



自動化安裝指南

Wickr Enterprise



Wickr Enterprise: 自動化安裝指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 Wickr Enterprise ?	1
開始使用	2
要求	2
安裝相依項目	3
設定	4
引導	6
部署	6
產生 KOTS Config	7
連線至 Kubernetes	8
透過堡壘代理連線	8
安裝 Wickr Enterprise	10
手動安裝 Wickr Enterprise	10
使用 Lambda 安裝 Wickr Enterprise	10
安裝後	11
KOTS 管理員主控台	11
Wickr 管理員主控台	11
內容值	13
銷毀資源	16
疑難排解	17
刪除 Wickr 命名空間	17
重設 KOTS 管理員主控台密碼	17
使用堡壘連線至 EKS 叢集的問題	17
自訂安裝	18
要求	18
硬體要求	18
軟體需求	21
網路需求	21
Architecture	22
安裝	23
KOTS 管理主控台	24
輸入設定	24
資料庫設定	25
外部資料庫設定	25
內部資料庫設定	26

升級到 MySQL 8.0	27
S3 檔案儲存	27
持久性磁碟區宣告設定	28
TLS 憑證設定	28
Let's Encrypt	29
固定憑證	29
憑證提供者	29
產生自我簽署憑證	29
呼叫設定	30
呼叫輸入設定	31
考量事項	31
參考架構	32
Kubernetes 叢集自動擴展器 (選用)	33
AWS	33
Google 雲端	34
Azure	35
備份	36
使用 Velero 文件進行安裝	36
Airgap 安裝	37
Airgap 安裝的行動通知	38
Wickr 管理員主控台	38
安全性設定	39
常見問答集	39
內嵌叢集安裝	40
開始使用	40
要求	40
標準安裝	41
多節點安裝	42
連接埠需求	42
授權需求	43
在初始設定期間建立其他節點	43
將其他節點新增至現有的內嵌叢集安裝	44
KOTS 管理員主控台組態	44
其他安裝需求	46
對內嵌叢集安裝進行故障診斷	49
一般問題	49

升級問題	49
文件歷史紀錄	52
.....	liii

什麼是 Wickr Enterprise ?

Wickr Enterprise 是一種end-to-end加密的自我託管服務，可協助組織和政府機構透過one-to-one和群組傳訊、語音和視訊通話、檔案共用和螢幕共用安全地通訊。客戶可以使用 Wickr Enterprise 來克服與消費者級訊息應用程式相關的資料保留義務，並安全地促進協作。進階安全和管理控制可協助組織滿足法律和法規要求，並針對資料安全挑戰建置自訂解決方案。

資訊可以記錄到私有、客戶控制的資料存放區，以供保留和稽核之用。客戶對資料具有全面的管理控制，包括設定許可、設定暫時性傳訊選項，以及定義安全群組。管理員也可以使用 Wickr 機器人安全地自動化工作流程。Wickr Enterprise 與其他服務整合，例如 Active Directory 和單一登入 (SSO) 與 OpenID Connect (OIDC)。若要開始設定 Wickr Enterprise，請參閱 [Wickr Enterprise 入門](#)。

Note

如果您還沒有 Wickr Enterprise 部署套件，請參閱[聯絡我們](#)以進行業務查詢。

Wickr Enterprise 入門

主題

- [要求](#)
- [安裝相依項目](#)
- [設定](#)
- [引導](#)
- [部署](#)
- [產生 KOTS Config](#)

要求

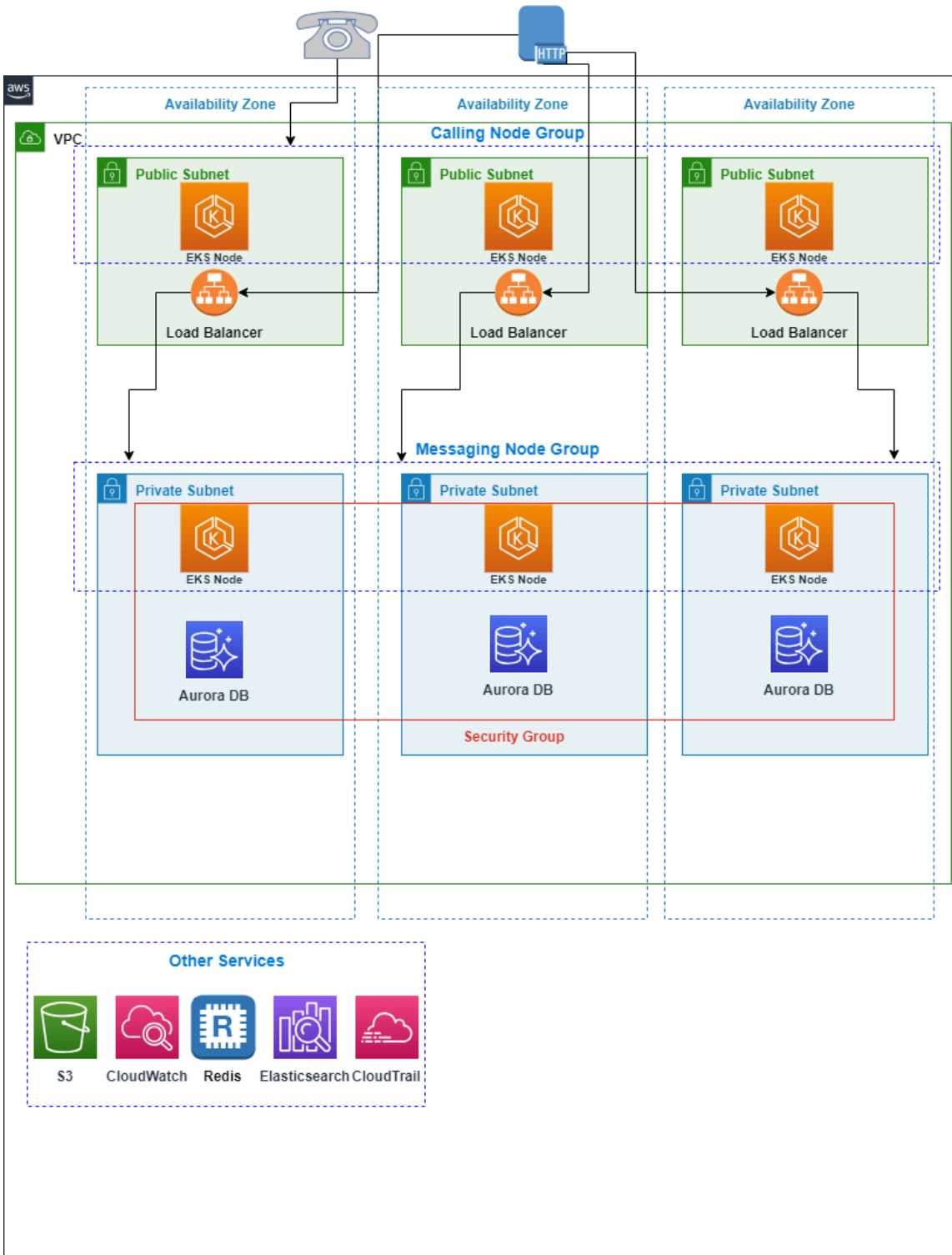
開始之前，請確認符合下列要求：

- 下載 Node.js 16+
- AWS CLI 已使用您帳戶的登入資料設定。

這些將來自您在的組態檔案，`~/.aws/config`或使用AWS_環境變數。

- 安裝 kubectl。如需詳細資訊，請參閱《Amazon EKSUser》中的[安裝或更新 kubectl](#)。
- 安裝 kots CLI。如需詳細資訊，請參閱[安裝 kots CLI](#)。
- 允許清單的連接埠：HTTPS 和 TCP 呼叫流量為 443/TCP；UDP 呼叫流量為 16384-19999/UDP；TCP/8443

架構



安裝相依項目

您可以使用下列命令，將所有相依性新增至預設套件：

```
npm install
```

設定

AWS Cloud Development Kit (AWS CDK) 使用內容值來控制應用程式的組態。Wickr Enterprise 使用 CDK 內容值來控制設定，例如 Wickr Enterprise 安裝的網域名稱或保留 RDS 備份的天數。如需詳細資訊，請參閱《AWS Cloud Development Kit (AWS CDK) 開發人員指南》中的[執行期內容](#)。

有多種方式可以設定內容值，但建議您在 `cdk.context.json` 中編輯這些值，以符合您的特定使用案例。只有開頭為 `wickr/` 的內容值與 Wickr Enterprise 部署相關；其餘則是 CDK 特定的內容值。若要在下次透過 CDK 進行更新時保持相同的設定，請儲存此檔案。

您至少必須設定 `wickr/licensePath`、`wickr/domainName` 和 `wickr/acm:certificateArn` 或 `wickr/route53:hostedZoneId` 和 `wickr/route53:hostedZoneName`。

使用公有託管區域

如果您的 `cdk.context.json` 中有 Route 53 公有託管區域 AWS 帳戶，建議您使用下列設定來設定 CDK 內容：

- `wickr/domainName` - 用於此 Wickr Enterprise 部署的網域名稱。如果您使用 Route 53 公有託管區域，系統會自動建立此網域名稱的 DNS 記錄和 ACM 憑證。
- `wickr/route53:hostedZoneName` - 要在其中建立 DNS 記錄的 Route 53 託管區域名稱。
- `wickr/route53:hostedZoneId` - Route 53 託管區域 ID，在其中建立 DNS 記錄。

此方法會代表您建立 ACM 憑證，以及將網域名稱指向 Wickr Enterprise 部署前負載平衡器的 DNS 記錄。

沒有公有託管區域

如果您的帳戶中沒有 Route 53 公有託管區域，則必須手動建立 ACM 憑證，並使用 `wickr/acm:certificateArn` 內容值匯入 CDK。

- `wickr/domainName` - 用於此 Wickr Enterprise 部署的網域名稱。如果您使用 Route 53 公有託管區域，系統會自動建立此網域名稱的 DNS 記錄和 ACM 憑證。
- `wickr/acm:certificateArn` - 要在負載平衡器上使用的 ACM 憑證 ARN。如果您的帳戶無法使用 Route 53 公有託管區域，則必須提供此值。

將憑證匯入 ACM

您可以使用下列命令匯入外部取得的憑證：

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

輸出將是憑證 ARN，應該用於 `wickr/acm:certificateArn` 內容設定的值。請務必讓上傳的憑證對有效 `wickr/domainName`，否則 HTTPS 連線將無法驗證。如需詳細資訊，請參閱 AWS Certificate Manager 《使用者指南》中的 [匯入憑證](#)。

建立 DNS 記錄

由於沒有可用的公有託管區域，因此必須在部署完成後手動建立 DNS 記錄，以指向 Wickr Enterprise 部署前的負載平衡器。

部署到現有的 VPC

如果您需要使用現有的 VPC，則可以使用。不過，VPC 必須設定為符合 EKS 所需的規格。如需詳細資訊，請參閱 [《Amazon EKS 使用者指南》中的檢視 VPC 和子網路的 Amazon EKS 聯網需求](#)，並確保要使用的 VPC 符合這些需求。

此外，強烈建議您確保具有下列服務的 VPC 端點：

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

若要將資源部署到現有的 VPC，請設定下列內容值：

- `wickr/vpc:id` - 要將資源部署到其中的 VPC ID (例如 `vpc-412beef`)。
- `wickr/vpc:cidr` - VPC 的 IPv4 CIDR (例如 `172.16.0.0/16`)。
- `wickr/vpc:publicSubnetIds` - VPC 中以逗號分隔的公有子網路清單。Application Load Balancer 和呼叫 EKS 工作者節點將部署在這些子網路中 (例如 `subnet-6ce9941,subnet-1785141,subnet-2e7dc10`)。
- `wickr/vpc:privateSubnetIds` - VPC 中以逗號分隔的私有子網路清單。EKS 工作者節點和堡壘伺服器將部署在這些子網路中 (例如 `subnet-f448ea8,subnet-3eb0da4,subnet-ad800b5`)。
- `wickr/vpc:isolatedSubnetIds` - VPC 中隔離子網路的逗號分隔清單。RDS 資料庫將部署在這些子網路中 (例如 `subnet-d1273a2,subnet-33504ae,subnet-0bc83ac`)。
- `wickr/vpc:availabilityZones` - VPC 中子網路可用區域的逗號分隔清單 (例如 `us-east-1a,us-east-1b,us-east-1c`)。

如需介面 VPC 端點的詳細資訊，請參閱[使用介面 VPC 端點存取 AWS 服務](#)。

其他設定

如需詳細資訊，請參閱[內容值](#)。

引導

如果這是您第一次在此特定 AWS 帳戶和區域上使用 CDK，您必須先引導帳戶以開始使用 CDK。

```
npx cdk bootstrap
```

部署

此程序大約需要 45 分鐘。

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

完成之後，基礎設施已建立，您可以開始安裝 Wickr Enterprise。

建立 DNS 記錄

如果您在設定 CDK 時使用公有託管區域，則不需要此步驟。

部署程序的輸出將包含值 `WickrAlb.AlbDnsName`，這是負載平衡器的 DNS 名稱。輸出看起來像：

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

在此情況下，DNS 名稱為 `Wickr-Alb-1Q5IBPJR4ZVZR-409483305.us-west-2.elb.amazonaws.com`。這是為您的網域名稱建立 CNAME 或 A/AAAA (ALIAS) 記錄時應使用的值。

如果您沒有部署的輸出，請執行下列命令來顯示負載平衡器 DNS 名稱：

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

產生 KOTS Config

Warning

此檔案包含安裝的敏感資訊。請勿公開共用或儲存。

Wickr Enterprise 安裝程式需要一些有關基礎設施的組態值，才能成功安裝。您可以使用協助程式指令碼來產生組態值。

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

如果您在第一個步驟中將外部憑證匯入 ACM，請將 `--ca-file` 旗標傳遞至此指令碼，例如：

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

如果您收到堆疊不存在的錯誤，請將 `AWS_REGION` 環境變數 (`export AWS_REGION=us-west-2`) 設定為您選取的區域，然後再試一次。或者，如果您設定內容值 `wickr/stackSuffix`，請使用 `--stack-suffix` 旗標傳遞尾碼。

連線至 Kubernetes 叢集

Amazon EKS API 只能透過建立為部署一部分的堡壘主機存取。因此，所有 `kubectl` 命令都必須在堡壘主機本身上執行，或透過堡壘主機代理。

透過堡壘代理連線

第一次連線到叢集時，您必須使用 `aws eks update-kubeconfig` 命令更新本機 `kubeconfig` 檔案，然後在組態 `proxy-url` 中設定。然後，每次您想要連線到叢集時，您都會使用堡壘主機啟動 SSM 工作階段，以將連接埠轉送至代理以進行 API 存取。

一次性設定

WickrEks CloudFormation 堆疊上有一個以開頭的名稱的輸出值 `WickrEnterpriseConfigCommand`。值包含為叢集產生 `kubectl` 組態所需的完整命令。您可以使用下列命令檢視此輸出：

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

這應該會輸出開頭為 `aws eks update-kubeconfig` 的命令。執行此命令。

接著，必須將 Kubernetes 組態修改為透過堡壘主機代理請求。您可以使用下列命令來完成此操作：

```
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
```

如果正常運作，您會看到類似的輸出 `'Property "clusters.arn:aws:eks:us-west-2:012345678912:cluster/WickrEnterprise5B8BF472-1234a41c4ec48b7b615c6789d93dcce.proxy-url" set.'`

連接埠轉送至堡壘

若要連線至 Amazon EKS 叢集，您必須啟動 SSM 工作階段，將轉送請求移植到在堡壘主機上執行的代理。執行此操作的命令會以 `WickrEks` 堆疊 `BastionSSMProxyEKSCCommand` 上的輸出的形式提供。執行下列命令以檢視輸出值：

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

其輸出的命令將以開頭 `aws ssm start-session`。執行此命令以啟動在連接埠 8888 上執行的本機代理，您可以透過此連接埠連線到 Amazon EKS 叢集。如果連接埠轉送正常運作，輸出應顯示「等待連線...」。請讓此程序在存取 Amazon EKS 叢集所需的整個時間內持續執行。

如果一切設定正確，您將能夠在另一個終端機 `kubectl get nodes` 中執行，以列出 Amazon EKS 叢集中的工作者節點：

```
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
```

安裝 Wickr Enterprise

與 Kubernetes 叢集建立連線後，您可以使用 `kubectl kots` 外掛程式開始安裝 Wickr Enterprise。您將需要 KOTS 授權檔案 (Wickr 提供的 `.yaml` 檔案) 和 Config 值檔案，這些檔案儲存在產生 KOTS Config 區段 `wickr-config.json` 中的檔案。如需產生 KOTS Config 的詳細資訊，請參閱 [產生 KOTS Config](#)。

手動安裝 Wickr Enterprise

下列命令將開始安裝 Wickr Enterprise：

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

系統會提示您輸入 KOTS 管理員主控台的密碼。儲存此密碼，因為未來您需要它來升級或變更 Wickr Enterprise 安裝的組態。

安裝完成後，`kubectl kots` 會開啟本機連接埠 (通常是 `http://localhost:8080`)，提供 KOTS 管理員主控台的存取權。您可以在此網站上變更或監控 Wickr Enterprise 安裝的狀態，或前往您在瀏覽器中為安裝設定的網域名稱，開始設定 Wickr。

使用 Lambda 安裝 Wickr Enterprise

在 CDK 部署期間，會建立並叫用 Lambda，以代表您自動完成 Wickr Enterprise 安裝。若要手動叫用，請開啟 AWS 主控台並尋找 `WickrLambda-func*` lambda 函數，在測試索引標籤下，選取 `test`，輸入不相關。

安裝後

有兩個 Web 主控台可用於管理您的 Wickr Enterprise 安裝：KOTS 管理員主控台和 Wickr 管理員主控台。

Note

進行任何必要的變更，以反映組織的備份和記錄政策 (Amazon S3 設定、Elastic Load Balancing 存取日誌、Amazon Virtual Private Cloud 流程日誌)。

KOTS 管理員主控台

此界面用於管理已部署的 Wickr Enterprise 版本。您可以查看安裝、修改組態或執行升級的狀態。KOTS 管理員主控台只能透過 Kubernetes 連接埠向前存取，可使用下列命令開啟：

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

Note

您必須先依照轉送至堡壘區段的連接埠所述來設定堡壘連線。如需轉送至堡壘之連接埠的詳細資訊，請參閱[透過堡壘代理連線](#)。

成功設定連接埠轉送時，上一個命令會輸出下列項目：

- Press Ctrl+C to exit
- Go to <http://localhost:8800> to access the Admin Console

使用提供的 URL 存取 KOTS 管理員主控台。登入的密碼是您在安裝 `kubectl kots install` 期間執行時所選擇的密碼。如果您需要重設密碼，請參閱[重設 KOTS 管理員主控台密碼](#)。

Wickr 管理員主控台

此界面用於設定 Wickr Enterprise 安裝，以設定網路、使用者和聯合。您可以在設定為指向 Load Balancer 的 DNS 名稱上，透過 HTTPS 存取它。如果 DNS 是使用公有託管區域自動設定，則網域名稱是 `wickr/domainName` 內容值的值。

預設使用者名稱為 admin，密碼為 Password123。您必須在第一次登入時變更此密碼。

內容值

內容值是可與應用程式、堆疊或建構相關聯的鍵值對。它們可以從檔案（通常是 `cdk.context.json` 專案目錄中的 `cdk.json` 或）或在命令列上提供給應用程式。CDK 使用內容值來控制應用程式的組態。Wickr Enterprise 使用 CDK 內容值來控制設定，例如 Wickr Enterprise 安裝的網域名稱或保留 RDS 備份的天數。

有多種方式可以設定內容值，但建議您在 `cdk.context.json` 中編輯這些值，以符合您的特定使用案例。只有開頭為 `wickr/` 的內容值與 Wickr Enterprise 部署相關。

名稱	描述	預設
<code>wickr/licensePath</code>	KOTS 授權的路徑 (Wickr 提供的 <code>.yaml</code> 檔案)。	null
<code>wickr/domainName</code>	用於此 Wickr Enterprise 部署的網域名稱。如果使用 Route 53 公有託管區域，系統會自動建立此網域名稱的 DNS 記錄和 ACM 憑證。	null
<code>wickr/route53:hostedZoneId</code>	要在其中建立 DNS 記錄的 Route 53 託管區域 ID。	null
<code>wickr/route53:hostedZoneName</code>	Route 53 託管區域名稱，要在其中建立 DNS 記錄。	null
<code>wickr/acm:certificateArn</code>	要在 Load Balancer 上使用的 ACM 憑證 ARN。如果您的帳戶無法使用 Route 53 公有託管區域，則必須提供此值。	null
<code>wickr/caPath</code>	憑證路徑，只有在使用自我簽署憑證時才需要。	null
<code>wickr/vpc:id</code>	要部署資源的 VPC ID。只有在部署到現有的 VPC 時才需要。	null

名稱	描述	預設
	如果取消設定，將會建立新的 VPC。	
wickr/vpc:cidr	要與建立的 VPC 建立關聯的 IPv4 CIDR。如果部署到現有的 VPC，請將此設定為現有 VPC 的 CIDR。	172.16.0.0/16
wickr/vpc:availabilityZones	可用區域的逗號分隔清單。只有在部署到現有的 VPC 時才需要。	null
wickr/vpc:publicSubnetIds	以逗號分隔的公有子網路 IDs 清單。只有在部署到現有的 VPC 時才需要。	null
wickr/vpc:privateSubnetIds	以逗號分隔的私有子網路 IDs 清單。只有在部署到現有的 VPC 時才需要。	null
wickr/vpc:isolatedSubnetIds	RDS 資料庫隔離子網路 IDs 的逗號分隔清單。只有在部署到現有的 VPC 時才需要。	null
wickr/rds:deletionProtection	在 RDS 執行個體上啟用刪除保護。	true
wickr/rds:removalPolicy	RDS 執行個體 'snapshot'、'destroy' 或 'retain.' 的移除政策	快照
wickr/rds:readerCount	要在 RDS 叢集中建立的讀取器執行個體數目。	1
wickr/rds:instanceType	用於 RDS 執行個體的執行個體類型。	r6g.xlarge

名稱	描述	預設
wickr/rds:backupRetentionDays	保留備份的天數。	7
wickr/eks:namespace	EKS 中 Wickr 服務的預設命名空間。	柳條
wickr/eks:defaultCapacity	訊息基礎設施的 EKS 工作者節點數量。	3
wickr/eks:defaultCapacityCalling	呼叫基礎設施的 EKS 工作者節點數量。	2
wickr/eks:instanceTypes	用於 Messaging EKS 工作者節點的執行個體類型逗號分隔清單。	m5.xlarge
wickr/eks:instanceTypesCalling	用於呼叫 EKS 工作者節點的執行個體類型逗號分隔清單。	c5n.large
wickr/eks:enableAutoscaler	切換為 EKS 啟用 Cluster Autoscaler 功能。	true
wickr/s3:expireAfterDays	將檔案上傳從 S3 儲存貯體中移除的天數。	1095
wickr/eks:clusterVersion	叢集版本，包括 Kubernetes 版本、kubectlLayer 版本、albController版本、nodeGroupRelease 版本等。	1.27
wickr/stackSuffix	要套用至 CloudFormation 堆疊名稱的尾碼。	"
wickr/autoDeployWickr	使用 lambda 自動部署 Wickr 應用程式。	true

銷毀資源

若要刪除 AWS CDK 此應用程式建立的所有項目，您必須在所有其他WickrRds堆疊之前刪除堆疊。

為了讓 Amazon RDS 資源正確刪除，必須停用刪除保護，且移除政策必須設定為 snapshot 或 destroy。如果這些不是目前的設定，請修改內容 AWS CDK 中的 wickr/rds:deletionProtection 和 wickr/rds:removalPolicy 值，並透過執行 重新部署 Amazon RDS 堆疊 `npx cdk deploy -e WickrRds`。

正確設定刪除保護和移除政策後，請 `cdk destroy` 為 WickrRds 堆疊執行：

```
npx cdk destroy WickrRds
```

當 WickrRds 堆疊完成銷毀時，可以使用下列命令銷毀剩餘的 CloudFormation 堆疊：

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

疑難排解

刪除 Wickr 命名空間

如果您需要刪除wickr命名空間以重新開始，請務必先備份 CDK 在該命名空間內建立的任何服務帳戶。這些服務帳戶允許 Wickr 服務透過 IAM 角色與 AWS APIs 通訊。如果沒有它們，透過 Amazon Simple Storage Service (Amazon S3) 上傳檔案之類的任務將無法再運作。

使用下列命令來備份服務帳戶，並刪除和重新建立wickr命名空間和適當的服務帳戶：

```
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
```

重設 KOTS 管理員主控台密碼

您可以使用下列命令重設 KOTS 管理員主控台密碼：

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

當您變更此密碼時，您可能也想要更新 wickr/kots Secrets Manager 秘密，但任何自動化通常不會再次使用它。

使用堡壘連線至 EKS 叢集的問題

如果您透過堡壘與 EKS 叢集的連線似乎緩慢或偶爾逾時，則在執行kubectl命令時可能會看到下列錯誤：

net/http：請求在等待連線時取消（在等待標頭時超過用戶端逾時）

透過 SSM 登入堡壘主機（請參閱 WickrEks 堆疊BastionSSMCommand上的）並重新啟動tinyproxy服務，通常可以解決此問題：

```
sudo systemctl restart tinyproxy
```

自訂安裝

在自訂安裝區段中，您將了解如何安裝 Wickr Enterprise。

主題

- [要求](#)
- [Architecture](#)
- [安裝](#)
- [輸入設定](#)
- [資料庫設定](#)
- [S3 檔案儲存](#)
- [持久性磁碟區宣告設定](#)
- [TLS 憑證設定](#)
- [呼叫設定](#)
- [呼叫輸入設定](#)
- [Kubernetes 叢集自動擴展器 \(選用\)](#)
- [備份](#)
- [Airgap 安裝](#)
- [Wickr 管理員主控台](#)
- [安全性設定](#)
- [常見問答集](#)

要求

開始安裝 Wickr Enterprise 之前，請確認符合下列要求。

硬體要求

Wickr Enterprise 需要 Kubernetes 叢集才能運作。您可以在已啟用低資源模式的單一節點上操作，但這不建議用於一般生產用途。在生產部署中，我們建議至少三個簡訊工作者節點，以及至少兩個呼叫工作者節點。

工作者節點應具有下列最低規格。

- 2 到 4 個 CPU 核心
- 8 GB 的 Ram
- 200 GB 的磁碟空間

最低硬體需求

在低資源模式下執行的單一工作者節點叢集至少需要 3000 公尺 CPU 和 5846Mi Ram。這不包含 kube-system Pod。

Pod 的資源需求

Pod 名稱	Owner	CPU	記憶體
admin-api	Wickr	100 公尺	256Mi
directory	Wickr	100 公尺	128Mi
過期者	Wickr	100 公尺	128Mi
fileproxy	Wickr	100 公尺	256Mi
oidc	Wickr	100 公尺	128Mi
opensearch	Wickr	500 公尺	100Mi
奧維爾	Wickr	50 公尺	128Mi
奧維爾雷迪斯	Wickr	50 公尺	128Mi
push-device	Wickr	100 公尺	128Mi
偵錯工具	Wickr	50 公尺	256Mi
反應	Wickr	100 公尺	64Mi
收據	Wickr	250 公尺	128Mi
redis	Wickr	50 公尺	128Mi
server-api	Wickr	250 公尺	256Mi

Pod 名稱	Owner	CPU	記憶體
切換板	Wickr	250 公尺	512Mi
kotsadm	KOTS	50 公尺	50Mi
kotsadm-minio	KOTS	100 公尺	512Mi
kotsadm-rqlite	KOTS	200 m	1Gi
minio-operator	內部 S3	200 m	256Mi
微型租戶	內部 S3	100 公尺	256Mi
mysql-主要	內部 MySQL	100 公尺	512Mi
mysql-secondary	內部 MySQL	100 公尺	512Mi

儲存需求

Wickr Enterprise 需要在建立持久性磁碟區宣告時使用預設 StorageClass。在氣隙隔離環境或內部部署中部署時，您可能需要為叢集設定一個。一個可用的選項是[長角](#)。建議的磁碟空間需求會根據使用內部 S3 選項和內部 Mysql 選項，以及您想要用於檔案上傳的空間量而有所不同。

- 內部映像快取：~60 Gi
- RabbitMQ：24 Gi 預設/8 Gi 低資源模式
- Redis：在低資源模式下為 24 Gi 預設 / 8 Gi
- OpenSearch：在低資源模式下為 24 Gi 預設 / 8 Gi
- 內部 Mysql：80 Gi 預設/20Gi 低資源模式
- 內部 S3：160 Gi 預設/2Gi 低資源模式
- KOTS Minio：4 Gi
- KOTS Rqlite：1 Gi

儲存體大小下限

- 具有內部 S3 和內部 Mysql 的 377 Gi 預設
- 低資源模式下的 111 Gi

Kubernetes 版本需求

Wickr Enterprise 依賴複寫的 KOTS。已複寫的商業軟體分發平台，提供目前支援的 Kubernetes 版本清單。如需詳細資訊，請參閱 [Kubernetes 版本相容性](#)。

軟體需求

Wickr Enterprise 需要 Kubernetes 叢集和 KOTS 才能運作。如需支援的作業系統和 Kubernetes 版本，請參閱 KOTS 文件。如需詳細資訊，請參閱 [最低系統需求](#)。

開發人員主機系統

作業系統 — 本文件中的命令設計用於已安裝 WSL（適用於 Linux 的 Windows 子系統）的 Linux、MacOS 或 Windows 上。

內部狀態服務

Wickr Enterprise 可以為 MySQL 資料庫和 S3 相容儲存體提供內部服務，但對於一般生產用途，建議您從 Kubernetes 叢集外部提供這些服務。

- MySQL 5.7 資料庫
 - Amazon RDS MySQL 5.7 或 MySQL 5.7 資料庫（外部）
 - Mysql Bitnami Helm Chart（內部）
 - 檔案儲存
 - Amazon S3 或 S3 相容儲存提供者（外部）
 - Minio Operator Helm Chart（內部）

網路需求

Wickr Enterprise 需要 FQDN、SSL 憑證，以及特定的開放 TCP 和 UDP 連接埠。

- FQDN：Wickr Enterprise 部署要使用的網域或子網域。
- SSL 憑證：由公有 CA 簽署的 SSL 憑證金鑰對或自我簽署的憑證金鑰對。憑證必須在一般名稱中列出 FQDN，並做為 SAN DNS 項目。憑證也必須啟用 serverAuth extendedKeyUsage 延伸模組。
- 線上安裝將需要對複寫和第三方資源的輸出存取權。複寫 會維護其 IP 地址的清單。如需詳細資訊，請參閱 [複寫的 IP 地址](#)。複寫也會維護所需的第三方資源清單。如需詳細資訊，請參閱 [線上安裝的防火牆開啟](#)。

- 氣隙隔離安裝需要存取私有容器登錄檔。

訊息節點

訊息節點不需要公有 IPV4 地址，且應位於私有子網路中。訊息流量將透過 LoadBalancer 或 Ingress 進入叢集。

呼叫節點

呼叫節點需要公有 IPV4 地址，因此它們必須位於公有子網路中。根據預設，通話媒體會透過 UDP 傳輸。啟用 TCP 呼叫時，TCP Proxy 將接受 TCP 443 上的連線，並將它們代理到 Orville 服務。

- TCP : 443 呼叫 TCP Proxy
- UDP : 16384-16484 音訊/視訊串流

安裝和組態存取

透過 Kubernetes 連接埠向前存取 KOTS 管理員主控台以進行安裝和組態。

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

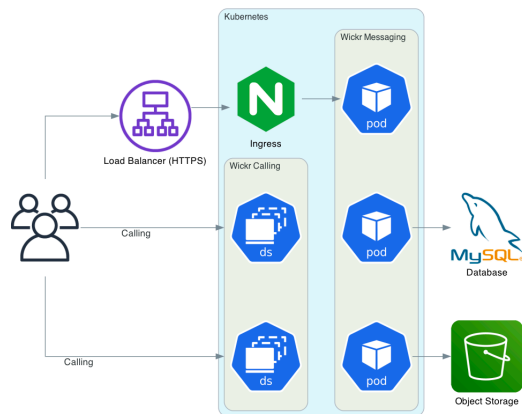
授權需求

安裝需要 .yaml 格式的授權檔案，這會由 Wickr Support 提供給您。

Architecture

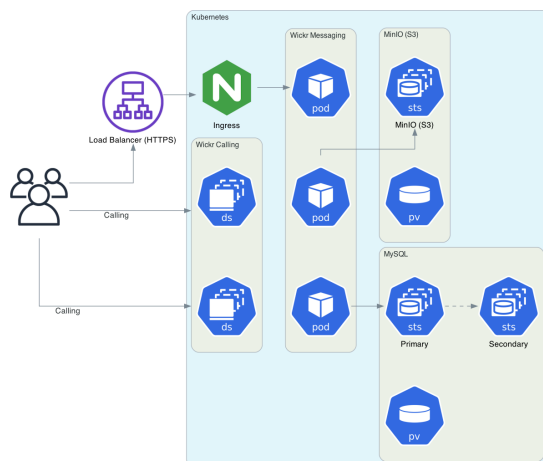
建議的生產架構

下圖顯示設定為建議用於生產的 Wickr Enterprise，其中 MySQL 和物件儲存服務都位於 Kubernetes 叢集外部。



內部或測試架構

下圖顯示使用內部 MySQL 和物件儲存服務的 Wickr Enterprise 組態。雖然它可以滿足特定部署的特定需求，但不建議用於一般生產用途。



安裝

1. 安裝 [kubectI](#) 和 [kots CLI](#)。
2. 連線至 Kubernetes 叢集。

3. 從 Wickr Support 取得 Wickr Enterprise 授權檔案。
4. 使用下列命令安裝 Wickr Enterprise。

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

Note

license.yaml 代表您提供的授權檔案。

初始安裝後，KOTS 管理員主控台將提供叢集層級管理和組態選項。

KOTS 管理主控台

此界面用於管理已部署的 Wickr Enterprise 版本。您可以查看安裝狀態、修改組態或執行 Wickr Enterprise 的升級。KOTS 管理員主控台只能透過 Kubernetes 連接埠向前存取，可使用下列命令開啟：

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

輸入設定

輸入控制器

Wickr Enterprise 支援四種輸入控制器類型：

- LoadBalancer (預設)
 - 負載平衡器物件可能需要完全內部部署安裝中的明確組態，即使它通常由雲端提供者提供。
 - 使用 LoadBalancer 服務類型部署輸入控制器 (inress-nginx) 服務。這需要 Kubernetes 叢集在支援外部負載平衡器的平台上執行。
- 現有的 ALB
 - 將輸入控制器連接至現有的 ALB。
 - 您需要提供現有的 Application Load Balancer 目標群組 ARN。
- 現有的 NLB

- 將輸入控制器連接至現有的 NLB。
- 您需要提供現有的 Network Load Balancer 目標群組 ARN。
- NodePort
 - 輸入控制器 (inress-nginx) 將設定為使用 NodePort 服務類型，這會在 Kubernetes 叢集中的所有節點上開啟連接埠，並將流量轉送至輸入。然後，可以透過 DNS 或一些外部負載平衡器將用戶端流量導向這些節點。
 - 您可以選擇 1-65535 的連接埠範圍，或使用 30000-32767 的隨機連接埠。
- Ingress
 - 使用您自己的輸入控制器。此組態會接受輸入類別名稱，然後服務會在其輸入資訊清單中使用該類別名稱。這表示輸入控制器已透過其他一些負載平衡機制設定一些外部連線。
 - 目前僅支援 [ingress-nginx](#) 控制器。

萬用字元主機名稱

根據預設，輸入路由會以主機值 `*` 定義。停用此設定以使用 Wickr Enterprise Server 定義的主機名稱。IP 型主機名稱需要萬用字元主機名稱。

資料庫設定

Wickr Enterprise 需要 MySQL 8.0 資料庫。如果您在 MySQL 5.7 上，請參閱 [升級到 MySQL 8.0](#) 升級。我們建議您使用 Kubernetes 叢集外部的資料庫，例如 Amazon RDS，但您也可以選擇在 Kubernetes 叢集內部署內部 MySQL 資料庫做為安裝的一部分。

外部資料庫設定

- 主機名稱：資料庫伺服器的主機名稱或 IP 地址。
- 讀取器主機名稱：資料庫伺服器的唯讀端點的主機名稱或 IP 地址（如果可用）。
- 連接埠：將存取 MySQL 的連接埠。
- 資料庫名稱：在伺服器上建立的資料庫名稱。
- 使用者名稱：具有資料庫存取許可的使用者。
- 密碼：該使用者的密碼。
- CA 憑證：透過 TLS 連線至資料庫的 PEM 憑證。

Note

確保您的 MySQL 安裝使用預設的 latin1 字元集搭配 latin1_swedish_ci 定序。這可以透過驗證您的 MySQL 伺服器是否以下列旗標啟動來完成：

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

內部資料庫設定

對於具有二進位複寫的 MySQL 主要和次要資料庫類型，內部資料庫類型會將兩個 StatefulSets 部署到您的叢集。次要 不會接收任何流量，且僅適用於災難復原和備份。

儲存大小：資料庫 Pod 的持久性磁碟區大小（以 GB 為單位）。

增加 MySQL 儲存體大小

Note

StorageClass 的磁碟區類型必須支援磁碟區擴展，才能增加儲存體大小。如需詳細資訊，請參閱[磁碟區擴展](#)。

Wickr Enterprise 中使用的 MySQL 服務會部署為 Kubernetes 中的 StatefulSet 資源。StatefulSets 可讓資源的許多屬性不可變，包括持久性磁碟區宣告範本。做為 StatefulSets 不可變性的解決方法，必須執行下列動作來增加 MySQL 使用的磁碟區大小。

1. 編輯 data-mysql-primary-0 和 的持久性磁碟區宣告 data-mysql-secondary-0。

```
1. kubectl -n wickr edit pvc data-mysql-primary-0. Set spec.resources.requests.storage 至所需的儲存體大小。
```

```
2. kubectl -n wickr edit pvc data-mysql-secondary-0. Set spec.resources.requests.storage 至所需的儲存體大小。
```

2. 刪除現有的 StatefulSets，但傳遞 --cascade=orphan 旗標以離開 Pod。

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

3. 在 KOTS UI 中，更新儲存大小設定，以符合您在步驟 1 中設定的值。儲存並部署此組態。

4. 重新啟動 StatefulSets 以展開磁碟區，並使 MySQL 服務重新上線。

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

升級到 MySQL 8.0

外部資料庫 (RDS)

若要讓 Wickr 後端離線，請完成下列步驟。

1. 尋找傳入的命名空間 `kubectl get deployments --all-namespaces`

在下面的範例中，命名空間為 Wickr，複本為 3。

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

2. 向下擴展傳入 `kubectl scale deployment/ingress-nginx-controller --replicas=0 -n wickr`
3. 拍攝快照以備份資料庫。如需詳細資訊，請參閱《Amazon Relational Database Service 使用者指南》中的[管理手動備份](#)。
4. 將引擎版本升級至 MySQL 8.0.x (不支援 MySQL 8.4)。如需詳細資訊，請參閱《Amazon Relational Database Service 使用者指南》中的[升級資料庫執行個體引擎版本](#)。

若要讓 Wickr 後端上線，請向後擴展輸入 `kubectl scale deployment/ingress-nginx-controller --replicas=3 -n wickr`

內部資料庫

如需詳細資訊，請參閱[備份和還原 MySQL](#)。

S3 檔案儲存

Wickr Enterprise 需要 S3 相容儲存服務。我們建議您使用 Kubernetes 叢集外部的 S3 服務，例如 Amazon S3，但您也可以選擇在 Kubernetes 叢集內部署內部 S3 服務做為安裝的一部分。

外部 S3 設定

- 儲存貯體名稱：儲存檔案上傳的 S3 儲存貯體名稱。
- 區域：S3 儲存貯 AWS 體的區域。
- 端點：設定 Wickr 將用來與 S3 API 互動的端點。預設為區域的 S3 服務端點。
- Fileproxy Service 帳戶名稱：僅限 Amazon S3。用於使用服務帳戶的 IAM 角色向 S3 驗證的現有 Kubernetes 服務帳戶名稱。
- 外部 S3 存取金鑰：這是您現有的 S3 存取金鑰。
- 外部 S3 私密金鑰：這是您現有的 S3 私密金鑰。

內部 S3 設定

內部 S3 類型將部署預設的 4 個 MinIO 伺服器 Pod，每個都包含 4 個持久性磁碟區宣告。預設組態會使用 MinIO 的清除編碼來提高容錯能力。

- 內部 S3 伺服器計數：要建立的 MinIO 伺服器 Pod 數量，容錯部署的預設值為 4。對於開發/測試部署，此值可以設定為低至 1。
- 內部 S3 磁碟區計數：在每個 MinIO 伺服器 Pod 中建立的 MinIO 磁碟區數目，容錯部署的預設值為 4。對於開發/測試部署，此值可以設定為低至 1。
- 內部 S3 磁碟區大小：在 MinIO 伺服器 Pod 中建立的 MinIO 磁碟區大小，預設為 10GB。
- 預設的內部 S3 部署將使用具有 4 個 PVCs 4 個伺服器。每個 PVC 都是 10 Gi，產生 160 Gi 原始儲存體，並提供 120 Gi 刪除編碼儲存體供使用者使用。
- Minio Erasure 編碼計算器可用。如需詳細資訊，請參閱[清除程式碼計算器](#)。

持久性磁碟區宣告設定

Wickr Enterprise 需要持久性磁碟區宣告來存放具狀態的資料。此設定可讓您指定要使用之 Storage Class 的名稱。如果保留空白，Wickr 將嘗試使用預設儲存體方案。不支援在部署 Wickr 之後變更儲存體方案。

預設 StorageClass for Persistent Volume Claims 通常由雲端供應商提供，但在完全內部部署安裝中，可能需要使用第三方服務進行明確組態，例如 [Longhorn](#)。

TLS 憑證設定

上傳 PEM 憑證和私有金鑰以終止 TLS。憑證上的主體別名必須符合 Wickr Enterprise 部署設定中設定的主機名稱。

對於憑證鏈欄位，在上傳之前，請將任何中繼憑證（如果需要）與根 CA 憑證串連。

Let's Encrypt

選取此選項以使用 [Let's Encrypt](#) 自動產生憑證。憑證是透過 cert-manager 運算子使用 [HTTP-01 挑戰](#) 發出。

HTTP-01 挑戰要求所需的 DNS 名稱解析為叢集的輸入點（通常是 Load Balancer），且 TCP 連接埠 80 的流量開放給公有。這些憑證是短期的，將定期續約。必須保持連接埠 80 開啟，以允許憑證自動續約。

Note

本節明確參考 Wickr Enterprise 應用程式本身所使用的憑證。

固定憑證

使用自我簽署憑證或非用戶端裝置信任的憑證時，Wickr Enterprise 需要憑證鎖定。如果您的 Load Balancer 提供的憑證是自我簽署的，或是由與 Wickr Enterprise 安裝不同的 CA 簽署的，請在此處上傳 CA 憑證，讓用戶端接腳。

在大多數情況下，不需要此設定。

憑證提供者

如果您計劃購買憑證以與 Wickr Enterprise 搭配使用，請參閱下列清單，以取得憑證依預設可正常運作的提供者清單。如果供應商列於下方，則其憑證已使用軟體進行明確驗證。

- Digicert
- RapidSSL

產生自我簽署憑證

如果您想要建立自己的自我簽署憑證以與 Wickr Enterprise 搭配使用，下面的範例命令會包含產生所需的所有必要旗標。

```
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"
```

如果您想要建立以 IP 為基礎的自我簽署憑證，請改用下列命令。若要使用 IP 型憑證，請確定已在輸入設定下啟用萬用字元主機名稱欄位。如需詳細資訊，請參閱[傳入設定](#)。

```
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

以您想要使用的網域名稱或 IP 地址取代範例中的 \$YOUR_DOMAIN。

呼叫設定

- **需要呼叫節點**：啟用此設定時，Wickr 的呼叫服務只會部署在標籤為的 Kubernetes 節點上 `role=calling`。停用此設定以在相同節點或單一節點部署上部署呼叫和傳訊服務。

您通常也會想要在停用此設定時停用呼叫 TCP Proxy，因為 TCP Proxy 服務會在連接埠 443 上執行。

- **啟用 TCP Proxy**：此設定會控制是否在呼叫上部署 TCP 備用模式的服務。如果您有其他在 443/tcp 上執行的服務，或不需要 TCP 備用模式來進行呼叫，請停用此設定。對於計劃使用 Wickr Open Access 的部署，必須啟用此功能。
- **自動探索伺服器公有 IP 地址**：啟用此設定時，呼叫服務將透過向 <https://ipv4.icanhazip.com/> 和提出 HTTPS 請求來探索其公有 IP 地址 <https://ipv6.icanhazip.com/>。停用時，您必須啟用「使用主機主要 IP 地址來呼叫流量」或「主機名稱覆寫」設定，否則呼叫服務將無法啟動。
- **使用主機主要 IP 地址來呼叫流量**：使用 Kubernetes 節點的主要 IP 地址來呼叫服務。這表示所有 Wickr 用戶端都可以連線到節點主要 IP 地址上的 Kubernetes 節點，如[向下 API](#) `status.hostIP` 中的所示。
- **主機名稱覆寫**：提供主機名稱或 IP 地址，以做為呼叫服務的連線點傳回。只有在執行單一呼叫伺服器時，才應該使用此設定，因為針對服務的所有複本傳回相同的值。設定主機名稱覆寫並啟用「使用主機主要 IP 地址」設定時，主機主要 IP 地址設定優先。

- 呼叫已啟用主機網路：根據預設，呼叫 Pod 會使用節點的主機網路進行連線。停用此選項以公開用於呼叫流量的 NodePort 服務。如果啟用呼叫輸入，請確定已設定適當的服務來允許輸入流量。這必須停用才能符合 STIG 規範。

呼叫輸入設定

Wickr 支援呼叫輸入設定，允許用戶端連線到叢集內的任何呼叫節點，並讓呼叫路由到正確的呼叫伺服器。Wickr 支援四種呼叫輸入類型：

- LoadBalancer (預設)
 - LoadBalancer 將由雲端提供者佈建 (完全內部部署安裝需要額外的組態)。佈建 LoadBalancer 之後，必須再次更新 KOTS 組態，以提供負載平衡器的主機名稱或 IP 地址。
- NodePort
 - 在每個呼叫節點上公開 NodePort 服務，做為呼叫流量的進入點。必須提供解析為一或多個節點的主機名稱或一或多個節點的 IP 地址。對於 UDP 和選用的 TCP 流量，您可以選擇從 30000-32767 的連接埠範圍。
- 現有的 NLB
 - 將呼叫輸入服務連接到現有的 NLB。您需要為 UDP 和選用的 TCP 流量提供目標群組 ARN。
- 無服務
 - 如果您不需要額外的 Kubernetes 服務來允許輸入流量，請選取此選項。這通常會與主機網路設定搭配使用，以將呼叫傳入流量直接路由至呼叫節點。

考量事項

- 為確保啟用呼叫輸入時，舊版呼叫模式仍可使用 (直接連線至呼叫伺服器)，以確保與較舊的用戶端和聯合網路回溯相容性，而無需呼叫輸入。如果變更任何預設連接埠，請確保您在呼叫節點上沒有任何連接埠衝突。
- 提供 UDP 流量的雙堆疊 NLBs 必須具有 IPv6 後端目標。如需詳細資訊，請參閱 [Network Load Balancer 目標群組](#)。
- 如果您需要 STIG 合規，則必須停用用於呼叫的主機網路選項。如果節點是在雙堆疊模式中設定，但叢集不是，您可能會失去 IPv6 連線 (假設是 IPv4 叢集)。
- 呼叫輸入需要預先定義的主機名稱或 IP 地址。擴展節點或提供自訂路由可能需要修改組態。
- 預設呼叫輸入連接埠的 TCP 為 8443，UDP 為 16384。確定防火牆和安全群組允許這些連接埠的流量，如果覆寫預設值，則為替代連接埠。

參考架構

使用負載平衡器傳入

此選項公開單一負載平衡器做為所有呼叫流量的進入點。

1. 針對呼叫輸入類型，選擇Load Balancer或現有的 NLB。如需現有 NLB 的詳細資訊，請參閱 GitHub 上 [Wickr Enterprise CDK 範例中的 NLB 堆疊](#)。
2. 根據呼叫輸入類型，執行下列其中一項操作：
 - 針對現有的 NLB，提供 UDP 和 TCP 流量的目標群組 ARNs，以及 NLB 的主機名稱。
 - 對於Load Balancer，請在 Kubernetes 佈建主機名稱之後提供主機名稱。

或者，對於呼叫輸入類型，您可以提供負載平衡器的 IP 地址或指向負載平衡器的自訂主機名稱。

3. （選用）若要在單一 NLB 下合併簡訊和通話流量，請在輸入區段中選擇現有 NLB，並提供 HTTPS 目標群組。

使用 NodePort 傳入

如果主機聯網已停用，而且您不想公開額外的負載平衡器，此選項很有用。

Note

確保您的防火牆和安全群組允許 NodePorts 的流量。

1. 針對呼叫輸入類型，選擇 NodePort。
2. 新增呼叫節點主機名稱或 IP 地址。
3. 停用呼叫主機網路。

使用 HostNetwork 直接輸入

此選項不會公開任何其他 Kubernetes 服務，並允許呼叫輸入流量直接透過呼叫節點的主機網路進行連線。如果需要 IPv6 連線，則偏好此方法。

1. 針對呼叫輸入類型，選取無服務。
2. 新增呼叫節點主機名稱或 IP 地址。

3. 啟用呼叫主機網路。

Kubernetes 叢集自動擴展器（選用）

Kubernetes Cluster Autoscaler 是 Wickr Enterprise 安裝的選用組態值。如果流量增加或其他資源限制可能導致效能不佳，這將有助於擴展 Kubernetes 節點群組。

Wickr Enterprise 安裝支援 3 個雲端提供者整合：AWS、Google Cloud 和 Azure。每個雲端供應商對此整合都有不同的需求。請遵循以下特定雲端供應商的說明來啟用此功能。

AWS

如果您未使用 WickrEnterpriseCDK 在上安裝 Wickr 環境 AWS，則需要採取一些額外的步驟來啟用 Cluster Autoscaler。

- 將下列標籤新增至節點群組。這可讓 Cluster Autoscaler 自動探索適當的節點。
 - `k8s.io/cluster-autoscaler/clusterName` = owned 其中 clusterName 是您 Kubernetes 叢集的名稱
 - `k8s.io/cluster-autoscaler-enabled` = true
 - 在 kube-system 命名空間中新增 Kubernetes 服務帳戶，並將其與允許自動擴展和 ec2 動作的 IAM 政策建立關聯。如需詳細資訊和詳細說明，請參閱《Amazon EKS 使用者指南》中的[設定 Kubernetes 服務帳戶以擔任 IAM 角色](#)。
- 設定服務帳戶時，您將需要使用「kube-system」命名空間
 - 下列政策可用於服務帳戶：

JSON

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

```

        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstances",
        "ec2:DescribeLaunchTemplateVersions"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
}

```

在設定 Cluster Autoscaler 時的複寫 UI 中，選取 AWS 做為您的雲端提供者，並提供您在上面建立的服務帳戶名稱，指示 Cluster Autoscaler 利用該服務帳戶。

Google 雲端

強烈建議對 Autopilot 和標準叢集使用 GKE 內建的 Autoscaling 功能。不過，如果您想要繼續此整合，在繼續之前必須符合下列要求。

使用要求：

1. 必須使用安全範圍建立受管執行個體群組 (MIG)，包括至少運算引擎資源的「讀取/寫入」。這目前無法新增至 MIG。
2. 叢集必須啟用工作負載聯合身分。您可以執行下列動作，在現有叢集上啟用此功能：
`gcloud container clusters update ${CLUSTER_NAME} --workload-pool=${PROJECT_ID}.svc.id.goog`
3. 可存取角色 `roles/compute.instanceAdmin.v1`。您可以使用下列指示來建立：

```

# Create GCP Service Account
gcloud iam service-accounts create k8s-cluster-autoscaler

# Add role to GCP Service Account
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
--member "serviceAccount:k8s-cluster-autoscaler@${PROJECT_ID}.iam.gserviceaccount.com" \
--role "roles/compute.instanceAdmin.v1"

# Link GCP Service Account to Kubernetes Service Account
gcloud iam service-accounts add-iam-policy-binding k8s-cluster-autoscaler@
${PROJECT_ID}.iam.gserviceaccount.com \

```

```
--role roles/iam.workloadIdentityUser \  
--member "serviceAccount:${PROJECT_ID}.svc.id.goog[kube-system/cluster-autoscaler-gce-  
cluster-autoscaler]"
```

Azure

Azure Kubernetes Service (AKS) 為大多數部署提供整合式叢集自動擴展，強烈建議將這些方法用於叢集自動擴展。不過，如果您的需求使得這些方法無法運作，我們已為 Azure Kubernetes Service 提供 Kubernetes Cluster Autoscaler 整合。若要利用此整合，您需要收集下列資訊，並在選取 Azure 做為雲端提供者之後，將其放入 Cluster Autoscaler 下的 KOTS 管理員面板組態中。

Azure 身分驗證

訂閱 ID：訂閱 ID 可透過 Azure 入口網站，依照官方文件取得。如需詳細資訊，請參閱 [Azure 入口網站中的取得訂閱和租戶 IDs](#)。

您可以使用 az 命令列公用程式建立 AD Service Principal，以取得下列參數。

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

應用程式 ID：

用戶端密碼：

租戶 ID：

Azure Cluster Autoscaler 組態

除了身分驗證要求之外，下列欄位是叢集自動擴展器正常運作的必要欄位。為方便起見提供了取得此資訊的命令，但根據您的特定 AKS 組態，可能需要一些修改。

Azure 受管節點資源群組：此值是您建立 AKS 叢集時由 Azure 建立的受管資源群組，而不是您定義的資源群組。若要取得此值，您需要在建立叢集時從取得 CLUSTER_NAME 和 RESOURCE_GROUP。取得這些值後，您可以執行下列動作來取得此值：

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

應用程式節點集區 VMSS 名稱：這是與 Wickr 應用程式之 AKS 節點集區相關聯的虛擬機器擴展集 (VMSS) 名稱。這是將根據您的叢集需求向上或向下擴展的資源。若要取得此值，您可以執行下列 az 命令：

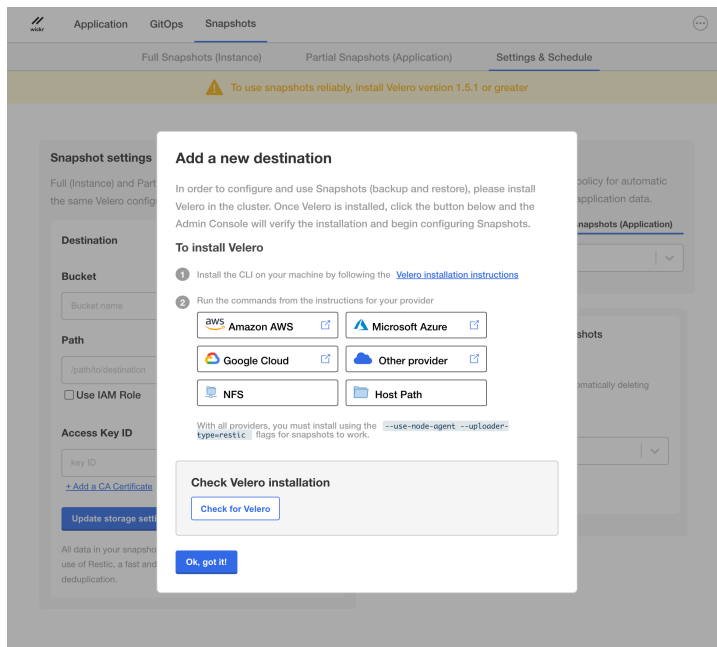
```
CLUSTER_NODEPOOL_NAME="(Your-NodePool-Name)"
CLUSTER_RESOURCE_GROUP="(Your-Managed-Node-Resource-Group-As-Defined-Above>)"
az vmss list -g ${CLUSTER_RESOURCE_GROUP} --query '[?tags."aks-managed-poolName"==`''`${CLUSTER_NODEPOOL_NAME}`''`].{VMSS_name:name}' -o tsv
```

ACalling Node Pool VMSS 名稱（選用）：如果您有 VMSS，這是與呼叫 Nodepool 相關聯的 VMSS 名稱。若要取得此值，您可以針對應用程式節點集區 VMSS 名稱執行修改版本的命令，以切換呼叫節點集區的節點集區名稱的 CLUSTER_NODEPOOL_NAME 值。

備份

Wickr Enterprise 將 Velero 用於備份目的。Velero 提供必要的工具來備份和還原 Kubernetes 叢集資源和持久性磁碟區，無論是在雲端供應商或內部部署操作。

使用 Minio 的 Velero 備份：目前 Velero 備份僅在低資源模式下為 Minio 啟用。



使用 Velero 文件進行安裝

- 安裝 Velero CLI。如需詳細資訊，請參閱[安裝 Velero CLI](#)。
- 在叢集上安裝 Velero，並根據提供者設定儲存體：
 - [AWS](#)。
 - [GCP](#)。
 - [Azure](#)。

- [其他供應商](#)。

限制

根據預設，備份中不會包含磁碟區。如果任何 Pod 掛載應備份的磁碟區，您必須使用列出要包含在備份中的特定磁碟區的註釋來設定備份。

對於每個需要備份的磁碟區，新增 `backup.velero.io/backup-volumes` 註釋。註釋名稱為 `backup.velero.io/backup-volumes : //`。如需詳細資訊，請參閱[設定快照](#)。

Airgap 安裝

Wickr Enterprise 和 KOTS 都支援部署到完全隔離的 Kubernetes 叢集。您必須提供私有 Docker 映像登錄檔的存取權，該登錄檔可從氣隙隔離的 Kubernetes 叢集存取。提供給 KOTS 的 Private Docker Image Registry 必須使用使用者名稱/密碼身分驗證進行保護，才能為此正確運作。KOTS 將利用私有 Docker 映像登錄檔來託管所有 Wickr Enterprise 映像。

- 啟用氣隙的 Wickr Enterprise `license.yaml`（聯絡 Wickr 銷售或客戶支援團隊）
- Wickr Enterprise `wickr.airgap` 封存套件（聯絡 Wickr 銷售或客戶支援團隊）
- 存取[私有 Docker 映像登錄檔](#)。
- 存取在 Airgap 環境中部署的 [Kubernetes 叢集](#)。
- 已安裝 [Kubectl](#)。
- 已安裝 [KOTS CLI](#)。
- [kotsadm.tar.gz](#) 已下載。

執行下列命令，在氣隙隔離的 kubernetes 叢集上部署 KOTS 和 Wickr Enterprise。這些命令會將 KOTS 管理員映像和 Wickr Enterprise 映像上傳至私有 Docker 映像登錄檔。命令完成後，系統會提示您存取 KOTS 管理員主控台，以完成上述的 Wickr Enterprise 安裝。

```
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 安裝的行動通知

從伺服器後端到行動用戶端的推送通知需要額外的聯網允許清單。此需求是由於 Apple iOS 和 Google Android 如何為離線和背景裝置實作此功能。請參閱這些服務的文件，並允許列出指定的 IP 地址和連接埠。

- [iOS](#)
- [Android](#)

Wickr 管理員主控台

Wickr Admin 主控台界面用於管理 Wickr Enterprise 應用程式本身。它可用於設定網路、使用者、聯合等等。您可以在設定為指向 Load Balancer 的 DNS 名稱上，透過 HTTPS 存取它。預設使用者名稱為 admin，密碼為 Password123。您必須在第一次登入時變更此密碼。



Network Admin Sign In

Sign In With SSO

or

Username

Password

Remember Me

SIGN IN

[Server Open Source Licenses](#)
[Admin Console Open Source Licenses](#)

安全性設定

AWS Wickr Enterprise 提供組態設定，為您的部署強制執行增強型安全內容。此更高的安全標準適用於 Pod 和容器層級，因此需要符合 安全技術實作指南 (STIG)。

設定下列組態參數以強制執行增強型安全內容：

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

Warning

對於 Opensearch，此安全組態會停用更新持久性儲存體許可的 `fsgroup-volume` `initContainer`，這可能會導致與許可相關的相容性問題。

常見問答集

問：我的部署失敗，在 `helm stderr` 中出現下列錯誤：

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

答：啟用偵錯記錄時可能會發生這種情況。請停用偵錯記錄、刪除有問題的任務，然後再試一次。

Wickr Enterprise 的內嵌叢集

Wickr Enterprise 的內嵌叢集安裝選項為 Wickr Enterprise 產品提供小型且高效率的安裝產品。它利用複寫的內嵌叢集，使用可安裝 Wickr Enterprise 的 k0s 提供小型 Kubernetes 安裝。使用此安裝方法可降低 Wickr Enterprise 安裝的技術技能需求和整體硬體需求，方法是以彈性和高可用性的成本提供「all-in-one」解決方案。

主題

- [Wickr Enterprise 內嵌叢集入門](#)
- [Wickr Enterprise 內嵌叢集需求](#)
- [安裝 Wickr Enterprise 內嵌叢集 \(標準\)](#)
- [多節點安裝](#)
- [KOTS 管理員主控台組態](#)
- [其他常見安裝需求](#)
- [故障診斷 Wickr 內嵌叢集安裝](#)

Wickr Enterprise 內嵌叢集入門

若要開始使用 Wickr Enterprise 內嵌叢集選項，請聯絡 Support 以取得授權。如果您有現有的授權，並想要使用此選項，請聯絡 支援以取得更新現有授權的協助和其他安裝指示。

Wickr Enterprise 內嵌叢集需求

開始安裝 Wickr Enterprise 內嵌叢集之前，請確認符合下列要求。

網路需求

您將需要允許在下列連接埠輸入 Wickr 伺服器：

- HTTPS 443/TCP
- 僅限呼叫 TCP Proxy - 為 KOTS 中的 TCP 呼叫流量設定的 TCP 代理連接埠
- UDP 呼叫流量為 16384-19999/UDP
- 僅限 LAN - 用於存取 KOTS 管理員主控台的 30000/TCP

系統要求

安裝之前，請確定您有一個 VM（虛擬機器）或執行 Linux 作業系統 (OS) 的實體機器，並提供下列最低可用資源：

- 8 個 CPU 核心
- 12 GB (GB) 的 RAM
- / (根) 分割區上的 100 GB (GB) 磁碟儲存體

Wickr Enterprise 內嵌叢集已在下列 Linux 作業系統上經過測試，但其他以 Linux 為基礎的作業系統選項可能也適用：

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

安裝 Wickr Enterprise 內嵌叢集（標準）

取得下載指示後，請將 Wickr Enterprise 套件下載到目的地機器，然後解壓縮。

```
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
```

您現在應該有兩個檔案 `wickr-enterprise-ha` 和 `license.yaml`。`wickr-enterprise-ha` 檔案是一個二進位檔案，其中包含安裝內嵌叢集所需的所有必要組件，而 `license.yaml` 是您的 Wickr 授權，將用於驗證您的安裝。

您可以執行 `wickr-enterprise-ha` 檔案，在此階段執行基本安裝：

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

安裝程序開始後，系統會提示您輸入管理員主控台密碼。輸入安全密碼，並確保您將其儲存為存取 KOTS 管理員主控台以繼續設定安裝時所需的密碼。

安裝完成後，輸出類似以下內容：

```
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
```

標準安裝之後，請使用 Web 瀏覽器前往輸出中提供的 KOTS 管理員主控台 URL。在此範例中，URL 為 `http://192.168.1.100:30000`。不過，您的 URL 會根據您的聯網組態而有所不同。

多節點安裝

Wickr Enterprise Embedded Cluster Multi-Node 安裝可為 Embedded Cluster 使用者提供將 Wickr Calling 和 Wickr Messaging 工作負載分開到不同實體機器的選項。若要這樣做，Wickr Enterprise 會利用複寫的內嵌叢集多節點工具。

連接埠需求

下列連接埠必須在叢集的所有成員上開啟，多節點功能才能正常運作。這些只需要在節點本身之間開啟，而不是開放給更廣泛的網際網路。

- 53 TCP/UDP
- 2380/TCP
- 4789/UDP
- 6443/TCP
- 8080/TCP
- 9091/TCP
- 9443/TCP

- 10249/TCP
- 10250/TCP
- 10256/TCP
- 30000/TCP
- 50000/TCP

授權需求

Wickr 內嵌叢集多節點組態選項需要額外的授權權限。請聯絡 [支援](#) 以確保您的授權支援此功能。

在初始設定期間建立其他節點

當您最初設定 Wickr Enterprise Embedded Cluster 時，您可以在設定程序期間建立額外的呼叫節點。首先遵循[安裝 Wickr Enterprise 內嵌叢集 \(標準\)](#) 中所述的程序。當您導覽至 KOTS 管理員面板時，系統會提示您建立其他節點。

Note

目前，內嵌叢集多節點僅支援 1 個呼叫節點和 1 個簡訊/控制器節點。

若要開始，請取消選取控制器角色選項，然後選取呼叫角色選項。這會填入用於設定新節點的其他指示集。在新節點上執行這些指示，以將其設定為將叢集加入為呼叫節點。

在新節點上執行類似下列範例的說明：

1. 在新節點上下載二進位檔：

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

2. 擷取二進位檔：

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

3. 將節點加入叢集：

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

在聯結命令成功完成之後，新的節點會出現在設定叢集頁面上，並指派呼叫角色。選擇繼續以繼續前往 Wickr Enterprise 組態頁面。遵循 [KOTS 管理員主控台組態中概述的內嵌節點組態](#) 選項的指示。

將其他節點新增至現有的內嵌叢集安裝

若要將呼叫節點新增至現有的 Wickr Enterprise Embedded Cluster 安裝，請導覽至 KOTS 管理員主控台。若要這樣做，請透過 ssh 或其他機制登入節點，然後導覽至包含用於安裝之 wickr-enterprise-ha 二進位檔的安裝目錄。執行 `./wickr-enterprise-ha admin-console` 以啟動 KOTS 管理員主控台。如果此命令未傳回任何輸出，則 KOTS 管理員主控台已在執行中，可以透過 Web 瀏覽器中導覽至節點 IP 上的連接埠 30000 來存取，例如：`https://127.0.0.1:30000/`。

要求時輸入 KOTS Admin 密碼，然後執行下列程序來建立額外的節點：

1. 登入後，導覽至 KOTS 管理主控台左上方的叢集管理頁面。
2. 選擇 Add node (新增節點)。
3. 在下取消選取控制器 Roles。
4. 選取 下的呼叫 Roles
5. 遵循提供的指示，在您要新增的新節點上執行命令。
6. 完成後，選擇關閉
7. 您的新節點會出現在具有 呼叫角色的節點清單中。
8. 導覽至 KOTS 管理員主控台左上角的應用程式頁面
9. 從頁面頂端的導覽列中選擇組態。
10. 導覽至左側導覽面板中的呼叫區段。
11. 選取需要呼叫節點以允許使用呼叫節點。
12. 捲動至頁面底部，然後選擇儲存組態。
13. 隨即出現快顯視窗，指出 Config 已更新。選擇移至更新版本。
14. 在更新版本頁面上，會顯示目前安裝的版本。新的明細項目會列在名為 Config Change 的已安裝版本下。選擇部署以部署此新版本，並啟用新的呼叫節點。

KOTS 管理員主控台組態

KOTS 管理員主控台一開始會使用自我簽署的憑證，您需要在瀏覽器中允許 做為例外狀況。接受此例外狀況後，KOTS 管理員主控台的組態精靈會歡迎您。此精靈會引導您完成設定 KOTS 管理員主控台行為的其他設定步驟，包括在必要時新增自訂憑證的選項。

KOTS 管理員主控台的初始組態完成後，系統會提示您輸入在安裝程序期間建立的管理員主控台密碼。第一次登入時，您需要設定叢集。

選擇繼續以前往 Wickr 的 KOTS 管理員主控台。

對於單一節點內嵌叢集，選擇繼續以前往 Wickr 的 KOTS 管理員主控台。如需多節點安裝，請參閱[多節點安裝](#)。

在 KOTS 管理員主控台中，根據您的需求設定您的安裝。使用內嵌叢集產品時，應該設定一些金鑰組態設定，以確保 Wickr Enterprise 安裝的適當功能。

- 主機名稱 - 這是您與 Wickr 安裝通訊時所使用的主機名稱。請務必為此網域建立適當的 DNS 記錄，以指向您的 Wickr Enterprise 安裝。
- 在進階選項下，檢查 **【】** 設定傳入控制器選項，以公開設定 Kubernetes 傳入的組態區塊。在輸入組態區塊中，選取單一節點內嵌叢集，然後在標記 Loadbalancer 外部 IP (僅限 IPv4) 的文字方塊中輸入與您的 Wickr 伺服器相關聯的「公有」IP。

如果您不確定此 IP 是什麼，您可以從 Wickr 伺服器上的命令列執行下列命令，以判斷此值：`ip route get 1.1.1.1|awk '{print $7}'`

- 在進階選項下，勾選啟用低資源模式選項。
- 在呼叫下，如果您使用的是單一節點內嵌叢集，請確定取消選取需要呼叫節點。否則，如果您已在初始設定期間新增呼叫節點，請確定已選取需要呼叫節點。
- 如果您想要在不使用外部資料庫或 S3 相容儲存體進行檔案共用的單一解決方案中，選取下列設定的內部選項：
 - 資料庫
 - S3 儲存位置

內部 S3 儲存位置提供設定儲存容量的其他選項。建議從小規模開始並視需要展開，因為佈建後不是選項。

設定所有必要功能後，請捲動至組態頁面底部，然後選擇儲存組態。這將啟動一些預檢主機檢查。完成預檢檢查後，請選擇部署以開始 Wickr Enterprise Installation。

現在您已準備好開始設定 Wickr Enterprise 安裝。如需設定 Wickr Enterprise 的詳細資訊，請參閱[什麼是 Wickr Enterprise ?](#)。

其他常見安裝需求

IP 主機名稱安裝

如果您的安裝需要 IP 型主機名稱，則有一些額外的組態選項。這些指示專屬於 IP 型主機名稱，建議您遵循上述基本設定的其他指示。

在 KOTS 管理員面板中，完成下列步驟。

1. 將主機名稱設定為您將使用的 IP。
2. 在憑證下，選取上傳憑證。然後，依照 IP 型憑證的指示產生自我簽署憑證。如需詳細資訊，請參閱[產生自我簽署憑證](#)。
3. 上傳憑證的 .crt 檔案和私有金鑰的 .key 檔案
4. 對於憑證鏈，請再次上傳.crt檔案。
5. 勾選設定固定憑證核取方塊。
6. 上傳 Pinned 憑證.crt的。
7. 在呼叫下，取消勾選自動探索伺服器公有 IP 地址和使用主機主要 IP 地址來呼叫流量核取方塊。
8. 在呼叫下，將主機名稱的 IP 地址放在主機名稱覆寫文字方塊中。
9. 在進階選項下，勾選設定傳入控制器核取方塊。下方會顯示名為輸入的新組態區段。
10. 在輸入下，選取單一節點內嵌叢集。
11. 在輸入下，輸入 Wickr 伺服器上 'public' 介面的 IP。這可能與用作主機名稱的 IP 不同。請參閱基本組態步驟中有關此值的其他資訊。
12. 在輸入下，勾選使用萬用字元主機名稱。

SELinux 強制執行模式

如果您需要在強制執行模式中使用 SELinux，請修改用來安裝內嵌叢集的預設資料目錄。建議使用 `/opt` 因為它已通過測試，可與此使用案例的大多數 SELinux 政策搭配使用。

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

複寫的內嵌叢集預設安裝預檢會嘗試驗證 SELinux 處於寬鬆模式，並在 SELinux 處於強制執行時失敗。若要略過此項目，必須使用 `--ignore-host-preflights` 命令列引數。使用命令列選項時，以下提示相同。出現提示時輸入是。

```
# 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 安裝

Wickr Enterprise 的內嵌叢集安裝選項支援氣隙化安裝。授權需要額外的組態和啟用。如果您有興趣在氣隙隔離的環境中使用 Wickr Enterprise 內嵌叢集，請聯絡 支援。

執行氣隙安裝時，下載指示與標準安裝方法不同。它們應該類似以下內容：

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

將套件下載到可存取網際網路的機器，然後使用您偏好的資料運輸方法將其傳輸到氣隙環境。傳輸套件後，請像使用任何標準安裝套件一樣擷取套件。將包含包含 `wickr-enterprise-ha.airgap` 所有相關聯 Wickr Enterprise 應用程式服務映像的第三個檔案。

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

在安裝期間，必須在擷取後設定 `--airgap-bundle` 命令列引數；否則，程序會遵循標準安裝程序。

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

更新 airGapped 內嵌叢集

若要更新 AirGapped Embedded 叢集，請完成下列步驟。

1. 從已複製下載新的內嵌叢集套件，並使用您氣隙環境的標準資料傳輸方法來將其傳輸到主機機器。新套件在主機電腦上後，擷取 tarball：

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

2. 使用新的二進位和 Airgap 套件執行更新：

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

3. 啟動 KOTS 管理員主控台，並使用您存取 KOTS 管理員主控台的標準方法登入提供的 URL

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

4. 登入 KOTS 管理員主控台後，請在版本 下方找到左側的最新可用更新，然後按前往版本歷史記錄按鈕。
5. 在可用更新下，為新版本選擇部署。逐步解說畫面：
 1. 變更任何組態選項、向下捲動，然後選擇下一步。
 2. 確認沒有預檢檢查失敗，選擇下一步：確認並部署。
 3. 選擇部署。

Wickr Enterprise 內嵌叢集的其他備註

- **NAMESPACE**：與大多數 Wickr Enterprise 安裝不同，嵌入式叢集安裝會將 Wickr 資產安裝到 kubernetes 中的 kotsadm 命名空間，而不是 wickr。修改您儲存 -n wickr 用於 kubectl、helm 或任何其他公用程式的任何指令碼或命令，以 -n kotsadm 改用。
- **與 Kubernetes 叢集互動**：從主機機器使用 ./wickr-enterprise-ha 二進位來建立具有適當變數設定的 shell，以透過執行與 Kubernetes 安裝互動 ./wickr-enterprise-ha shell。這將在 shell 的 PATH 中提供 kubectl 公用程式，並將適當的 kube 組態設定為本機安裝。

故障診斷 Wickr 內嵌叢集安裝

這些疑難排解步驟的所有執行個體都假設您擁有執行 Wickr Embedded Cluster 安裝的執行個體的 shell 存取權，並已執行 `./wickr-enterprise-ha shell` 命令，以便能夠直接與 Kubernetes 安裝互動。

一般問題

從叢集管理畫面新增遺失的節點按鈕

封裝的安裝

如果您正在進行空隙安裝，請聯絡 Wickr Support 尋求協助以修正此行為。

標準安裝

如果您的授權包含內嵌叢集多節點權利，請執行授權同步以取得最新版本。如果您不確定或沒有此權利，請聯絡 Wickr Support。

若要執行授權同步，請完成下列步驟。

1. 導覽至 KOTS 控制面板。
2. 在儀表板頁面上，找到頁面右上角的授權區段。
3. 在本節的右上角，您應該會看到同步授權超連結。選取超連結。
4. 授權同步後，UI 更新和幾秒鐘前的上次同步隨即出現。
5. 從 KOTS 儀表板頁面的版本區段中選擇重新部署。
6. 重新部署完成後，導覽回叢集管理，您就可以新增節點。

升級問題

升級叢集上的升級卡住

如果您的升級卡在升級叢集上，這可能表示某些 Pod 未適當終止。登入執行個體並使用 `./wickr-enterprise-ha shell` 命令來進入 Shell 環境，以管理 kubernetes 安裝。

1. 識別仍在執行的 Pod：

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

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

Note

如果其中一個執行中的 Pod 是 `embedded-cluster-upgrade-XXXXXXXXXXXXXXXX-xxxxx kotsadm-xxxxxxx` 或類似，請勿將其刪除，因為這些 Pod 是執行升級的必要項目。

3. 確認沒有剩餘的執行中 Pod。

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

此程序應允許叢集升級透過 Wickr 升級繼續進行。

應用程式未在叢集升級期間更新，且無法部署新版本

如果應用程式在升級後仍保留在舊版本上，則新版本可能處於不一致的狀態。

檢查 Kubernetes 安裝記錄：

1. 從安裝程式開啟 Kubernetes shell。

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

2. 執行下列 `kubectl` 命令：

```
kubectl get installations
```

3. 輸出看起來像這樣：

```
[root@ip-172-31-6-72 ~]# kubectl get installations
NAME                STATE      INSTALLERVERSION  CREATEDAT                AGE
20251113170603      Obsolete   2.1.3+k8s-1.30    2025-11-13T17:06:05Z    22h
20251113180133      Failed     2.6.0+k8s-1.31    2025-11-13T18:01:37Z    21h
```

4. 刪除失敗的安裝。

```
kubectl delete installation 20251113180133
```

5. 嘗試透過 KOTS Admin 面板再次執行升級。

RabbitMQ Pod 日誌行失敗 **Error while waiting for Mnesia tables: {timeout_waiting_for_tables}**

RabbitMQ 秘密和儲存體不同步。這通常發生在多個 RabbitMQ 執行個體執行並導致領導者選擇或規定人數錯誤時。若要修正此問題，請刪除 RabbitMQ 服務及其儲存磁碟區，然後重新部署。

若要刪除失敗的 RabbitMQ，請完成下列步驟。

1. 刪除 RabbitMQ 狀態集。

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

2. 刪除剩餘的 RabbitMQ Pod。如果有多個 RabbitMQ-X Pod 正在執行，請多次發出此命令，更新 RabbitMQ-X 值以對應至其他 Pod 名稱。

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

3. 刪除對應的 PVCs。如果有多個 Pod 正在執行，請多次發出此命令，更新 data-RabbitMQ-X 以對應至適當的 Pod。

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

4. 檢查是否有任何剩餘的 Pod，如果成功，這應該不會輸出任何內容。

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

5. 檢查是否有任何剩餘的 PVCs，如果成功，這應該不會輸出任何內容。

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

6. 透過 KOTS 管理面板重新部署。

如需更多故障診斷資訊，請參閱[故障診斷](#)。

文件歷史記錄

下表說明 Wickr Enterprise 自動化安裝指南的文件版本。

變更	描述	日期
安全性設定	已新增安全性設定。如需詳細資訊，請參閱 安全性設定 。	2025 年 8 月 26 日
多節點安裝	已新增多節點安裝。如需詳細資訊，請參閱 多節點安裝 。	2025 年 8 月 26 日
呼叫輸入設定	已新增呼叫輸入設定。如需詳細資訊，請參閱 呼叫輸入設定 。	2025 年 8 月 26 日
自動部署選項	已新增自動部署選項。如需詳細資訊，請參閱 安裝 Wickr Enterprise 。	2024 年 2 月 23 日
允許清單的连接埠	连接埠 TCP/8443 已新增至允許清單。如需詳細資訊，請參閱 需求 。	2024 年 2 月 12 日
銷毀資源和连接埠以允許清單	已新增如何銷毀資源的說明。如需詳細資訊，請參閱 銷毀資源 。此外，已新增要允許清單的连接埠。如需詳細資訊，請參閱 需求 。	2023 年 8 月 17 日
初始版本	Wickr Enterprise 自動化安裝指南的初始版本	2023 年 8 月 4 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。