态、修改配置或执行升级。KOTS 管理员控制台只能通过 Kubernetes 端口转发功能来访问，该端口可以使用以下命令打开：

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

入口设置

入口控制器

Wickr Enterprise 支持四种入口控制器类型：

- LoadBalancer（默认）
  - 尽管负载均衡器对象通常由云提供商提供，但在完全本地安装中可能需要显式配置。
  - 使用服务类型部署入口控制器 (ingress-nginx) 服务。 LoadBalancer 这要求 Kubernetes 集群在支持外部负载均衡器的平台上运行。
- 现有的 ALB
  - 将入口控制器连接到现有 ALB。
  - 您需要提供现有的 Application Load Balancer 目标组 ARN。

- NodePort
  - 入口控制器（ingress-nginx）将被配置为使用服务类型，该 NodePort 服务类型在 Kubernetes 集群中的所有节点上打开一个端口，并将流量转发到入口。然后，可以通过 DNS 或某些外部负载平衡器将客户端流量定向到这些节点。
  - 你可以选择一个介于 1-65535 之间的端口范围，也可以随机使用 30000-32767 之间的端口。
- 入口
  - 带上自己的入口控制器。此配置将接受一个入口类名，然后服务将在其 Ingress 清单中使用该名称。这意味着入口控制器已经通过其他负载平衡机制配置了一些外部连接。
  - 目前仅支持 ingress-ngin x 控制器。

通配符主机名

默认情况下，入口路由将使用主机值为 `*` 进行定义。禁用此设置可使用为 Wickr 企业服务器定义的主机名。基于 IP 的主机名需要使用通配符主机名。

数据库设置

Wickr Enterprise 需要 MySQL 5.7 数据库。我们建议使用 Kubernetes 集群外部的数据库，例如 Amazon RDS，但作为安装的一部分，您也可以选择在 Kubernetes 集群内部署 MyS Q L 内部 MySQL 数据库。

外部数据库设置

- 主机名：数据库服务器的主机名或 IP 地址。
- 读取器主机名：数据库服务器只读端点的主机名或 IP 地址（如果有）。
- 端口：用于访问 MySQL 的端口。
- 数据库名称：在服务器上创建的数据库的名称。
- 用户名：有权访问数据库的用户。
- 密码：该用户的密码。
- CA 证书：用于通过 TLS 连接到数据库的 PEM 证书。

> **Note**
>
> 确保你的 MySQL 5.7 安装使用带有 latin1_swedish_ci 排序规则的默认 latin1 字符集。这可以通过验证您的 MySQL 服务器是否使用以下标志启动来实现：

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

内部数据库设置

内部数据库类型将在您的集群中部署两个 StatefulSets 用于二进制复制的 MySQL 主数据库和辅助数据库。辅助服务器不接收任何流量，只能用于灾难恢复和备份。

存储大小：数据库 Pod 的永久卷的大小（以 Gibibytes 为单位）。

增加 MySQL 存储大小

> **Note**
>
> 您的卷类型 StorageClass 必须支持卷扩展，才能增加存储大小。有关更多信息，请参阅卷扩展。

Wickr Enterprise 中使用的 MySQL 服务作为 StatefulSet 资源部署在 Kubernetes 中。 StatefulSets 使资源的许多属性不可变，包括永久卷声明模板。作为不可变性的解决方法 StatefulSets，必须执行以下操作来增加 MySQL 使用的卷的大小。

1. 编辑`data-mysql-primary-0`和的永久卷声明`data-mysql-secondary-0`。
    1. `kubectl -n wickr edit pvc data-mysql-primary-0. Set spec.resources.requests.storage`到所需的存储大小。
    2. `kubectl -n wickr edit pvc data-mysql-secondary-0. Set spec.resources.requests.storage`到所需的存储大小。
2. 删除现有的 Pod StatefulSets，但通过传递`--cascade=orphan`标志来保留 Pod。

    ```
    kubectl -n wickr delete statefulset --cascade=orphan mysql-primary
    mysql-secondary.
    ```
3. 在 KOTS 用户界面中，更新存储大小设置以匹配您在步骤 1 中设置的值。保存并部署此配置。
4. 重启 StatefulSets 以扩展卷并使 MySQL 服务恢复联机。

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

S3 文件存储

Wickr Enterprise 需要兼容 S3 的存储服务。我们建议使用 Kubernetes 集群外部的 S3 服务，例如 Amazon S3，但作为安装的一部分，您也可以选择在 Kubernetes 集群内部署内部 S3 服务。

外部 S3 设置

- 存储桶名称：用于存储文件上传的 S3 存储桶的名称。
- 区域：S3 存储桶的 AWS 区域。
- 端点：设置 Wickr 用于与 S3 API 交互的终端节点。默认为该区域的 S3 服务终端节点。
- Fileproxy 服务账户名称：仅限 Amazon S3。现有 Kubernetes 服务账户的名称，用于使用服务账户的 IAM 角色向 S3 进行身份验证。
- 外部 S3 访问密钥：这是您现有的 S3 访问密钥。
- 外部 S3 密钥：这是您现有的 S3 密钥。

内部 S3 设置

内部 S3 类型将默认部署 4 个 MinIO 服务器 pod，每个容器包含 4 个永久卷声明。默认配置使用MinIO 的擦除编码来提高容错能力。

- 内部 S3 服务器计数：要创建的 MinIO 服务器 pod 的数量，容错部署的默认值为 4。对于 development/test 部署，可以将此值设置为低至 1。
- 内部 S3 卷计数：要在每个 MinIO 服务器容器中创建的 MinIO 卷数，容错部署的默认值为 4。对于 development/test 部署，可以将此值设置为低至 1。
- 内部 S3 卷大小：在 MinIO 服务器容器中创建的 MinIO 卷的大小（以 GB 为单位），默认值为 10GB。
- 默认的内部 S3 部署将使用 4 台服务器和 4 台服务器 PVCs。每个 PVC 都是 10 Gi，可产生 160 Gi 的原始存储空间，120 Gi 的擦除编码存储可供用户使用。
- Minio 擦除编码计算器可用。有关更多信息，请参阅擦除码计算器。

永久音量声明设置

Wickr Enterprise 需要永久卷声明才能存储有状态的数据。此设置允许您指定要使用的存储类的名称名称。如果留空，Wickr 将尝试使用默认存储类别。不支持在部署 Wickr 之后更改存储类别。

Persi StorageClass stent Volume Claims 的默认值通常由云提供商提供，但是在完全本地安装中，可能需要使用第三方服务（例如 Longhorn）进行显式配置。

TLS 证书设置

上传用于终止 TLS 的 PEM 证书和私钥。证书上的主题备用名称必须与您的 Wickr Enterprise 部署设置中配置的主机名相匹配。

对于证书链字段，在上传之前，将所有中间证书（如果需要）与根 CA 证书连接起来。

Let's Encrypt

选择此选项可使用 Let's E ncrypt 自动生成证书。证书是通过 HTTP-01 质询通过证书管理器操作员颁发的。

HTTP-01 挑战要求所需的 DNS 名称解析到集群的入口点（通常是 Load Balancer），并且向 TCP 端口 80 的流量向公众开放。这些证书的有效期很短，将定期续订。必须保持端口 80 处于打开状态，以允许证书自动续订。

> **Note**
>
> 本节明确提及 Wickr Enterprise 应用程序本身使用的证书。

已锁定证书

使用自签名证书或客户端设备不信任的证书时，Wickr Enterprise 需要锁定证书。如果您的 Load Balancer 提供的证书是自签名的，或者由不同于 Wickr Enterprise 安装的 CA 签名，请在此处上传 CA 证书，让客户将其锁定。

在大多数情况下，此设置不是必需的。

证书提供商

如果您计划购买证书以用于 Wickr Enterprise，请参阅下文，了解已知默认情况下其证书可以正常运行的提供商列表。如果下面列出了提供商，则其证书已通过软件明确验证。

- Digicert

- RapidSSL

生成自签名证书

如果您想创建自己的自签名证书以用于 Wickr Enterprise，则以下示例命令包含生成所需的所有标志。

```
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"
```

如果要创建基于 IP 的自签名证书，请改用以下命令。要使用基于 IP 的证书，请确保在 Ingress 设置下启用 “通配符主机名” 字段。有关更多信息，请参阅 Ingress 设置。

```
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**
>
> 将示例中的 $YOUR_DOMAIN 替换为您要使用的域名或 IP 地址。

通话设置

- 需要调用节点：启用此设置后，Wickr 的调用服务仅部署在带有标签的 Kubernetes 节点上。`role=calling`禁用此设置可在相同节点上部署呼叫和消息服务，或者用于单节点部署。

  当禁用此设置时，您通常还需要禁用调用的 TCP 代理，因为 TCP 代理服务在端口 443 上运行。
- 启用 TCP 代理：此设置控制是否部署呼叫 TCP 回退模式的服务。如果您在 443/tcp 上运行其他服务，或者呼叫不需要 TCP 回退模式，请禁用此设置。
- 自动发现服务器公有 IP 地址：启用此设置后，呼叫服务将通过向https://ipv4.icanhazip.com/和发出 HTTPS 请求来发现其公有 IP 地址https://ipv6.icanhazip.com/。禁用后，必须启用 “使用主机主 IP 地址进行呼叫流量” 或 “主机名覆盖” 设置，否则呼叫服务将无法启动。
- 使用主机主 IP 地址进行呼叫流量：使用 Kubernetes 节点的主要 IP 地址来调用服务。这意味着所有 Wickr 客户端都能够通过节点的主要 IP 地址连接到你的 Kubernetes 节点，如向下 API `status.hostIP` 所示。

- 主机名覆盖：提供主机名或 IP 地址作为呼叫服务的连接点返回。只有在运行单个调用服务器时才应使用此设置，因为该服务的所有副本都会返回相同的值。如果设置了主机名覆盖并启用 “使用主机主 IP 地址” 设置，则优先考虑主机的主 IP 地址设置。

Kubernetes 集群自动扩缩器（可选）

Kubernetes 集群自动扩缩器是 Wickr Enterprise 安装的可选配置值。如果流量增加或其他资源限制可能导致性能不佳，它将有助于扩展 Kubernetes 节点组。

Wickr Enterprise 安装支持 3 种云提供商集成： AWS、谷歌云和 Azure。每个云提供商对这种集成都有不同的要求。请按照以下针对您的特定云提供商的说明启用此功能。

AWS

如果您没有使用 WickrEnterprise CDK 安装 Wickr 环境 AWS，则需要采取一些额外的步骤来启用集群自动扩缩程序。

1. 将以下标签添加到您的节点组。这允许集群自动扩缩程序自动发现相应的节点。
    1. `k8s.io/cluster-autoscaler/`**`clusterName`** `= owned`其中 clusterName 是你的 Kubernetes 集群的名称
    2. `k8s.io/cluster-autoscaler-enabled = true`
2. 在 kube-system 命名空间中添加一个 Kubernetes 服务账户，并将其与允许自动扩展和 ec2 操作的 IAM 策略相关联。有关更多信息和详细说明，请参阅 A mazon E KS 用户指南中的配置 Kubernetes 服务账户以担任 IAM 角色。
    1. 在设置服务账号时，你需要使用 “kube-system” 命名空间
    2. 以下政策可用于服务帐号：

       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"
        }
    ]
}
```

在配置集群自动扩缩器时，在 Replicated UI 中，选择AWS您的云提供商并提供您在上面创建的服务帐户的名称，以指示集群自动扩缩程序使用该服务帐户。

谷歌云

强烈建议将 GKE 的内置自动缩放功能用于自动驾驶和标准集群。但是，如果您想继续进行此集成，则必须满足以下要求才能继续。

要求：

1. 托管实例组 (MIG) 必须使用安全范围创建，包括至少 “读/写” Compute Engine 资源。以后无法将其添加到 MIG 中。
2. 集群必须启用工作负载身份联合。您可以通过运行以下命令在现有集群上启用此功能：`gcloud container clusters update ${CLUSTER_NAME} --workload-pool=${PROJECT_ID}.svc.id.goog`
3. 一个谷歌云平台 (GCP) 服务账户，可以访问 “roles/compute.instanceAdmin.v1” 角色。可以使用以下说明创建：

```
# 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 服务 (AKS) 为大多数部署提供集成的集群自动缩放，强烈建议使用这些方法进行集群自动缩放。但是，如果您的要求导致这些方法不起作用，我们为 Azure Kubernetes 服务提供了 Kubernetes 集群自动扩缩程序集成。要使用此集成，你需要收集以下信息，并在选择 Azure 作为云提供商后，将其放入 KOTS 管理面板的 Cluster Autoscaler 下的配置中。

天蓝色身份验证

订阅 ID：订阅 ID 可以通过官方文档通过 Azure 门户获取。有关更多信息，请参阅在 Azure 门户 IDs 中获取订阅和租户。

以下参数可以通过使用 az 命令行实用程序创建 AD 服务主体来获得。

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

应用程序 ID：

客户密码：

租户编号：

Azure 集群自动扩缩程序配置

除了身份验证要求外，群集自动扩缩器正常运行还需要以下字段。为方便起见，提供了获取此信息的命令，但是，根据您的特定 AKS 配置，可能需要对其进行一些修改。

Azure 托管节点资源组：此值是 AZure 在建立 AKS 集群时创建的托管资源组，而不是你定义的资源组。要获得此值，您需要创建集群时的 CLUSTER_NAME 和 RESOURCE_GROUP。一旦有了这些值，就可以通过运行以下命令来获得这些值：

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

应用程序节点池 VMSS 名称：这是与 Wickr 应用程序的 AKS 节点池关联的虚拟机扩展集 (VMSS) 的名称。该资源将根据您的集群需求向上或向下扩展。要获得此值，可以运行以下 az 命令：

```
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 节点池 VMSS 名称（可选）：这是与您的调用节点池关联的 VMSS 的名称（如果有）。要获得此值，您可以运行应用程序节点池 VMSS 名称命令的修改版本，将调用节点池的节点池名称的 CLUSTER_NODEPOOL_NAME 值移除。

备份

Wickr Enterprise 使用 Velero 进行备份。Velero 为备份和恢复 Kubernetes 集群资源和永久卷提供了必要的工具，无论是在云提供商还是在本地运行。

使用Minio进行Velero备份：目前，Velero备份仅在资源不足模式下为Minio启用。

使用 Velero 文档进行安装

- 安装 Velero CLI。有关更多信息，请参阅安装 Velero CLI。
- 在集群上安装 Velero 并根据您的提供商配置存储：
  - AWS.
  - GCP。
  - 天蓝色。
  - 其他提供商。

气隙安装

Wickr Enterprise 和 KOTS 都支持部署到完全隔开的 Kubernetes 集群中。你必须提供对私有 Docker 镜像注册表的访问权限，该注册表可以从隔空的 Kubernetes 集群中访问。为此，提供给 KOTS 的私有 Docker 镜像注册表必须通过 username/password 身份验证进行保护，才能正常运行。KOTS 将利用私有 Docker 镜像注册表来托管所有 Wickr Enterprise 镜像。

- 启用气隙的 Wickr Enterprise 许可证.yaml（联系 Wickr 销售人员或客户支持团队）

- Wickr Enterprise wickr.airgap 存档包（联系 Wickr 销售人员或客户支持团队）
- 访问私有 Docker 镜像注册表。
- 访问部署在气隙环境中的 Kubernetes 集群。
- Kubectl 安装好了。
- KOTS CLI 已安装。
- kotsadm.tar.gz 已下载。

运行以下命令将 KOTS 和 Wickr Enterprise 部署到你的 airgaped kubernetes 集群上。这些命令将 KOTS 管理镜像和 Wickr Enterprise 镜像上传到私有 Docker 镜像注册表。命令完成后，系统将提示您访问 KOTS 管理控制台以完成 Wickr Enterprise 的安装，如上所述。

```
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
```

airgap 安装的移动通知

要将通知从服务器后端推送到移动客户端，则需要其他网络允许列表。此要求是由于苹果 iOS 和谷歌安卓如何为离线和后台设备实现此功能。请参阅这些服务的文档，并允许列出指定的 IP 地址和端口。

- iOS
- Android

Wickr 管理员控制台

Wickr 管理控制台界面用于管理 Wickr Enterprise 应用程序本身。它可用于设置网络、用户、联盟等。它可以通过 HTTPS 访问，使用您配置为指向负载均衡器的 DNS 名称。默认用户名为管理员，密码为 Password123。首次登录时，您需要更改此密码。

常见问题解答

问：我的部署失败，在 helm stderr 中出现以下错误：

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

答：启用调试日志记录时可能会发生这种情况。请禁用调试日志记录，删除有问题的作业，然后重试。

适用于 Wickr Enterprise

Wickr Enterprise 的嵌入式集群安装选项为 Wickr Enterprise 产品提供了小型、高效的安装服务。它利用复制的嵌入式集群提供使用 k0 的小型 Kubernetes 安装，可以在上面安装 Wickr Enterprise。使用这种安装方法以牺牲弹性和高可用性为代价提供 “all-in-one” 解决方案，从而最大限度地降低了 Wickr Enterprise 安装的技术技能要求和总体硬件要求。

主题

Wickr Enterprise 嵌入式

要开始使用 Wickr Enterprise 嵌入式集群选项，请联系支持人员获取许可证。如果您已有许可证并想使用此选项，请联系支持人员以获取更新现有许可证的帮助以及其他安装说明。

Wickr 企业版嵌入式集群要求

在您开始安装 Wickr Enterprise 嵌入式（集群）之前，确认您满足以下要求。

网络要求

你需要允许通过以下端口进入你的 Wickr 服务器：

- 适用于 HTTPS 和 TCP 呼叫流量的 443/TC
- 16384-19999/UDP 适用于 UDP 呼叫流量
- 仅限局域网-30000/TCP 用于访问 KOTS 管理控制台

系统要求

安装之前，请确保您的虚拟机（虚拟机）或运行基于 Linux 的操作系统 (OS) 的物理机具有以下最低可用资源：

- 8 CPU 核心数
- 12 千兆字节 (GB) 的内存
- / (根) 分区上有 100 千兆字节 (GB) 的磁盘存储空间

Wickr Enterprise 嵌入式集群已在以下 Linux 操作系统上进行了测试，但其他基于 Linux 的操作系统选项也可能适合：

- Red Hat Enterpris
- Amazon Linux 2023
- Linux 9.5

安装 Wickr Enterprise 嵌入式集群（标准）

获得下载说明后，将 Wickr Enterprise 捆绑包下载到目标计算机并解压缩。

```
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
```

你现在应该有两个文件，`wickr-enterprise-ha`和`license.yaml`。该`wickr-enterprise-ha`文件是一个二进制文件，其中包含安装嵌入式集群的所有必要部分，而用于验证安装`license.yaml`的是您的 Wickr 许可证。

在此阶段可以通过执行以下`wickr-enterprise-ha`文件来执行基本安装：

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

安装过程开始后，系统会提示您输入管理员控制台密码。输入安全密码并确保将其保存，因为在访问 KOTS 管理控制台以继续配置安装时需要该密码。

安装完成后，输出类似于以下内容：

```
   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
```

标准安装完成后，使用 Web 浏览器前往输出中提供的 KOTS 管理控制台 URL。在本示例中，网址为`http://192.168.1.100:30000`。但是，根据您的网络配置，您的 URL 会有所不同。

KOTS 管理员控制台配置

KOTS 管理员控制台最初使用自签名证书，您需要在浏览器中允许该证书作为例外情况。接受此例外情况后，KOTS 管理员控制台的配置向导将欢迎您。此向导将指导您完成配置 KOTS 管理控制台行为的其他配置步骤，包括在必要时添加自定义证书的选项。

KOTS 管理控制台的初始配置完成后，系统会提示您输入在安装过程中创建的管理员控制台密码。首次登录时，您需要配置集群。

选择 “继续” 进入 Wickr 的 KOTS 管理控制台。

> **Note**
>
> 多节点安装目前处于测试阶段，Wickr 不支持它们。

进入 KOTS 管理员控制台后，根据需要配置您的安装。使用嵌入式集群产品时，应设置一些关键配置设置，以确保安装的 Wickr Enterprise 能够正常运行。

- 主机名-这是您在与 Wickr 安装进行通信时使用的主机名。请务必为此域创建相应的 DNS 记录，以指向您的 Wickr Enterprise 安装。
- 在 “高级选项” 下，选中 [] 配置入口控制器选项，以显示用于配置 Kubernetes Ingress 的配置块。在 Ingress 配置块中，选择单节点嵌入式集群，然后在标有 L oadbalancer 外部 IP（仅限）的文本框中输入与您的 Wickr 服务器关联的 “公共” IP。IPv4

如果您不确定此 IP 是什么，可以从 Wickr 服务器上的命令行运行以下命令来确定此值：`ip route get 1.1.1.1|awk '{print $7}'`

- 在“高级选项”下，选中“启用低资源模式”选项。
- 在“呼叫”下，确保禁用“需要呼叫节点”。
- 如果您想要一个不使用外部数据库或 S3 兼容存储进行文件共享的多合一解决方案，请为以下设置选择内部选项：
  - 数据库
  - S3 存储位置

    内部 S3 存储位置为配置存储容量提供了其他选项。建议从小规模开始，然后根据需要进行扩展，因为配置后不能选择缩小规模。

配置完所有必需的功能后，滚动到配置页面的底部，然后选择保存 Config。这将启动一些预检主机检查。印前检查完成后，选择“部署”开始安装 Wickr Enterprise。

现在，您已准备就绪，可开始配置 Wickr Enterprise 安装。有关配置 Wickr Enterprise 的更多信息，请参阅什么是 Wickr Enterpr ise？ 。

其他常见安装要求

IP 主机名安装

如果您的安装需要基于 IP 的主机名，则还有一些其他配置选项。这些说明专门针对基于 IP 的主机名，建议您按照上面列出的其他基本设置说明进行操作。

在 KOTS 管理面板中，完成以下步骤。

1. 将主机名设置为您要使用的 IP。
2. 在“证书”下，选择“上传证书”。然后，按照基于 IP 的证书的说明生成自签名证书。有关更多信息，请参阅生成自签名证书。
3. 上传证书`.crt`.`key`文件和私钥文件
4. 对于证书链，请再次上传`.crt`文件。
5. 选中“设置固定证书”复选框。
6. 上传已锁**`.crt`**定证书的。

7. 在 “呼叫” 下，取消选中 “自动发现服务器公有 IP 地址” 和 “使用主机主 IP 地址进行呼叫流量” 复选框。
8. 在 “呼叫” 下，在 “主机名覆盖” 文本框中输入主机名的 IP 地址。
9. 在 “高级选项” 下，选中 “配置入口控制器” 复选框。下面显示了一个名为 Ingress 的新配置部分。
10. 在 Ingress 下，选择单节点嵌入式集群。
11. 在 Ingress 下，输入 Wickr 服务器上 “公共” 接口的 IP。这可能与用作主机名的 IP 不同。在基本配置步骤中查看有关此值的更多信息。
12. 在 Ingress 下，选中使用通配符主机名。

SELinux 强制模式

如果您需要 SELinux 在强制模式下使用，请修改用于安装嵌入式集群的默认数据目录。建议使用，`/opt`因为它已经过测试，可以与该用例的大多数 SELinux 策略配合使用。

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

复制的嵌入式集群默认安装预检检查将尝试验证是否 SELinux 处于允许模式，如果处于强制模式，SELinux 则会失败。要绕过这个问题，需要使用`--ignore-host-preflights`命令行参数。使用命令行选项时，会出现类似于以下提示的提示。出现提示时输入 “是”。

```
#  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 装置

Wickr Enterprise 的嵌入式集群安装选项支持隔空安装。需要对许可证进行其他配置和启用。如果您有兴趣在离线环境中使用 Wickr Enterprise 嵌入式集群，请联系支持人员。

执行 airgap 安装时，下载说明与标准安装方法不同。它们应类似于以下内容：

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

将捆绑包下载到可以访问互联网的计算机上，然后使用您首选的数据传输方式将其传输到隔开的环境。传输捆绑包后，请像使用任何标准安装包一样将其解压缩。将包括第三个文件wickr-enterprise-ha.airgap，其中包含所有相关的 Wickr Enterprise 应用程序服务镜像。

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

在安装过程中，必须在解压后设置--airgap-bundle命令行参数；否则，该过程将遵循标准安装程序。

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

更新 AirGapped 嵌入式集群

要更新 AirGapped 嵌入式（集群），请完成以下步骤。

1. 从 Replicated 下载新的嵌入式集群软件包，然后使用适用于隔空环境的标准数据传输方法将其传输到主机。将新捆绑包放到主机上后，解压压缩包：

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

2. 使用新的二进制文件和 airgap 捆绑包运行更新：

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

3. 启动 KOTS 管理控制台，然后使用访问 KOTS 管理控制台的标准方法登录到提供的 URL

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

4. 登录 KOTS 管理控制台后，在左侧的 “版本” 下找到 “最新可用更新”，然后按 “转到版本历史记录” 按钮。
5. 在 “可用更新” 下为新版本选择 “部署”。浏览屏幕：

    1. 更改任何配置选项，向下滚动，然后选择 “下一步”。
    2. 验证没有失败的印前检查，选择下一步：确认并部署。
    3. 选择部署。

关于 Wickr Enterprise 嵌入式集群的其他说明

- 命名空间：与大多数 Wickr Enterprise 安装不同，嵌入式集群安装将 Wickr 资产安装到 kubernetes 而不是 wic kr 中的 kotsadm 命名空间。修改你保存的、用`-n wickr`于 kubectl、helm 或任何其他实用程序的所有脚本或命令，改为使用`-n kotsadm`。
- 与 Kubernetes 集群交互：在主机上，使用`./wickr-enterprise-ha`二进制文件创建一个 shell，设置了适当的变量，以便通过运行与 Kubernetes 安装进行交互。./wickr-enterprise-ha shell这将在外壳的 PATH 中提供 kubectl 实用程序，并将相应的 kube 配置设置为本地安装。

文档历史记录

下表描述了 Wickr Enterprise 自动安装指南的文档版本。

变更	说明	日期
自动部署选项	已添加自动部署选项。有关更多信息，请参阅安装 Wickr Enterpris e。	2024 年 2 月 23 日
允许列入许可名单的端口	端口 TCP/8443 已添加到许可名单中。有关更多信息，请参阅要求。	2024 年 2 月 12 日
销毁允许名单上的资源和端口	已添加有关如何销毁资源的说明。有关更多信息，请参阅销毁资源。此外，已将端口添加到许可名单中。有关更多信息，请参阅要求。	2023 年 8 月 17 日
初始版本	《Wickr Enterprise 自动安装指南》的初始版本	2023 年 8 月 4 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。