aws

Panduan Instalasi Otomatis

# Perusahaan Wickr

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

# Perusahaan Wickr: Panduan Instalasi Otomatis

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

# Table of Contents

Apa itu Wickr Enterprise? ..... 1
Mulai menggunakan ..... 2
Persyaratan ..... 2
Instal dependensi ..... 3
Konfigurasi ..... 4
mengebut ..... 7
Deploy ..... 7
Hasilkan KOTS Config ..... 8
Menghubungkan ke Kubernetes ..... 9
Memproksi koneksi melalui benteng ..... 9
Menginstal Wickr Enterprise ..... 11
Menginstal Wickr Enterprise secara manual ..... 11
Instalasi Wickr Enterprise dengan Lambda ..... 11
Instalasi pasca ..... 12
Konsol Admin KOTS ..... 12
Konsol Admin Wickr ..... 13
Nilai konteks ..... 14
Menghancurkan sumber daya ..... 18
Pemecahan Masalah ..... 19
Menghapus namespace Wickr ..... 19
Menyetel ulang kata sandi Konsol Admin KOTS ..... 19
Masalah menghubungkan ke cluster EKS dengan benteng ..... 19
Instalasi kustom ..... 21
Persyaratan ..... 21
Persyaratan Perangkat Keras ..... 21
Persyaratan perangkat lunak ..... 24
Persyaratan jaringan ..... 24
Arsitektur ..... 25
Penginstalan ..... 27
Konsol Admin KOTS ..... 27
Pengaturan masuk ..... 27
Pengaturan Database ..... 28
Pengaturan Database Eksternal ..... 29
Pengaturan Database Internal ..... 29

Tingkatkan ke MySQL 8.0 ..... 30
Penyimpanan file S3 ..... 31
Pengaturan klaim volume persisten ..... 32
Pengaturan sertifikat TLS ..... 33
Mari Enkripsi ..... 33
Sertifikat yang disematkan ..... 33
Penyedia Sertifikat ..... 33
Menghasilkan sertifikat yang ditandatangani sendiri ..... 34
Pengaturan panggilan ..... 34
Memanggil pengaturan ingress ..... 35
Pertimbangan-pertimbangan ..... 36
Arsitektur referensi ..... 36
Autoscaler cluster Kubernetes (opsional) ..... 38
AWS ..... 38
Awan Google ..... 39
Biru langit ..... 40
Pencadangan ..... 41
Instalasi menggunakan dokumentasi Velero ..... 42
Instalasi airgap ..... 43
Pemberitahuan seluler untuk pemasangan airgap ..... 44
Konsol admin Wickr ..... 44
Pengaturan keamanan ..... 45
Pertanyaan yang Sering Diajukan ..... 45
Instalasi cluster tertanam ..... 46
Memulai ..... 46
Persyaratan ..... 46
Instalasi standar ..... 47
Instalasi Multi-Node ..... 48
Persyaratan port ..... 49
Persyaratan lisensi ..... 49
Membuat node tambahan selama penyiapan awal ..... 49
Menambahkan node tambahan ke instalasi cluster tertanam yang ada ..... 50
Konfigurasi konsol admin KOTS ..... 51
Persyaratan instalasi tambahan ..... 52
Memecahkan masalah instalasi cluster tertanam ..... 56
Masalah umum ..... 56

Masalah peningkatan ..... 57
Riwayat dokumen ..... 60
..... lxii

# Apa itu Wickr Enterprise?

Wickr Enterprise adalah layanan end-to-end terenkripsi yang dihosting sendiri yang membantu organisasi dan lembaga pemerintah untuk berkomunikasi dengan aman melalui dan mengelompokkan pesan, panggilan suara one-to-one dan video, berbagi file, dan berbagi layar. Pelanggan dapat menggunakan Wickr Enterprise untuk mengatasi kewajiban penyimpanan data yang terkait dengan aplikasi perpesanan tingkat konsumen, dan memfasilitasi kolaborasi dengan aman. Kontrol keamanan dan administratif tingkat lanjut membantu organisasi memenuhi persyaratan hukum dan peraturan, dan membangun solusi khusus untuk tantangan keamanan data.

Informasi dapat dicatat ke penyimpanan data pribadi yang dikendalikan pelanggan untuk tujuan retensi dan audit. Pelanggan memiliki kontrol administratif yang komprehensif atas data, yang mencakup pengaturan izin, mengonfigurasi opsi pesan singkat, dan mendefinisikan grup keamanan. Administrator juga dapat mengotomatiskan alur kerja dengan aman dengan bot Wickr. Wickr Enterprise terintegrasi dengan layanan tambahan seperti Active Directory dan single sign-on (SSO) dengan OpenID Connect (OIDC). Untuk mulai mengkonfigurasi Wickr Enterprise, lihat Memulai dengan Wickr Enterprise.

> **Note**
>
> Jika Anda belum memiliki paket penyebaran Wickr Enterprise, lihat Hubungi Kami untuk pertanyaan bisnis.

# Memulai dengan Wickr Enterprise

Topik


- Persyaratan
- Instal dependensi
- Konfigurasi
- mengebut
- Deploy
- Hasilkan KOTS Config

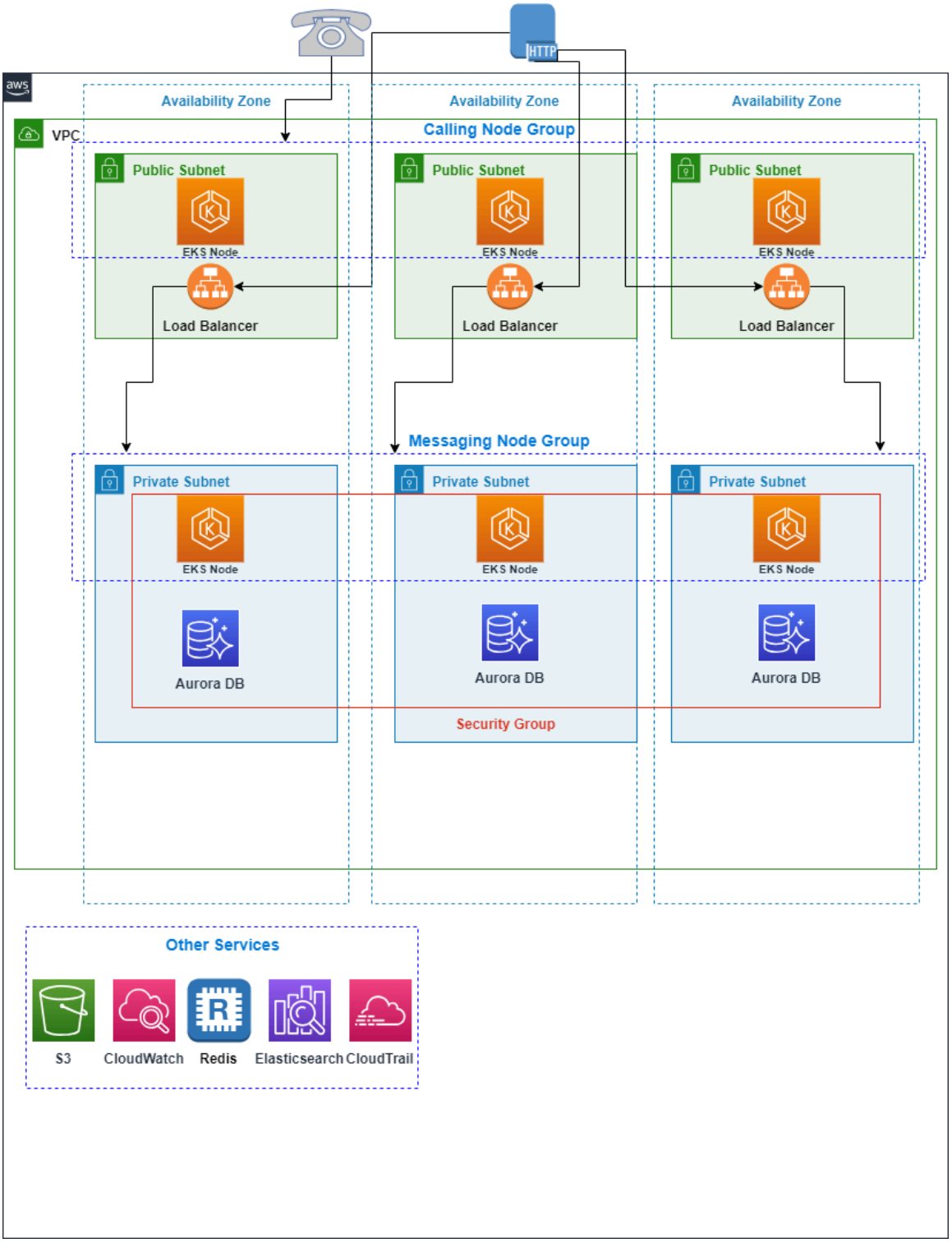

## Persyaratan

Sebelum Anda mulai, verifikasi bahwa persyaratan berikut terpenuhi:

- Unduh Node.js 16+
- AWS CLI dikonfigurasi dengan kredensil untuk akun Anda.

  Ini akan bersumber dari file konfigurasi Anda di `~/.aws/config` atau menggunakan variabel `AWS_` lingkungan.
- Instal kubectl. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui kubectl di Panduan Amazon. EKSUser
- Instal CLI kots. Untuk informasi selengkapnya, lihat Menginstal CLI kots.
- Port ke daftar yang diizinkan: 443/TCP untuk lalu lintas Panggilan HTTPS dan TCP; 16384-19999/UDP untuk lalu lintas Panggilan UDP; TCP/8443

Arsitektur




# Instal dependensi

Anda dapat menambahkan semua dependensi ke paket default dengan perintah berikut:

```
npm install
```

# Konfigurasi

AWS Cloud Development Kit (AWS CDK) menggunakan nilai konteks untuk mengontrol konfigurasi aplikasi. Wickr Enterprise menggunakan nilai konteks CDK untuk memberikan kontrol atas pengaturan seperti nama domain instalasi Wickr Enterprise Anda atau jumlah hari untuk menyimpan cadangan RDS. Untuk informasi selengkapnya, lihat konteks Runtime di Panduan AWS Cloud Development Kit (AWS CDK) Pengembang.

Ada beberapa cara untuk mengatur nilai konteks, tetapi kami sarankan untuk mengedit nilai `cdk.context.json` agar sesuai dengan kasus penggunaan khusus Anda. Hanya nilai konteks yang dimulai dengan `wickr/` yang terkait dengan penerapan Wickr Enterprise; sisanya adalah nilai konteks khusus CDK. Untuk menjaga pengaturan yang sama saat berikutnya Anda melakukan pembaruan melalui CDK, simpan file ini.

Minimal, Anda harus mengatur`wickr/licensePath`,`wickr/domainName`, dan salah satu `wickr/acm:certificateArn` atau `wickr/route53:hostedZoneId` dan`wickr/route53:hostedZoneName`.

Dengan zona yang dihosting publik

Jika Anda memiliki zona host publik Route 53 Akun AWS, sebaiknya gunakan pengaturan berikut untuk mengonfigurasi konteks CDK Anda:

- `wickr/domainName`- Nama domain yang akan digunakan untuk penyebaran Wickr Enterprise ini. Jika Anda menggunakan zona host publik Route 53, catatan DNS dan sertifikat ACM untuk nama domain ini akan dibuat secara otomatis.
- `wickr/route53:hostedZoneName`- Route 53 nama zona host untuk membuat catatan DNS.
- `wickr/route53:hostedZoneId`- Route 53 ID zona host untuk membuat catatan DNS.

Metode ini membuat sertifikat ACM atas nama Anda, bersama dengan catatan DNS yang mengarahkan nama domain Anda ke penyeimbang beban di depan penerapan Wickr Enterprise Anda.

Tanpa zona yang dihosting publik

Jika Anda tidak memiliki zona host publik Route 53 di akun Anda, sertifikat ACM harus dibuat secara manual dan diimpor ke CDK menggunakan nilai `wickr/acm:certificateArn` konteks.

- `wickr/domainName`- Nama domain yang akan digunakan untuk penyebaran Wickr Enterprise ini. Jika Anda menggunakan zona host publik Route 53, catatan DNS dan sertifikat ACM untuk nama domain ini akan dibuat secara otomatis.
- `wickr/acm:certificateArn`- ARN dari sertifikat ACM untuk digunakan pada penyeimbang beban. Nilai ini harus diberikan jika zona yang dihosting publik Route 53 tidak tersedia di akun Anda.

Mengimpor sertifikat ke ACM

Anda dapat mengimpor sertifikat yang diperoleh secara eksternal dengan perintah berikut:

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

Outputnya adalah ARN Sertifikat, yang harus digunakan untuk nilai pengaturan `wickr/acm:certificateArn` konteks. Penting bahwa sertifikat yang diunggah valid untuk`wickr/domainName`, atau koneksi HTTPS tidak akan dapat memvalidasi. Untuk informasi selengkapnya, lihat Mengimpor sertifikat di Panduan AWS Certificate Manager Pengguna.

Buat catatan DNS

Karena tidak ada zona host publik yang tersedia, catatan DNS harus dibuat secara manual setelah penerapan selesai untuk menunjuk ke penyeimbang beban di depan penerapan Wickr Enterprise Anda.

Menerapkan ke VPC yang ada

Jika Anda memerlukan penggunaan VPC yang ada, Anda dapat menggunakannya. Namun, VPC harus dikonfigurasi untuk memenuhi spesifikasi yang diperlukan untuk EKS. Untuk informasi selengkapnya, lihat Lihat persyaratan jaringan Amazon EKS untuk VPC dan subnet di Panduan Pengguna Amazon EKS, dan pastikan bahwa VPC yang akan digunakan memenuhi persyaratan ini.

Selain itu, sangat disarankan untuk memastikan Anda memiliki titik akhir VPC untuk layanan berikut:

- CLOUDWATCH

- CLOUDWATCH_LOGS
- EC2
- EC2_PESAN
- ECR
- ECR_DOCKER
- ELASTIC_LOAD_BALANCING
- KM
- SECRETS_MANAGER
- SSM
- SSM_MESSAGES

Untuk menyebarkan sumber daya ke VPC yang ada, tetapkan nilai konteks berikut:

- `wickr/vpc:id`- ID VPC untuk menyebarkan sumber daya ke (misalnya). `vpc-412beef`
- `wickr/vpc:cidr`- IPv4 CIDR dari VPC (`172.16.0.0/16`mis.).
- `wickr/vpc:publicSubnetIds`- Daftar subnet publik yang dipisahkan koma di VPC. Application Load Balancer dan memanggil node pekerja EKS akan digunakan dalam subnet ini (misalnya). `subnet-6ce9941,subnet-1785141,subnet-2e7dc10`
- `wickr/vpc:privateSubnetIds`- Daftar subnet pribadi yang dipisahkan koma di VPC. Node pekerja EKS dan server bastion akan digunakan di subnet ini (misalnya). `subnet-f448ea8,subnet-3eb0da4,subnet-ad800b5`
- `wickr/vpc:isolatedSubnetIds`- Daftar subnet terisolasi yang dipisahkan koma di VPC. Database RDS akan digunakan dalam subnet ini (misalnya). `subnet-d1273a2,subnet-33504ae,subnet-0bc83ac`
- `wickr/vpc:availabilityZones`- Daftar zona ketersediaan yang dipisahkan koma untuk subnet di VPC (misalnya). `us-east-1a,us-east-1b,us-east-1c`

Untuk informasi selengkapnya tentang titik akhir VPC antarmuka, lihat Mengakses AWS layanan menggunakan titik akhir VPC antarmuka.

Pengaturan lainnya

Untuk informasi selengkapnya, lihat Nilai konteks.

# mengebut

Jika ini adalah pertama kalinya Anda menggunakan CDK di wilayah khusus Akun AWS ini, Anda harus mem-bootstrap akun terlebih dahulu untuk mulai menggunakan CDK.

```
npx cdk bootstrap
```

# Deploy

Proses ini akan memakan waktu sekitar 45 menit.

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

Setelah selesai, infrastruktur telah dibuat dan Anda dapat mulai menginstal Wickr Enterprise.

Buat catatan DNS

Langkah ini tidak diperlukan jika Anda menggunakan zona yang dihosting publik saat mengonfigurasi CDK.

Output dari proses penyebaran akan menyertakan nilai`WickrAlb.AlbDnsName`, yang merupakan nama DNS dari penyeimbang beban. Outputnya akan terlihat seperti:

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

Dalam hal ini, nama DNS adalah`Wickr-Alb-1Q5IBPJR4ZVZR-409483305.us-west-2.elb.amazonaws.com`. Itu adalah nilai yang harus digunakan saat membuat catatan CNAME atau A/AAAA (ALIAS) untuk nama domain Anda.

Jika Anda tidak memiliki output dari penerapan, jalankan perintah berikut untuk menampilkan nama DNS penyeimbang beban:

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

# Hasilkan KOTS Config

> **Warning**
>
> File ini berisi informasi sensitif tentang instalasi Anda. Jangan berbagi atau menyimpannya secara publik.

Installer Wickr Enterprise memerlukan sejumlah nilai konfigurasi tentang infrastruktur agar berhasil menginstal. Anda dapat menggunakan skrip pembantu untuk menghasilkan nilai konfigurasi.

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

Jika Anda mengimpor sertifikat eksternal ke ACM pada langkah pertama, berikan `--ca-file` tanda ke skrip ini, misalnya:

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

Jika Anda menerima kesalahan yang mengatakan tumpukan tidak ada, setel variabel `AWS_REGION` lingkungan (`export AWS_REGION=us-west-2`) ke Wilayah yang Anda pilih dan coba lagi. Atau, jika Anda menetapkan nilai konteks`wickr/stackSuffix`, berikan sufiks dengan `--stack-suffix` bendera.

# Menghubungkan ke klaster Kubernetes

Amazon EKS API hanya dapat diakses melalui host bastion yang dibuat sebagai bagian dari penerapan. Akibatnya, semua `kubectl` perintah harus dijalankan pada host bastion itu sendiri atau diproksi melalui host bastion.

## Memproksi koneksi melalui benteng

Pertama kali Anda terhubung ke cluster, Anda harus memperbarui file kubeconfig lokal Anda menggunakan `aws eks update-kubeconfig` perintah, dan kemudian mengatur konfigurasi Anda. `proxy-url` Kemudian, setiap kali Anda ingin terhubung ke cluster, Anda memulai sesi SSM dengan host bastion untuk meneruskan ke proxy untuk akses API.

Pengaturan satu kali

Ada nilai output pada `WickrEks` CloudFormation tumpukan dengan nama yang dimulai dengan`WickrEnterpriseConfigCommand`. Nilai tersebut berisi perintah lengkap yang diperlukan untuk menghasilkan konfigurasi kubectl untuk klaster Anda. Output ini dapat dilihat dengan perintah berikut:

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

Ini harus menampilkan perintah yang dimulai dengan`aws eks update-kubeconfig`. Jalankan perintah ini.

Selanjutnya, konfigurasi Kubernetes harus dimodifikasi menjadi permintaan proxy melalui host bastion. Ini dapat dilakukan dengan menggunakan perintah berikut:

```
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
```

Jika bekerja dengan benar, Anda akan melihat output seperti `'Property "clusters.arn:aws:eks:us-west-2:012345678912:cluster/WickrEnterprise5B8BF472-1234a41c4ec48b7b615c6789d93dcce.proxy-url" set.'`

Port maju ke benteng

Untuk terhubung ke kluster Amazon EKS, Anda harus memulai sesi SSM untuk mem-port permintaan penerusan ke proxy yang berjalan di host bastion Anda. Perintah untuk melakukan ini disediakan sebagai output `BastionSSMProxyEKSCommand` pada `WickrEks` tumpukan. Jalankan perintah berikut untuk melihat nilai output:

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

Perintah yang dikeluarkannya akan dimulai dengan`aws ssm start-session`. Jalankan perintah ini untuk memulai proxy lokal yang berjalan pada port 8888 di mana Anda dapat terhubung ke cluster Amazon EKS. Jika port forward bekerja dengan benar, output harus mengatakan 'Menunggu koneksi... '. Jaga agar proses ini berjalan sepanjang waktu yang Anda butuhkan untuk mengakses kluster Amazon EKS.

Jika semuanya diatur dengan benar, Anda akan dapat menjalankan `kubectl get nodes` di terminal lain untuk mencantumkan node pekerja di klaster 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
```

# Menginstal Wickr Enterprise

Setelah koneksi Anda ke cluster Kubernetes telah dibuat, Anda dapat mulai menginstal Wickr Enterprise menggunakan plugin. `kubectl kots` Anda akan memerlukan file Lisensi KOTS Anda (`.yaml`file yang disediakan oleh Wickr) dan file Config Values Anda, yang disimpan ke file di bagian Generate KOTS Config. `wickr-config.json` Untuk informasi selengkapnya tentang Menghasilkan Konfigurasi KOTS, lihat Menghasilkan Konfigurasi KOTS.

## Menginstal Wickr Enterprise secara manual

Perintah berikut akan memulai instalasi Wickr Enterprise:

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

Anda akan diminta untuk memasukkan kata sandi untuk Konsol Admin KOTS. Simpan kata sandi ini karena Anda akan memerlukannya untuk meningkatkan atau mengubah konfigurasi instalasi Wickr Enterprise Anda di masa depan.

Ketika instalasi selesai, `kubectl kots` akan membuka port lokal (biasanya`http://localhost:8080`), yang menyediakan akses ke Konsol Admin KOTS. Anda dapat mengubah atau memantau status instalasi Wickr Enterprise Anda di situs ini, atau mulai menyiapkan Wickr dengan mengunjungi nama domain yang Anda konfigurasi untuk instalasi Anda di browser Anda.

## Instalasi Wickr Enterprise dengan Lambda

Selama penyebaran CDK, Lambda dibuat dan dipanggil untuk menyelesaikan instalasi Wickr Enterprise atas nama Anda secara otomatis. Untuk memanggilnya secara manual, buka AWS konsol dan temukan fungsi `WickrLambda-func*` lambda, di bawah tab uji, pilih`test`, input tidak relevan.

# Instalasi pasca

Ada dua konsol web yang tersedia untuk mengelola instalasi Wickr Enterprise Anda: Konsol Admin KOTS dan Konsol Admin Wickr.

> **Note**
>
> Buat perubahan apa pun yang diperlukan untuk mencerminkan kebijakan pencadangan dan pencatatan organisasi Anda (pengaturan Amazon S3, log akses Elastic Load Balancing Amazon Virtual Private Cloud , log aliran).

## Konsol Admin KOTS

Antarmuka ini digunakan untuk mengelola versi Wickr Enterprise yang digunakan. Anda dapat melihat status instalasi, memodifikasi konfigurasi, atau melakukan upgrade. Konsol Admin KOTS hanya dapat diakses melalui port forward Kubernetes, yang dapat dibuka menggunakan perintah berikut:

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

> **Note**
>
> Anda harus terlebih dahulu mengatur koneksi bastion Anda seperti yang dijelaskan di port maju ke bagian bastion. Untuk informasi lebih lanjut tentang port forward ke benteng, lihat Memproksi koneksi melalui benteng.

Ketika port forward berhasil dikonfigurasi, perintah sebelumnya akan menampilkan yang berikut:

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

Gunakan URL yang disediakan untuk mengakses Konsol Admin KOTS. Kata sandi untuk masuk adalah yang Anda pilih saat menjalankan `kubectl kots install` selama instalasi. Jika Anda perlu mengatur ulang kata sandi, lihat Mengatur Ulang Kata Sandi Konsol Admin KOTS.

# Konsol Admin Wickr

Antarmuka ini digunakan untuk mengkonfigurasi instalasi Wickr Enterprise Anda untuk mengatur jaringan, pengguna, dan federasi. Ini dapat diakses melalui HTTPS pada nama DNS yang Anda konfigurasikan untuk menunjuk ke Load Balancer Anda. Jika DNS dikonfigurasi secara otomatis dengan zona yang dihosting publik, nama domain adalah nilai dari nilai `wickr/domainName` konteks.

Nama pengguna default adalah`admin`, dengan kata sandi`Password123`. Anda akan diminta untuk mengubah kata sandi ini saat masuk pertama.

# Nilai konteks

Nilai konteks adalah pasangan nilai kunci yang dapat dikaitkan dengan aplikasi, tumpukan, atau konstruksi. Mereka dapat diberikan ke aplikasi Anda dari file (biasanya baik `cdk.json` atau `cdk.context.json` di direktori proyek Anda) atau pada baris perintah. CDK menggunakan nilai konteks untuk mengontrol konfigurasi aplikasi. Wickr Enterprise menggunakan nilai konteks CDK untuk memberikan kontrol atas pengaturan seperti nama domain instalasi Wickr Enterprise Anda atau jumlah hari untuk menyimpan cadangan RDS.

Ada beberapa cara untuk mengatur nilai konteks, tetapi kami sarankan untuk mengedit nilai `cdk.context.json` agar sesuai dengan kasus penggunaan khusus Anda. Hanya nilai konteks yang dimulai dengan `wickr/` yang terkait dengan penyebaran Wickr Enterprise.

| Nama | Deskripsi | Default |
| --- | --- | --- |
| `wickr/licensePath` | Jalur ke lisensi KOTS Anda (`.yaml`file yang disediakan oleh Wickr). | null |
| `wickr/domainName` | Nama domain yang akan digunakan untuk penyebara n Wickr Enterprise ini. Jika menggunakan zona host publik Route 53, catatan DNS dan sertifikat ACM untuk nama domain ini akan dibuat secara otomatis. | null |
| `wickr/route53:host edZoneId` | Route 53 ID zona yang dihosting untuk membuat catatan DNS. | null |
| `wickr/route53:host edZoneName` | Route 53 Nama zona yang dihosting untuk membuat catatan DNS. | null |
| `wickr/acm:certific ateArn` | ARN dari sertifikat ACM untuk digunakan pada Load | null |

| Nama | Deskripsi | Default |
| --- | --- | --- |
| | Balancer. Nilai ini harus diberikan jika zona yang dihosting publik Route 53 tidak tersedia di akun Anda. | |
| `wickr/caPath` | Jalur sertifikat, hanya diperlukan saat menggunakan sertifikat yang ditandatangani sendiri. | null |
| `wickr/vpc:id` | ID VPC untuk menyebarkan sumber daya ke dalam. Hanya diperlukan saat menerapkan ke VPC yang ada. Jika tidak disetel, VPC baru akan dibuat. | null |
| `wickr/vpc:cidr` | IPv4 CIDR untuk mengasosi asikan dengan VPC yang dibuat. Jika menerapkan ke VPC yang ada, atur ini ke CIDR dari VPC yang ada. | 172.16.0.0/16 |
| `wickr/vpc:availabi lityZones` | Daftar zona ketersediaan yang dipisahkan koma. Hanya diperlukan saat menerapkan ke VPC yang ada. | null |
| `wickr/vpc:publicSu bnetIds` | Daftar subnet publik yang dipisahkan koma. IDs Hanya diperlukan saat menerapkan ke VPC yang ada. | null |
| `wickr/vpc:privateS ubnetIds` | Daftar subnet pribadi yang dipisahkan koma. IDs Hanya diperlukan saat menerapkan ke VPC yang ada. | null |

| Nama | Deskripsi | Default |
| --- | --- | --- |
| `wickr/vpc:isolatedSubnetIds` | Daftar subnet terisolasi yang dipisahkan koma IDs untuk database RDS. Hanya diperlukan saat menerapkan ke VPC yang ada. | null |
| `wickr/rds:deletionProtection` | Aktifkan perlindungan penghapusan pada instans RDS. | true |
| `wickr/rds:removalPolicy` | Kebijakan penghapusan untuk instance RDS 'snapshot', 'destroy', atau 'pertahankan.' | snapshot |
| `wickr/rds:readerCount` | Jumlah instance pembaca yang akan dibuat di cluster RDS. | 1 |
| `wickr/rds:instanceType` | Jenis instans yang akan digunakan untuk instance RDS. | r6g.xlarge |
| `wickr/rds:backupRetentionDays` | Jumlah hari untuk menyimpan cadangan. | 7 |
| `wickr/eks:namespace` | Namespace default untuk layanan Wickr di EKS. | anyaman |
| `wickr/eks:defaultCapacity` | Jumlah node pekerja EKS untuk infrastruktur Messaging. | 3 |
| `wickr/eks:defaultCapacityCalling` | Jumlah node pekerja EKS untuk infrastruktur Panggilan. | 2 |

| Nama | Deskripsi | Default |
| --- | --- | --- |
| wickr/eks:instanceTypes | Daftar tipe instans yang dipisahkan koma yang akan digunakan untuk node pekerja Messaging EKS. | m5.xlarge |
| wickr/eks:instanceTypesCalling | Daftar tipe instance yang dipisahkan koma untuk digunakan untuk Memanggil node pekerja EKS. | c5n.large |
| wickr/eks:enableAutoscaler | Beralih mengaktifkan fungsionalitas Cluster Autoscaler untuk EKS. | true |
| wickr/s3:expireAfterDays | Jumlah hari setelah unggahan file akan dihapus dari bucket S3. | 1095 |
| wickr/eks:clusterVersion | Versi cluster, termasuk versi Kubernetes, versi KubeCTLlayer, versi AlbController, versi, dan lainnya. nodeGroup Release | 1.27 |
| wickr/stackSuffix | Sufiks untuk diterapkan pada nama CloudFormation tumpukan. | " |
| wickr/autoDeployWickr | Terapkan aplikasi Wickr secara otomatis dengan lambda. | true |

# Menghancurkan sumber daya

Untuk menghapus semua yang dibuat oleh AWS CDK aplikasi ini, Anda harus menghapus `WickrRds` tumpukan sebelum semua tumpukan lainnya.

Agar sumber daya Amazon RDS dapat dihapus dengan benar, perlindungan penghapusan harus dinonaktifkan, dan kebijakan penghapusan harus disetel ke salah satu atau. `snapshot` `destroy` Jika ini bukan pengaturan saat ini, ubah `wickr/rds:deletionProtection` dan `wickr/rds:removalPolicy` nilai dalam AWS CDK konteks Anda dan terapkan kembali tumpukan Amazon RDS dengan menjalankannya. `npx cdk deploy -e WickrRds`

Setelah kebijakan perlindungan penghapusan dan penghapusan disetel dengan benar, jalankan `cdk destroy` untuk tumpukan: `WickrRds`

```
npx cdk destroy WickrRds
```

Ketika `WickrRds` tumpukan telah selesai dihancurkan, CloudFormation tumpukan yang tersisa dapat dihancurkan dengan perintah berikut:

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

# Pemecahan Masalah

## Menghapus namespace Wickr

Jika Anda perlu menghapus `wickr` namespace untuk memulai kembali, penting bagi Anda untuk membuat cadangan terlebih dahulu Akun Layanan apa pun yang dibuat oleh CDK di dalam namespace tersebut. Akun Layanan ini memungkinkan layanan Wickr untuk berkomunikasi AWS APIs melalui peran IAM. Tanpa mereka, tugas seperti unggahan file melalui Amazon Simple Storage Service (Amazon S3) tidak akan berfungsi lagi.

Gunakan perintah berikut untuk mencadangkan Akun Layanan dan menghapus dan membuat ulang `wickr` namespace dan Akun Layanan yang sesuai:

```
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
```

## Menyetel ulang kata sandi Konsol Admin KOTS

Anda dapat mengatur ulang kata sandi Konsol Admin KOTS Anda dengan perintah berikut:

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

Ketika Anda mengubah kata sandi ini, Anda mungkin juga ingin memperbarui `wickr/kots` rahasia Secrets Manager juga, meskipun umumnya tidak akan digunakan lagi oleh otomatisasi apa pun.

## Masalah menghubungkan ke cluster EKS dengan benteng

Jika koneksi Anda ke cluster EKS melalui bastion tampak lambat atau kadang-kadang habis waktu, Anda mungkin melihat kesalahan berikut saat menjalankan `kubectl` perintah:

net/http: permintaan dibatalkan sambil menunggu koneksi (Client.Timeout terlampaui saat menunggu header)

Masalah ini seringkali dapat diatasi dengan masuk ke host bastion melalui SSM (lihat `BastionSSMCommand` di WickrEks tumpukan) dan memulai ulang layanan: `tinyproxy`

```
sudo systemctl restart tinyproxy
```

# Instalasi kustom

Di bagian instalasi Kustom, Anda akan belajar cara menginstal Wickr Enterprise.

Topik


- Persyaratan
- Arsitektur
- Penginstalan
- Pengaturan masuk
- Pengaturan basis data
- Penyimpanan file S3
- Pengaturan klaim volume persisten
- Pengaturan sertifikat TLS
- Pengaturan panggilan
- Memanggil pengaturan ingress
- Autoscaler cluster Kubernetes (opsional)
- Pencadangan
- Instalasi airgap
- Konsol admin Wickr
- Pengaturan keamanan
- Pertanyaan yang Sering Diajukan


# Persyaratan

Sebelum Anda mulai menginstal Wickr Enterprise, verifikasi bahwa persyaratan berikut terpenuhi.

## Persyaratan Perangkat Keras

Wickr Enterprise membutuhkan klaster Kubernetes untuk beroperasi. Dimungkinkan untuk beroperasi pada satu node dengan Mode Sumber Daya Rendah diaktifkan, tetapi ini tidak disarankan untuk penggunaan produksi umum. Dalam penerapan Produksi, kami merekomendasikan minimal tiga node pekerja pesan serta minimal dua node pekerja panggilan.

Node pekerja harus memiliki spesifikasi minimum berikut.

- 2 hingga 4 core CPU
- 8 GB Ram
- 200 GB ruang disk

Persyaratan Perangkat Keras Minimum

Cluster node pekerja tunggal yang berjalan dalam Mode Sumber Daya Rendah membutuhkan minimal 3000m CPU dan 5846Mi Ram. Ini tidak termasuk pod sistem kube.

Persyaratan Sumber Daya Berdasarkan Pod

| Nama Pod | Pemilik | CPU | Memori |
|---|---|---|---|
| admin-api | Anyaman | 100m | 256Mi |
| direktori | Anyaman | 100m | 128Mi |
| kadaluarsa | Anyaman | 100m | 128Mi |
| proksi file | Anyaman | 100m | 256Mi |
| oidc | Anyaman | 100m | 128Mi |
| pencarian terbuka | Anyaman | 500m | 100Mi |
| orville | Anyaman | 50m | 128Mi |
| orville-redis | Anyaman | 50m | 128Mi |
| perangkat dorong | Anyaman | 100m | 128Mi |
| rabbitmq | Anyaman | 50m | 256Mi |
| bereaksi | Anyaman | 100m | 64Mi |
| tanda terima | Anyaman | 250m | 128Mi |
| redis | Anyaman | 50m | 128Mi |
| server-api | Anyaman | 250m | 256Mi |

| Nama Pod | Pemilik | CPU | Memori |
|---|---|---|---|
| switchboard | Anyaman | 250m | 512Mi |
| kotsadm | KOTS | 50m | 50Mi |
| kotsadm-minio | KOTS | 100m | 512Mi |
| kotsadm-rqlite | KOTS | 200m | 1Gi |
| minio-operator | Internal S3 | 200m | 256Mi |
| penyewa mini | Internal S3 | 100m | 256Mi |
| mysql-primer | MySQL Internal | 100m | 512Mi |
| mysql-sekunder | MySQL Internal | 100m | 512Mi |

Persyaratan Penyimpanan

Wickr Enterprise memerlukan default StorageClass untuk digunakan saat membuat Klaim Volume Persisten. Saat menerapkan di lingkungan yang memiliki celah udara atau di tempat, Anda mungkin perlu mengonfigurasinya untuk klaster Anda. Salah satu opsi yang tersedia adalah Longhorn. Persyaratan ruang disk yang disarankan akan bervariasi berdasarkan penggunaan opsi Internal S3 dan opsi Internal Mysql dan jumlah ruang yang ingin Anda miliki untuk unggahan file.

- Caching Gambar Internal: ~ 60 Gi
- RabbitMQ: 24 Gi Default/8 Gi dalam Mode Sumber Daya Rendah
- Redis: 24 Gi Default/8 Gi dalam Mode Sumber Daya Rendah
- OpenSearch: 24 Gi Default/8 Gi dalam Mode Sumber Daya Rendah
- Mysql Internal: 80 Gi Default/20Gi dalam Mode Sumber Daya Rendah
- Internal S3:160 Gi Default/2Gi dalam Mode Sumber Daya Rendah
- KOTS Minio: 4 Gi
- KOTS Rqlite: 1 Gi

Ukuran Penyimpanan Minimum

- 377 Gi Default dengan Internal S3 dan Internal Mysql
- 111 Gi dalam Mode Sumber Daya Rendah

Persyaratan Versi Kubernetes

Wickr Enterprise mengandalkan REPLICATED KOTS. Replicated, platform distribusi perangkat lunak komersial, menyediakan daftar versi Kubernetes yang saat ini didukung. Untuk informasi selengkapnya, lihat Kompatibilitas Versi Kubernetes.

## Persyaratan perangkat lunak

Wickr Enterprise membutuhkan klaster Kubernetes dan KOTS untuk beroperasi. Silakan merujuk ke dokumentasi KOTS untuk versi OS dan Kubernetes yang didukung. Untuk informasi selengkapnya, lihat Persyaratan Sistem Minimum.

Sistem Host Pengembang

Sistem Operasi — Perintah dalam dokumentasi ini dirancang untuk bekerja di Linux, macOS, atau Windows dengan WSL (Windows Subsystem for Linux) diinstal.

Layanan Stateful Internal

Wickr Enterprise dapat menyediakan layanan internal untuk database MySQL dan penyimpanan yang kompatibel dengan S3 namun untuk penggunaan produksi umum disarankan Anda menyediakan layanan ini eksternal dari cluster Kubernetes.

- MySQL 5.7 Database
  - Amazon RDS MySQL 5.7 atau MySQL 5.7 database (Eksternal)
  - Bagan Helm Mysql Bitnami (Internal)
  - Penyimpanan File
    - Penyedia penyimpanan yang kompatibel dengan Amazon S3 atau S3 (Eksternal)
    - Bagan Helm Operator Minio (Internal)

## Persyaratan jaringan

Wickr Enterprise membutuhkan FQDN, sertifikat SSL, dan port TCP dan UDP terbuka tertentu.

- FQDN: Domain atau sub-domain yang akan digunakan oleh penyebaran Wickr Enterprise.

- Sertifikat SSL: Sebuah key pair sertifikat SSL yang ditandatangani oleh CA publik atau self signed certificate key pair. Sertifikat harus mencantumkan FQDN dalam Nama Umum dan juga sebagai entri SAN DNS. Sertifikat juga harus mengaktifkan ekstensi ServerAuth extendedKeyUsage .
- Penginstalan online akan membutuhkan akses keluar ke sumber daya yang direplikasi dan pihak ketiga. Replicated memelihara daftar alamat IP mereka. Untuk informasi selengkapnya, lihat Alamat IP yang direplikasi. Replicated juga menyimpan daftar sumber daya pihak ketiga yang dibutuhkan. Untuk informasi selengkapnya, lihat Bukaan Firewall untuk Instalasi Online.
- Instalasi dengan celah udara memerlukan akses ke registri kontainer pribadi.

### Node Pesan

Node pesan tidak memerlukan IPV4 alamat publik dan harus ditempatkan di subnet pribadi. Lalu lintas pesan akan memasuki cluster melalui LoadBalancer atau Ingress.

### Memanggil Node

Memanggil node memerlukan IPV4 alamat publik sehingga mereka harus berada di subnet publik. Media panggilan ditransfer melalui UDP secara default. Ketika panggilan TCP diaktifkan, Proxy TCP akan menerima koneksi pada TCP 443 dan akan mem-proxy mereka ke layanan Orville.

- TCP: 443 Memanggil Proxy TCP
- UDP: 16384-16484 Aliran Audio/Video

### Akses Instalasi dan Konfigurasi

Akses ke Konsol Admin KOTS untuk instalasi dan konfigurasi dilakukan melalui port Kubernetes forward.

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

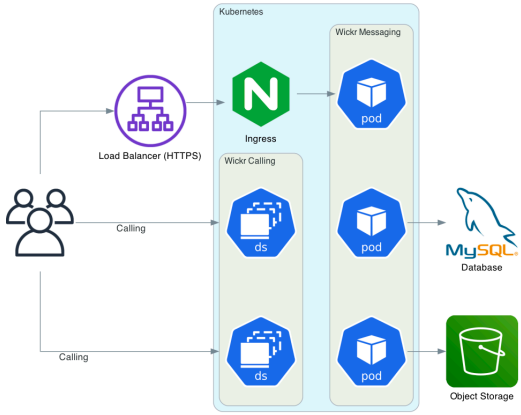
### Persyaratan Lisensi

Instalasi akan memerlukan file lisensi.yaml format.yaml, ini akan diberikan kepada Anda oleh Wickr Support.

## Arsitektur

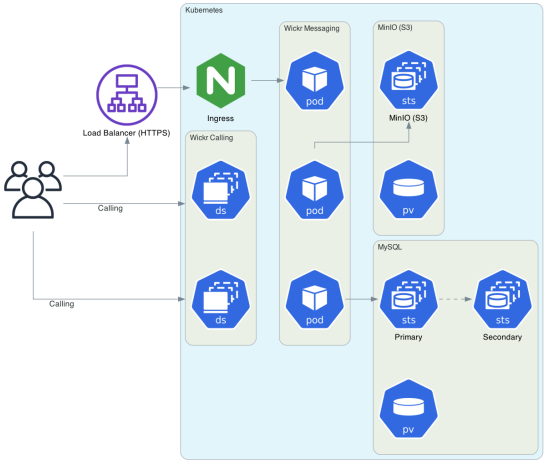
### Arsitektur Produksi yang Direkomendasikan

Diagram di bawah ini menunjukkan Wickr Enterprise dikonfigurasi seperti yang direkomendasikan untuk produksi, dengan layanan MySQL dan Object Storage terletak di luar klaster Kubernetes.




## Arsitektur Internal atau Uji

Diagram di bawah ini menampilkan konfigurasi Wickr Enterprise, memanfaatkan layanan MYSQL dan Object Storage internal. Meskipun dapat memenuhi kebutuhan spesifik penerapan tertentu, tidak disarankan untuk penggunaan produksi umum.



# Penginstalan

1. Instal kubectl dan kots CLI.
2. Connect ke cluster Kubernetes.
3. Dapatkan file lisensi Wickr Enterprise dari Wickr Support.
4. Instal Wickr Enterprise menggunakan perintah berikut.

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

> **Note**
>
> license.yaml mewakili file lisensi yang Anda berikan.

Setelah instalasi awal, Konsol Admin KOTS akan menyediakan manajemen tingkat cluster dan opsi konfigurasi.

## Konsol Admin KOTS

Antarmuka ini digunakan untuk mengelola versi Wickr Enterprise yang digunakan. Anda dapat melihat status instalasi, memodifikasi konfigurasi, atau melakukan upgrade Wickr Enterprise. Konsol Admin KOTS hanya dapat diakses melalui port forward Kubernetes, yang dapat dibuka menggunakan perintah berikut:

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

# Pengaturan masuk

Pengontrol Ingress

Wickr Enterprise mendukung empat jenis pengontrol ingress:

- LoadBalancer (Default)
  - Objek loadbalancer mungkin memerlukan konfigurasi eksplisit dalam instalasi on-prem sepenuhnya, meskipun sering disediakan oleh penyedia cloud.

  - Menyebarkan layanan ingress controller (ingress-nginx) dengan tipe layanan. LoadBalancer Ini mengharuskan klaster Kubernetes berjalan pada platform yang mendukung penyeimbang beban eksternal.
- ALB yang ada
  - Memasang pengontrol ingress ke ALB yang ada.
  - Anda harus menyediakan ARN Grup Target Application Load Balancer yang ada.
- NLB yang ada
  - Memasang pengontrol ingress ke NLB yang ada.
  - Anda harus menyediakan ARN Grup Target Network Load Balancer yang ada.
- NodePort
  - Pengontrol ingress (ingress-nginx) akan dikonfigurasi untuk menggunakan tipe NodePort layanan, yang membuka port pada semua node di cluster Kubernetes dan meneruskan lalu lintas ke ingress. Lalu lintas klien kemudian dapat diarahkan ke node ini baik melalui DNS atau beberapa penyeimbang beban eksternal.
  - Anda dapat memilih rentang port dari 1-65535, atau port acak dari 30000-32767 akan digunakan.
- Ingress
  - Bawa pengontrol masuk Anda sendiri. Konfigurasi ini akan menerima nama kelas ingress yang kemudian akan digunakan layanan dalam manifes Ingress mereka. Ini menyiratkan bahwa pengontrol ingress memiliki beberapa konektivitas eksternal yang sudah dikonfigurasi melalui beberapa mekanisme penyeimbangan beban lainnya.
  - Saat ini hanya pengontrol ingress-nginx yang didukung.

Nama Host Wildcard

Secara default, rute Ingress akan ditentukan dengan nilai host `*`. Nonaktifkan pengaturan ini untuk menggunakan nama host yang ditentukan untuk Wickr Enterprise Server. Wildcard Hostname diperlukan untuk nama host berbasis IP.

# Pengaturan basis data

Wickr Enterprise membutuhkan database MySQL 8.0. Jika Anda menggunakan MySQL 5.7, lihat untuk meng-upgrade. Tingkatkan ke MySQL 8.0 Sebaiknya gunakan database yang berada di luar

klaster Kubernetes Anda, seperti Amazon RDS, tetapi Anda juga memiliki opsi untuk menerapkan database MySQL Internal di dalam klaster Kubernetes sebagai bagian dari instalasi.

## Pengaturan Database Eksternal

- Nama host: Nama host atau alamat IP dari server database.
- Nama Host Pembaca: Nama host atau alamat IP dari endpoint read-only untuk server database (jika tersedia).
- Port: Port tempat MySQL akan diakses.
- Nama Database: Nama database yang dibuat di server.
- Username: Pengguna yang memiliki izin untuk mengakses database.
- Kata sandi: Kata sandi untuk pengguna itu.
- Sertifikat CA: Sertifikat PEM untuk menghubungkan ke database melalui TLS.

> **Note**
>
> Pastikan instalasi MySQL Anda menggunakan set karakter latin1 default dengan kolasi latin1_swedish_ci. Ini dapat dilakukan dengan memverifikasi bahwa server MySQL Anda dimulai dengan flag berikut:
> `"--character-set-server latin1", "--collation-server latin1_swedish_ci"`

## Pengaturan Database Internal

Jenis database internal akan menyebarkan dua StatefulSets ke dalam cluster Anda untuk MySQL primer dan sekunder dengan replikasi biner. Sekunder tidak menerima lalu lintas apa pun dan hanya tersedia untuk pemulihan dan cadangan bencana.

Storage Size: Size (dalam gibibytes) dari Persistent Volume untuk pod database.

Meningkatkan Ukuran Penyimpanan MySQL

> **Note**
>
> Jenis volume Anda StorageClass harus mendukung ekspansi volume untuk meningkatkan ukuran penyimpanan. Untuk informasi selengkapnya, lihat Ekspansi volume.

Layanan MySQL yang digunakan di Wickr Enterprise digunakan sebagai sumber daya di Kubernetes. StatefulSet StatefulSets membuat banyak properti sumber daya tidak dapat diubah, termasuk templat Klaim Volume Persisten. Sebagai solusi untuk kekekalan StatefulSets, tindakan berikut harus dilakukan untuk meningkatkan ukuran volume yang digunakan oleh MySQL.

1. Edit Klaim Volume Persisten untuk `data-mysql-primary-0` dan`data-mysql-secondary-0`.

    1. `kubectl -n wickr edit pvc data-mysql-primary-0`. Set `spec.resources.requests.storage`ke ukuran penyimpanan yang diinginkan.
    2. `kubectl -n wickr edit pvc data-mysql-secondary-0`. Set `spec.resources.requests.storage`ke ukuran penyimpanan yang diinginkan.
2. Hapus yang sudah ada StatefulSets, tetapi tinggalkan Pod dengan meneruskan `--cascade=orphan` flag.

    `kubectl -n wickr delete statefulset --cascade=orphan mysql-primary mysql-secondary`.
3. Di UI KOTS, perbarui pengaturan Ukuran Penyimpanan agar sesuai dengan nilai yang Anda tetapkan di Langkah 1. Simpan dan terapkan konfigurasi ini.
4. Mulai ulang StatefulSets untuk memperluas volume dan membawa layanan MySQL kembali online.

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

## Tingkatkan ke MySQL 8.0

Database Eksternal (RDS)

Untuk membuat Wickr Backend offline, selesaikan langkah-langkah berikut.

1. Temukan namespace dari ingress `kubectl get deployments --all-namespaces`

Dalam contoh di bawah ini, namespace adalah Wickr dan replika adalah 3.

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

2. Turunkan masuknya `kubectl scale deployment/ingress-nginx-controller --replicas=0 -n wickr`
3. Ambil snapshot untuk backup DB. Untuk informasi selengkapnya, lihat Mengelola backup manual di Panduan Pengguna Amazon Relational Database Service.
4. Tingkatkan versi mesin ke MySQL 8.0.x (MySQL 8.4 tidak didukung). Untuk informasi selengkapnya, lihat Memutakhirkan versi mesin instans DB di Panduan Pengguna Amazon Relational Database Service.

Untuk menghadirkan Wickr Backend online, tingkatkan kembali ingress `kubectl scale deployment/ingress-nginx-controller --replicas=3 -n wickr`

Database Internal

Untuk informasi selengkapnya, lihat Backup dan Restore MySQL.

# Penyimpanan file S3

Wickr Enterprise membutuhkan layanan penyimpanan yang kompatibel dengan S3. Sebaiknya gunakan layanan S3 yang berada di luar klaster Kubernetes Anda, seperti Amazon S3, tetapi Anda juga memiliki opsi untuk menerapkan layanan Internal S3 di dalam klaster Kubernetes sebagai bagian dari instalasi.

Pengaturan S3 Eksternal

- Nama Bucket: Nama bucket S3 tempat unggahan file akan disimpan.
- Wilayah: AWS Wilayah ember S3.
- Endpoint: Tetapkan titik akhir yang akan digunakan Wickr untuk berinteraksi dengan S3 API. Default ke titik akhir layanan S3 wilayah.
- Nama Akun Layanan Fileproxy: Hanya Amazon S3. Nama Akun Layanan Kubernetes yang ada untuk digunakan untuk mengautentikasi ke S3 menggunakan peran IAM untuk Akun Layanan.

- Kunci Akses S3 Eksternal: Ini adalah kunci Akses S3 Anda yang ada.
- Kunci Rahasia S3 Eksternal: Ini adalah kunci Rahasia S3 Anda yang ada.

Pengaturan S3 Internal

Tipe S3 internal akan menerapkan default dari 4 pod server MiniO yang masing-masing berisi 4 Klaim Volume Persisten. Konfigurasi default menggunakan Erasure Coding MiniO untuk meningkatkan toleransi kesalahan.

- Jumlah server S3 internal: Jumlah pod server MiniO yang akan dibuat, defaultnya adalah 4 untuk penerapan toleran kesalahan. Nilai ini dapat diatur serendah 1 untuk development/test penerapan.
- Jumlah volume S3 internal: Jumlah volume MiniO yang akan dibuat di setiap pod server MiniO, defaultnya adalah 4 untuk penerapan toleran kesalahan. Nilai ini dapat diatur serendah 1 untuk development/test penerapan.
- Ukuran volume S3 internal: Ukuran dalam GB volume MiniO yang dibuat di pod server MiniO, defaultnya adalah 10GB.
- Penyebaran Internal S3 default akan menggunakan 4 server dengan 4. PVCs Setiap PVC adalah 10 Gi menghasilkan 160 Gi Raw penyimpanan dengan 120 Gi Erasure Coded penyimpanan tersedia untuk pengguna.
- Kalkulator Minio Erasure Coding tersedia. Untuk informasi selengkapnya, lihat Kalkulator Kode Penghapusan.

# Pengaturan klaim volume persisten

Wickr Enterprise membutuhkan Klaim Volume Persisten untuk menyimpan data stateful. Pengaturan ini memungkinkan Anda untuk menentukan nama nama Storage Class yang ingin Anda gunakan. Jika dibiarkan kosong Wickr akan mencoba untuk menggunakan default Storage Class. Mengubah Storage Class setelah Wickr telah digunakan tidak didukung.

Default StorageClass untuk Klaim Volume Persisten sering disediakan oleh penyedia cloud, namun dalam instalasi onprem sepenuhnya mungkin memerlukan konfigurasi eksplisit menggunakan layanan pihak ketiga seperti Longhorn.

# Pengaturan sertifikat TLS

Unggah sertifikat PEM dan kunci pribadi untuk mengakhiri TLS. Nama Alternatif Subjek pada sertifikat harus sesuai dengan nama host yang dikonfigurasi dalam pengaturan penerapan Wickr Enterprise Anda.

Untuk bidang rantai sertifikat, gabungkan sertifikat perantara (jika diperlukan) dengan sertifikat CA root sebelum mengunggah.

## Mari Enkripsi

Pilih opsi ini untuk secara otomatis menghasilkan sertifikat menggunakan Let's Encrypt. Sertifikat dikeluarkan menggunakan tantangan HTTP-01 melalui operator cert-manager.

Tantangan HTTP-01 mengharuskan nama DNS yang diinginkan diselesaikan ke titik masuk untuk cluster Anda (biasanya Load Balancer), dan lalu lintas ke port TCP 80 terbuka untuk umum. Sertifikat ini berumur pendek dan akan diperbarui secara teratur. Port 80 harus tetap terbuka untuk memungkinkan sertifikat diperbarui secara otomatis.

> **Note**
>
> Bagian ini merujuk secara eksplisit ke sertifikat yang digunakan oleh aplikasi Wickr Enterprise itu sendiri.

## Sertifikat yang disematkan

Wickr Enterprise memerlukan penyematan sertifikat saat menggunakan sertifikat yang ditandatangani sendiri atau sertifikat yang tidak dipercaya oleh perangkat klien. Jika sertifikat yang diberikan oleh Load Balancer Anda ditandatangani sendiri atau ditandatangani oleh CA yang berbeda dari instalasi Wickr Enterprise, unggah sertifikat CA di sini agar klien menyematkannya.

Dalam kebanyakan situasi, pengaturan ini tidak diperlukan.

## Penyedia Sertifikat

Jika Anda berencana untuk membeli sertifikat untuk digunakan dengan Wickr Enterprise lihat di bawah untuk daftar penyedia yang sertifikat diketahui berfungsi dengan benar secara default. Jika

penyedia tercantum di bawah sertifikat mereka telah divalidasi dengan perangkat lunak secara eksplisit.

- Digicert
- RapidSSL

## Menghasilkan sertifikat yang ditandatangani sendiri

Jika Anda ingin membuat sertifikat yang ditandatangani sendiri untuk digunakan dengan Wickr Enterprise, perintah contoh di bawah ini berisi semua flag yang diperlukan untuk pembuatan.

```
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"
```

Jika Anda ingin membuat sertifikat yang ditandatangani sendiri berbasis IP, gunakan perintah berikut sebagai gantinya. Untuk menggunakan sertifikat berbasis IP, pastikan bahwa bidang Wildcard Hostname diaktifkan di bawah pengaturan Ingress. Untuk informasi selengkapnya, lihat Pengaturan masuk.

```
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**
>
> Ganti $YOUR_DOMAIN dalam contoh dengan nama domain atau alamat IP yang ingin Anda gunakan.

# Pengaturan panggilan

- Required Calling Nodes: Ketika pengaturan ini diaktifkan, layanan panggilan Wickr hanya akan di-deploy pada Kubernetes Nodes dengan label. `role=calling` Nonaktifkan pengaturan ini untuk menerapkan layanan Panggilan dan Pesan pada node yang sama, atau untuk penerapan node tunggal.

Anda biasanya juga ingin menonaktifkan panggilan TCP Proxy ketika pengaturan ini dinonaktifkan, karena layanan TCP Proxy berjalan pada port 443.

- Aktifkan Proxy TCP: Pengaturan ini mengontrol apakah layanan untuk mode fallback TCP pada panggilan diterapkan atau tidak. Nonaktifkan pengaturan ini jika Anda memiliki layanan lain yang berjalan pada 443/tcp atau tidak memerlukan mode fallback TCP untuk panggilan. Ini harus diaktifkan agar penerapan berencana menggunakan Wickr Open Access.
- Secara otomatis menemukan alamat IP publik server: Ketika pengaturan ini diaktifkan, layanan panggilan akan menemukan alamat IP publik mereka dengan membuat permintaan HTTPS ke https://ipv4.icanhazip.com/ danhttps://ipv6.icanhazip.com/. Saat dinonaktifkan, Anda harus mengaktifkan pengaturan “Gunakan alamat IP utama host untuk Memanggil lalu lintas” atau “penggantian nama host” jika tidak, layanan panggilan akan gagal dimulai.
- Gunakan alamat IP primer host untuk Memanggil lalu lintas: Gunakan alamat IP utama node Kubernetes untuk memanggil layanan. Ini menyiratkan bahwa semua klien Wickr dapat terhubung ke node Kubernetes Anda pada alamat IP utama node, seperti yang disajikan dari Downward API. `status.hostIP`
- Penggantian nama host: Berikan nama host atau alamat IP untuk kembali sebagai titik konektivitas untuk layanan Panggilan. Pengaturan ini hanya boleh digunakan saat menjalankan server panggilan tunggal, karena nilai yang sama dikembalikan untuk semua replika layanan. Ketika penggantian nama host disetel dan pengaturan “gunakan alamat IP primer host” diaktifkan, pengaturan alamat IP utama host diutamakan.
- Memanggil Host Network Enabled: Secara default, memanggil pod menggunakan jaringan host node untuk konektivitas. Nonaktifkan ini untuk mengekspos NodePort layanan untuk memanggil lalu lintas. Jika panggilan ingress diaktifkan, pastikan bahwa layanan yang sesuai dikonfigurasi untuk memungkinkan lalu lintas masuk. Ini harus dinonaktifkan untuk kepatuhan STIG.

## Memanggil pengaturan ingress

Wickr mendukung pengaturan ingress panggilan, memungkinkan klien untuk terhubung ke node panggilan apa pun di dalam cluster dan memiliki rute panggilan ke server panggilan yang benar. Wickr mendukung empat jenis ingress panggilan:

- LoadBalancer (default)
  - Ini LoadBalancer akan disediakan oleh penyedia cloud (instalasi sepenuhnya di lokasi akan memerlukan konfigurasi tambahan). Setelah disediakan, konfigurasi KOTS harus diperbarui lagi untuk memberikan nama host atau alamat IP penyeimbang beban. LoadBalancer

- NodePort
  - Mengekspos NodePort layanan pada setiap node panggilan yang akan berfungsi sebagai titik masuk untuk memanggil lalu lintas. Nama host yang menyelesaikan satu atau lebih node atau alamat IP dari satu atau lebih node harus disediakan. Anda dapat memilih rentang port dari 30000-32767 untuk UDP dan lalu lintas TCP opsional.
- NLB yang ada
  - Melampirkan layanan ingress panggilan ke NLB yang ada. Anda harus menyediakan ARN kelompok sasaran untuk UDP dan opsional, lalu lintas TCP.
- Tidak ada Layanan
  - Pilih ini jika Anda tidak memerlukan layanan Kubernetes tambahan untuk mengizinkan lalu lintas masuk. Ini biasanya akan digunakan dengan pengaturan jaringan host untuk merutekan lalu lintas masuk panggilan langsung ke node panggilan Anda.

## Pertimbangan-pertimbangan

- Untuk memastikan kompatibilitas mundur dengan klien lama dan jaringan federasi tanpa memanggil ingress, saat memanggil ingress diaktifkan, mode panggilan lama masih tersedia (koneksi langsung ke server panggilan). Jika mengubah port default apa pun, pastikan Anda tidak memiliki tabrakan port pada node panggilan.
- Dual-stack yang NLBs melayani lalu lintas UDP harus memiliki IPv6 target backend. Untuk informasi selengkapnya, lihat Grup Target Network Load Balancer.
- Jika Anda memerlukan kepatuhan STIG, Anda harus menonaktifkan opsi jaringan host untuk menelepon. Jika node dikonfigurasi dalam mode dual-stack, tetapi cluster tidak, Anda mungkin kehilangan IPv6 konektivitas (dengan asumsi cluster). IPv4
- Memanggil ingress memerlukan nama host atau Alamat IP yang telah ditentukan sebelumnya. Menskalakan node atau menyediakan perutean khusus mungkin memerlukan modifikasi konfigurasi.
- Port ingress panggilan default adalah 8443 untuk TCP dan 16384 untuk UDP. Pastikan firewall dan grup keamanan memungkinkan lalu lintas untuk port ini, atau port alternatif jika default diganti.

## Arsitektur referensi

Masuknya dengan penyeimbang beban

Opsi ini memperlihatkan penyeimbang beban tunggal sebagai titik masuk untuk semua lalu lintas panggilan.

1. Untuk Memanggil Jenis Ingress, pilih Load Balancer atau Existing NLB. Untuk informasi lebih lanjut tentang NLB yang Ada, rujuk tumpukan NLB di Sampel CDK Wickr Enterprise di. GitHub
2. Lakukan salah satu hal berikut, tergantung pada Jenis Panggilan Ingress:

   - Untuk NLB yang Ada, berikan grup target ARNs untuk lalu lintas UDP dan TCP dan nama host NLB.
   - Untuk Load Balancer, berikan nama host setelah disediakan oleh Kubernetes.

   Atau, untuk Calling Ingress Type, Anda dapat memberikan alamat IP penyeimbang beban atau nama host khusus yang menunjuk ke penyeimbang beban.
3. (Opsional) Untuk menggabungkan lalu lintas pesan dan panggilan di bawah satu NLB, pilih NLB yang ada di bagian Ingress, dan sediakan grup target HTTPS.

### Masuknya dengan NodePort

Opsi ini berguna jika jaringan host dinonaktifkan dan Anda tidak ingin mengekspos penyeimbang beban tambahan.

> **Note**
>
> Pastikan firewall dan grup keamanan Anda memungkinkan lalu lintas untuk. NodePorts

1. Untuk Memanggil Jenis Ingress, pilih NodePort.
2. Tambahkan nama host node panggilan atau alamat IP.
3. Nonaktifkan Jaringan Host Panggilan.

### Masuknya langsung dengan HostNetwork

Opsi ini tidak mengekspos layanan Kubernetes tambahan dan memungkinkan lalu lintas masuk panggilan untuk terhubung langsung melalui jaringan host node panggilan. Pendekatan ini lebih disukai jika IPv6 konektivitas diperlukan.

1. Untuk Memanggil Jenis Ingress, pilih No Service.

2. Tambahkan nama host node panggilan atau alamat IP.
3. Aktifkan Memanggil Jaringan Host.

# Autoscaler cluster Kubernetes (opsional)

Kubernetes Cluster Autoscaler adalah nilai konfigurasi opsional untuk instalasi Wickr Enterprise. Ini akan membantu dalam penskalaan grup node Kubernetes Anda jika terjadi peningkatan lalu lintas atau pembatasan sumber daya lainnya yang dapat menyebabkan kinerja buruk.

Instalasi Wickr Enterprise mendukung 3 integrasi penyedia Cloud: AWS, Google Cloud, dan Azure. Setiap penyedia Cloud memiliki persyaratan yang berbeda untuk integrasi ini. Ikuti petunjuk untuk penyedia cloud spesifik Anda di bawah ini untuk mengaktifkan fitur ini.

## AWS

Jika Anda tidak menggunakan WickrEnterprise CDK untuk menginstal Lingkungan Wickr Anda AWS, Anda perlu mengambil beberapa langkah tambahan untuk mengaktifkan Cluster Autoscaler.

1. Tambahkan tag berikut ke Grup Node Anda. Hal ini memungkinkan Cluster Autoscaler untuk secara otomatis menemukan node yang sesuai.
    1. `k8s.io/cluster-autoscaler/`**`clusterName`** `= owned`dimana clusterName adalah nama dari Cluster Kubernetes Anda
    2. `k8s.io/cluster-autoscaler-enabled = true`
2. Tambahkan Akun Layanan Kubernetes, di namespace sistem kube dan kaitkan dengan kebijakan IAM yang memungkinkan tindakan penskalaan otomatis dan ec2. Untuk informasi selengkapnya dan petunjuk terperinci, lihat Mengonfigurasi akun layanan Kubernetes untuk mengambil peran IAM dalam Panduan Pengguna Amazon EKS.
    1. Anda harus menggunakan namespace 'kube-system' saat menyiapkan Akun Layanan
    2. Kebijakan berikut dapat digunakan untuk Akun Layanan:

       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"
        }
    ]
}
```

Di UI Replikasi saat mengonfigurasi Cluster Autoscaler, pilih AWSsebagai penyedia cloud Anda dan berikan nama Akun Layanan yang Anda buat di atas untuk menginstruksikan Cluster Autoscaler agar menggunakan akun layanan tersebut.

## Awan Google

Sangat disarankan untuk menggunakan kemampuan Autoscaling bawaan dari GKE untuk Autopilot dan cluster standar. Namun, jika Anda ingin melanjutkan integrasi ini, persyaratan berikut harus dipenuhi sebelum melanjutkan.

Persyaratan:

1. Grup Instans Terkelola (MIG) harus dibuat dengan Security Scope termasuk minimal 'Baca/Tulis' ke Compute Engine Resources. Ini tidak dapat ditambahkan ke MIG nanti saat ini.
2. Cluster harus mengaktifkan Federasi Identitas Beban Kerja. Anda dapat mengaktifkan ini di klaster yang ada dengan menjalankan: `gcloud container clusters update ${CLUSTER_NAME} --workload-pool=${PROJECT_ID}.svc.id.goog`
3. Akun Layanan Google Cloud Platform (GCP) dengan akses ke peran `Roles/Compute.InstanceAdmin.v1`. Ini dapat dibuat menggunakan instruksi ini:

```
# 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]"
```

## Biru langit

Azure Kubernetes Service (AKS) menyediakan penskalaan otomatis klaster terintegrasi untuk sebagian besar penerapan dan sangat disarankan untuk menggunakan metode tersebut untuk penskalaan otomatis klaster. Namun, jika persyaratan Anda sedemikian rupa sehingga metode tersebut tidak berfungsi, kami telah menyediakan integrasi Autoscaler Kluster Kubernetes Kubernetes untuk Layanan Azure Kubernetes. Untuk memanfaatkan integrasi ini, Anda perlu mengumpulkan informasi berikut dan memasukkannya ke dalam konfigurasi panel admin KOTS di bawah Cluster Autoscaler setelah memilih Azure sebagai penyedia cloud Anda.

Otentikasi Azure

ID Langganan: ID berlangganan dapat diperoleh melalui portal Azure dengan mengikuti dokumentasi resmi. Untuk informasi selengkapnya, lihat Dapatkan langganan dan penyewa IDs di portal Azure.

Parameter berikut dapat diperoleh dengan membuat Principal Layanan AD menggunakan utilitas baris perintah az.

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

ID Aplikasi:

Kata Sandi Klien:

ID Penyewa:

Konfigurasi Autoscaler Azure Cluster

Selain persyaratan otentikasi, bidang-bidang berikut diperlukan untuk berfungsinya autoscaler cluster dengan benar. Perintah untuk mendapatkan informasi ini telah disediakan untuk kenyamanan, namun, mereka mungkin memerlukan beberapa modifikasi tergantung pada konfigurasi AKS spesifik Anda.

Grup Sumber Daya Node Terkelola Azure: Nilai ini adalah Grup Sumber Daya Terkelola yang dibuat oleh Azure saat Anda membuat Kluster AKS dan bukan Grup Sumber Daya yang Anda tetapkan. Untuk mendapatkan nilai ini, Anda memerlukan CLUSTER_NAME dan RESOURCE_GROUP dari saat Anda membuat cluster. Setelah Anda memiliki nilai-nilai tersebut, Anda dapat memperoleh ini dengan menjalankan:

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

Application Node Pool Nama VMSS: Ini adalah nama Virtual Machine Scaling Set (VMSS) yang terkait dengan AKS Nodepool Anda untuk Aplikasi Wickr. Ini adalah sumber daya yang akan ditingkatkan atau turun berdasarkan kebutuhan cluster Anda. Untuk mendapatkan nilai ini, Anda dapat menjalankan perintah az berikut:

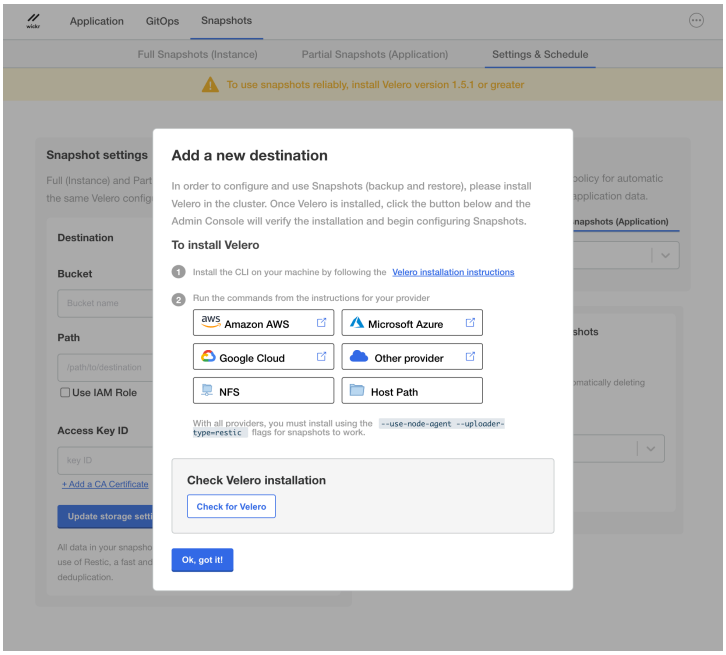
```
CLUSTER_NODEPOOL_NAME="(Your-NodePool-Name)"
CLUSTER_RESOURCE_GROUP="(Your-Managed-Node-Resource-Group-As-Defined-Above>)"
az vmss list -g ${CLUSTER_RESOURCE_GROUP} --query '[?tags."aks-managed-
poolName"==`'''${CLUSTER_NODEPOOL_NAME}'''`].{VMSS_name:name}' -o tsv
```

ACalling Nama VMSS Pool Node (opsional): Ini adalah nama VMSS yang terkait dengan Nodepool panggilan Anda jika Anda memilikinya. Untuk mendapatkan nilai ini, Anda dapat menjalankan versi modifikasi dari perintah untuk Application Node Pool VMSS Name mengganti nilai CLUSTER_NODEPOOL_NAME untuk nama nodepool untuk nodepool panggilan Anda.

# Pencadangan

Wickr Enterprise menggunakan Velero untuk tujuan Backup. Velero menyediakan alat yang diperlukan untuk mencadangkan dan memulihkan sumber daya klaster Kubernetes dan volume persisten, baik yang beroperasi di penyedia cloud maupun di tempat.

Pencadangan Velero dengan Minio: Saat ini cadangan Velero hanya diaktifkan untuk Minio dalam Mode Sumber Daya Rendah.




# Instalasi menggunakan dokumentasi Velero

- Instal CLI Velero. Untuk informasi lebih lanjut, lihat Menginstal CLI Velero.
- Instal Velero di cluster Anda dan konfigurasikan penyimpanan berdasarkan penyedia Anda:
  - AWS.
  - GCP.
  - Azure.
  - Penyedia lainnya.

## Batasan

Secara default, tidak ada volume yang disertakan dalam cadangan. Jika ada pod yang memasang volume yang harus dicadangkan, Anda harus mengonfigurasi cadangan dengan anotasi yang mencantumkan volume tertentu untuk disertakan dalam cadangan.

Untuk setiap volume yang membutuhkan cadangan, tambahkan backup.velero. io/backup-volumes annotation. The annotation name is backup.velero.io/backup-volume dan nilainya adalah daftar volume yang dipisahkan koma untuk disertakan dalam cadangan. Untuk informasi selengkapnya, lihat Mengkonfigurasi Snapshot.

# Instalasi airgap

Wickr Enterprise dan KOTS keduanya mendukung penerapan ke dalam klaster Kubernetes yang sepenuhnya di-airgapped. Anda harus menyediakan akses ke Private Docker Image Registry yang dapat dijangkau dari klaster Kubernetes yang di-airgapped. Private Docker Image Registry yang dipasok ke KOTS harus diamankan dengan username/password otentikasi agar berfungsi dengan benar untuk tujuan ini. KOTS akan menggunakan Private Docker Image Registry untuk meng-host semua gambar Wickr Enterprise.

- Lisensi Wickr Enterprise. Yaml dengan airgap diaktifkan (Hubungi Wickr Sales atau Customer Support Team)
- Paket arsip wickr Enterprise wickr.airgap (Hubungi Wickr Sales atau Tim Dukungan Pelanggan)
- Akses ke Private Docker Image Registry.
- Akses ke klaster Kubernetes yang ditempatkan di lingkungan airgap.
- Kubectl diinstal.
- KOTS CLI diinstal.
- kotsadm.tar.gz diunduh.

Jalankan perintah berikut untuk menerapkan KOTS dan Wickr Enterprise di klaster kubernetes airgapped Anda. Perintah ini mengunggah gambar admin KOTS dan gambar Wickr Enterprise ke Private Docker Image Registry. Setelah perintah selesai, Anda akan diminta untuk mengakses Konsol Admin KOTS untuk menyelesaikan instalasi Wickr Enterprise seperti di atas.

```
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
```

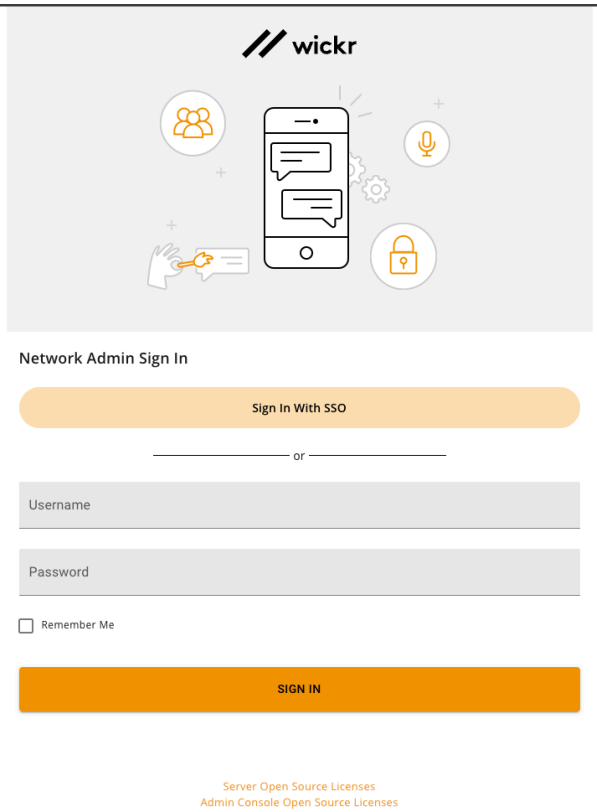
## Pemberitahuan seluler untuk pemasangan airgap

Daftar izin jaringan tambahan diperlukan untuk pemberitahuan push dari backend server ke klien seluler. Persyaratan ini disebabkan oleh bagaimana Apple iOS dan Google Android menerapkan fitur ini untuk perangkat offline dan latar belakang. Lihat dokumentasi untuk layanan ini dan izinkan daftar alamat IP dan port yang ditentukan.

- iOS
- Android

# Konsol admin Wickr

Antarmuka Konsol Admin Wickr digunakan untuk mengelola aplikasi Wickr Enterprise itu sendiri. Ini dapat digunakan untuk mengatur jaringan, pengguna, federasi, dan banyak lagi. Ini dapat diakses melalui HTTPS pada nama DNS yang Anda konfigurasikan untuk menunjuk ke Load Balancer Anda. Nama pengguna default adalah admin, dengan kata sandi Password123. Anda akan diminta untuk mengubah kata sandi ini saat masuk pertama.



# Pengaturan keamanan

AWS Wickr Enterprise menyediakan pengaturan konfigurasi untuk menerapkan konteks keamanan yang ditingkatkan untuk penerapan Anda. Standar keamanan yang lebih tinggi ini diterapkan pada tingkat pod dan kontainer, dan diperlukan agar sesuai dengan Security Technical Implementation Guide (STIG).

Setel parameter konfigurasi berikut untuk menerapkan konteks keamanan yang disempurnakan:

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

> **Warning**
>
> Untuk Opensearch, konfigurasi keamanan ini menonaktifkan `fsgroup-volume` InitContainer yang memperbarui izin pada penyimpanan persisten, yang dapat menyebabkan masalah kompatibilitas yang terkait dengan izin.

# Pertanyaan yang Sering Diajukan

T: Penerapan saya gagal dengan kesalahan berikut di helm stderr:

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

J: Hal ini dapat terjadi ketika Debug Logging diaktifkan. Harap nonaktifkan pencatatan debug, hapus pekerjaan yang bermasalah, dan coba lagi.

# Cluster tertanam untuk Wickr Enterprise

Opsi pemasangan cluster tertanam untuk Wickr Enterprise menyediakan penawaran instalasi kecil dan efisien untuk produk Wickr Enterprise. Ini memanfaatkan Replicated Embedded Cluster untuk menyediakan instalasi Kubernetes kecil menggunakan k0s di mana Wickr Enterprise dapat diinstal. Menggunakan metode instalasi ini meminimalkan persyaratan keterampilan teknis serta persyaratan perangkat keras keseluruhan untuk instalasi Wickr Enterprise dengan memberikan solusi “all-in-one” dengan biaya ketahanan dan ketersediaan tinggi.

Topik


- Memulai dengan cluster tertanam Wickr Enterprise
- Persyaratan cluster tertanam Wickr Enterprise
- Pemasangan cluster tertanam Wickr Enterprise (standar)
- Instalasi Multi-Node
- Konfigurasi konsol admin KOTS
- Persyaratan Instalasi Umum Tambahan
- Pemecahan masalah instalasi cluster tertanam Wickr


## Memulai dengan cluster tertanam Wickr Enterprise

Untuk mulai menggunakan opsi cluster tertanam Wickr Enterprise, hubungi dukungan untuk menerima lisensi. Jika Anda memiliki lisensi yang ada dan ingin menggunakan opsi ini, hubungi dukungan untuk bantuan dalam memperbarui lisensi yang ada dan instruksi instalasi tambahan.

## Persyaratan cluster tertanam Wickr Enterprise

Sebelum Anda mulai menginstal cluster tertanam Wickr Enterprise, verifikasi bahwa persyaratan berikut terpenuhi.

Persyaratan jaringan

Anda harus mengizinkan masuknya ke server Wickr Anda pada port berikut:

- 443/TCP untuk HTTPS

- Hanya Memanggil Proxy TCP - Port proxy TCP yang dikonfigurasi untuk lalu lintas Panggilan TCP di KOTS
- 16384-19999/UDP untuk lalu lintas Panggilan UDP
- Hanya LAN - 30000/TCP untuk Mengakses Konsol Admin KOTS

Persyaratan sistem

Sebelum instalasi, pastikan Anda memiliki VM (Virtual Machine) atau mesin fisik yang menjalankan Sistem Operasi (OS) berbasis Linux dengan sumber daya minimum berikut yang tersedia:

- 8 inti CPU
- RAM 12 gigabyte (GB)
- 100 gigabyte (GB) penyimpanan disk pada partisi/(root)

Cluster tertanam Wickr Enterprise telah diuji pada sistem OS Linux berikut tetapi opsi OS berbasis Linux lainnya mungkin cocok juga:

- Perusahaan Topi Merah Linux 9.5
- Amazon Linux 2023
- Rocky Linux 9.5

# Pemasangan cluster tertanam Wickr Enterprise (standar)

Setelah Anda memiliki instruksi unduhan, unduh bundel Wickr Enterprise ke mesin tujuan dan buka kemasannya.

```
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
```

Anda sekarang harus memiliki dua file, `wickr-enterprise-ha` dan`license.yaml`. `wickr-enterprise-ha`File ini adalah file biner yang mencakup semua bagian yang diperlukan untuk instalasi Cluster Tertanam, sedangkan lisensi Wickr Anda yang akan digunakan untuk memvalidasi instalasi Anda. `license.yaml`

Instalasi dasar dapat dilakukan pada tahap ini dengan menjalankan `wickr-enterprise-ha` file:

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

Setelah proses instalasi dimulai, Anda diminta untuk memasukkan kata sandi Konsol Admin. Masukkan kata sandi yang aman dan pastikan Anda menyimpannya karena Anda akan membutuhkannya saat mengakses Konsol Admin KOTS untuk melanjutkan konfigurasi instalasi Anda.

Setelah instalasi selesai, outputnya menyerupai yang berikut:

```
   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
```

Setelah instalasi standar, lanjutkan ke URL konsol admin KOTS yang disediakan dalam output menggunakan browser web. Untuk contoh ini, URL adalah`http://192.168.1.100:30000`. Namun URL Anda akan berbeda berdasarkan konfigurasi jaringan Anda.

## Instalasi Multi-Node

Instalasi Wickr Enterprise Embedded Cluster Multi-Node menyediakan opsi bagi pengguna Embedded Cluster untuk memisahkan beban kerja Wickr Calling dan Wickr Messaging ke mesin fisik yang berbeda. Untuk melakukan ini, Wickr Enterprise memanfaatkan perkakas Multi-Node Cluster Tertanam yang Direplikasi.

## Persyaratan port

Port berikut harus terbuka pada semua anggota cluster agar fungsionalitas Multi-Node berfungsi dengan benar. Ini hanya perlu terbuka di antara node itu sendiri, dan tidak terbuka ke internet yang lebih luas.

- 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

## Persyaratan lisensi

Opsi konfigurasi Multi-Node Cluster Tertanam Wickr memerlukan hak lisensi tambahan. Hubungi Dukungan untuk memastikan lisensi Anda mendukung fitur ini.

## Membuat node tambahan selama penyiapan awal

Ketika Anda awalnya mengkonfigurasi Wickr Enterprise Embedded Cluster, Anda dapat membuat node panggilan tambahan selama proses penyiapan. Mulailah dengan mengikuti prosedur yang dijelaskan dalam Instalasi cluster tertanam Wickr Enterprise (standar) Saat Anda menavigasi ke panel admin KOTS, Anda akan diminta untuk membuat node tambahan.

> **Note**
>
> Saat ini, Embedded Cluster Multi-Node hanya mendukung 1 node panggilan dan 1 messaging/controller node.

Untuk memulai, batalkan pilihan peran Controller dan pilih opsi Calling role. Ini mengisi set instruksi tambahan untuk mengkonfigurasi node baru. Jalankan instruksi ini pada node baru untuk mengkonfigurasinya untuk bergabung dengan cluster sebagai node panggilan.

Jalankan instruksi yang mirip dengan contoh berikut pada node baru:

1. Unduh biner pada node baru:

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

2. Ekstrak biner:

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

3. Bergabunglah dengan node ke cluster:

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

Setelah perintah join berhasil diselesaikan, node baru muncul di halaman Configure cluster dengan peran Calling ditetapkan. Pilih Lanjutkan untuk melanjutkan ke halaman konfigurasi Wickr Enterprise. Ikuti petunjuk untuk opsi konfigurasi node tertanam yang diuraikan dalam konfigurasi konsol admin KOTS.

## Menambahkan node tambahan ke instalasi cluster tertanam yang ada

Untuk menambahkan node panggilan ke instalasi Wickr Enterprise Embedded Cluster yang ada, navigasikan ke Konsol Admin KOTS. Untuk melakukan ini, masuk ke node melalui ssh atau mekanisme lain dan arahkan ke direktori instalasi yang berisi `wickr-enterprise-ha` biner yang digunakan untuk instalasi. Jalankan `./wickr-enterprise-ha admin-console` untuk memulai Konsol Admin KOTS. Jika perintah ini tidak mengembalikan output apa pun, Konsol Admin KOTS sudah berjalan dan dapat diakses dengan menavigasi ke port 30000 pada IP node di browser web, misalnya:. `https://127.0.0.1:30000/`

Masukkan kata sandi Admin KOTS saat diminta, lalu lakukan prosedur berikut untuk membuat node tambahan:

1. Setelah masuk, navigasikan ke halaman Manajemen Cluster di kiri atas Konsol Admin KOTS.
2. Pilih Tambahkan simpul.

3. Hapus pilihan Controller di bawah`Roles`.
4. Pilih Panggilan di bawah `Roles`
5. Ikuti instruksi yang diberikan untuk menjalankan perintah pada node baru yang ingin Anda tambahkan.
6. Setelah selesai, pilih Tutup
7. Node baru Anda muncul di daftar Node dengan peran Panggilan.
8. Arahkan ke halaman Aplikasi di kiri atas Konsol Admin KOTS
9. Pilih Config dari bilah navigasi di bagian atas halaman.
10. Arahkan ke bagian Panggilan di panel navigasi kiri.
11. Pilih Memerlukan Node Panggilan untuk mengizinkan penggunaan node Panggilan.
12. Gulir ke bagian bawah halaman dan pilih Simpan konfigurasi.
13. Sebuah popup muncul, menunjukkan Config telah diperbarui. Pilih Pergi ke versi yang diperbarui.
14. Pada halaman versi yang diperbarui, versi yang saat ini diinstal ditampilkan. Item baris baru tercantum di bawah versi yang diinstal dengan penunjukan Config Change. Pilih Deploy untuk menyebarkan versi baru ini dan aktifkan node panggilan baru.

# Konfigurasi konsol admin KOTS

Konsol admin KOTS awalnya menggunakan sertifikat yang ditandatangani sendiri, yang harus Anda izinkan sebagai pengecualian di browser Anda. Setelah Anda menerima pengecualian ini, Anda akan disambut oleh Configuration Wizard untuk konsol admin KOTS. Panduan ini memandu Anda melalui langkah-langkah konfigurasi tambahan untuk mengonfigurasi perilaku Konsol Admin KOTS, termasuk opsi untuk menambahkan sertifikat khusus jika perlu.

Setelah konfigurasi awal konsol admin KOTS selesai, Anda diminta untuk memasukkan kata sandi konsol admin yang Anda buat selama proses instalasi. Setelah login pertama Anda, Anda perlu mengkonfigurasi cluster.

Pilih Lanjutkan untuk melanjutkan ke konsol admin KOTS untuk Wickr.

Untuk satu node Embedded Cluster, pilih Lanjutkan untuk melanjutkan ke konsol admin KOTS untuk Wickr. Untuk instalasi multi-node, lihat Instalasi Multi-Node.

Setelah berada di konsol admin KOTS, konfigurasikan instalasi Anda sesuai dengan kebutuhan Anda. Saat menggunakan penawaran cluster tertanam ada beberapa pengaturan konfigurasi kunci yang harus diatur untuk memastikan fungsionalitas yang tepat dari instalasi Wickr Enterprise Anda.

- Nama host - Ini adalah nama host yang Anda gunakan saat berkomunikasi dengan instalasi Wickr. Pastikan Anda membuat catatan DNS yang sesuai untuk domain ini untuk mengarah ke instalasi Wickr Enterprise Anda.
- Di bawah Advanced Options, centang opsi [] Configure Ingress Controller untuk mengekspos blok konfigurasi untuk mengkonfigurasi Ingress Kubernetes. Di blok konfigurasi Ingress, pilih Single Node Embedded Cluster, lalu masukkan IP “publik” yang terkait dengan server Wickr Anda di kotak teks berlabel Loadbalancer External IP (Only). IPv4

  Jika Anda tidak yakin apa IP ini, Anda dapat menjalankan perintah berikut dari baris perintah di server Wickr untuk menentukan nilai ini: `ip route get 1.1.1.1|awk '{print $7}'`
- Di bawah Opsi Lanjutan, centang opsi Aktifkan Mode Sumber Daya Rendah.
- Di bawah Panggilan, jika Anda menggunakan satu node Embedded Cluster, pastikan Memerlukan Node Panggilan tidak dipilih. Jika tidak, jika Anda telah menambahkan node Panggilan selama penyiapan awal, pastikan Memerlukan Node Panggilan dipilih.
- Jika Anda menginginkan solusi all in one yang tidak menggunakan Database eksternal atau penyimpanan Kompatibel S3 untuk berbagi file, pilih opsi Internal untuk pengaturan berikut:
  - Basis Data
  - Lokasi Penyimpanan S3

    Lokasi penyimpanan S3 internal menyediakan opsi tambahan untuk mengonfigurasi kapasitas penyimpanan. Disarankan untuk memulai dari yang kecil dan memperluas seperlunya, karena penskalaan bukan merupakan pilihan setelah penyediaan.

Setelah Anda mengonfigurasi semua fitur yang diperlukan, gulir ke bagian bawah halaman konfigurasi, dan pilih Simpan Konfigurasi. Ini akan memulai beberapa pemeriksaan host preflight. Setelah pemeriksaan preflight selesai, pilih Deploy untuk memulai Instalasi Wickr Enterprise.

Sekarang Anda siap untuk mulai mengkonfigurasi instalasi Wickr Enterprise Anda. Untuk informasi lebih lanjut tentang mengonfigurasi Wickr Enterprise, lihat Apa itu Wickr Enterprise? .

# Persyaratan Instalasi Umum Tambahan

Instalasi IP Hostname

Jika instalasi Anda memerlukan nama host berbasis IP, ada beberapa opsi konfigurasi tambahan. Instruksi ini khusus untuk nama host berbasis IP, dan disarankan Anda mengikuti instruksi lain untuk pengaturan dasar yang tercantum di atas.

Di panel admin KOTS, selesaikan langkah-langkah berikut.

1. Atur nama host ke IP yang akan Anda gunakan.
2. Di bawah Sertifikat, pilih Unggah Sertifikat. Kemudian, buat sertifikat yang ditandatangani sendiri mengikuti instruksi untuk sertifikat berbasis IP. Untuk informasi selengkapnya, lihat Membuat sertifikat yang ditandatangani sendiri.
3. Unggah `.crt` file untuk Sertifikat dan `.key` file untuk kunci Pribadi
4. Untuk Rantai Sertifikat, unggah `.crt` file lagi.
5. Centang Setel kotak centang sertifikat yang disematkan.
6. Unggah `.crt` untuk Sertifikat yang Disematkan.
7. Di bawah Panggilan, hapus centang Secara otomatis menemukan alamat IP publik server dan Gunakan alamat IP utama host untuk Memanggil kotak centang lalu lintas.
8. Di bawah Calling, masukkan alamat IP dari nama host di kotak teks Hostname Override.
9. Di bawah Opsi Lanjutan, centang kotak centang Configurate Ingress Controller. Bagian konfigurasi baru yang disebut Ingress muncul di bawah ini.
10. Di bawah Ingress, pilih Single Node Embedded Cluster.
11. Di bawah Ingress, masukkan IP untuk antarmuka 'publik' di server Wickr. Ini mungkin berbeda dari IP yang digunakan sebagai nama host Anda. Lihat informasi tambahan tentang nilai ini di langkah-langkah konfigurasi dasar.
12. Di bawah Ingress, periksa Gunakan nama host wildcard.

### SELinux Mode Penegakan

Jika Anda memerlukan penggunaan SELinux dalam mode penegakan, ubah direktori data default yang digunakan untuk menginstal cluster tertanam. Disarankan untuk digunakan `/opt` karena telah diuji untuk bekerja dengan sebagian besar SELinux kebijakan untuk kasus penggunaan ini.

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

Pemeriksaan preflight instalasi default cluster tertanam yang direplikasi akan mencoba memvalidasi yang SELinux dalam mode permisif dan gagal jika dalam Menegakkan. SELinux Untuk melewati ini, diperlukan untuk menggunakan argumen baris `--ignore-host-preflights` perintah. Saat menggunakan opsi baris perintah, ada prompt yang mirip dengan yang di bawah ini. Masukkan Ya saat diminta.

```
#  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 instalasi

Opsi instalasi cluster tertanam untuk Wickr Enterprise mendukung instalasi airgapped. Konfigurasi dan pengaktifan tambahan untuk lisensi Anda diperlukan. Hubungi dukungan jika Anda tertarik untuk menggunakan cluster tertanam Wickr Enterprise di lingkungan airgapped.

Saat melakukan instalasi airgap, instruksi unduhan berbeda dari metode instalasi standar. Mereka harus menyerupai yang berikut:

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

Unduh bundel ke mesin yang memiliki akses internet, lalu transfer ke lingkungan airgapped Anda menggunakan metode transportasi data pilihan Anda. Setelah bundel ditransfer, ekstrak seperti yang Anda lakukan dengan bundel instalasi standar apa pun. File ketiga`wickr-enterprise-ha.airgap`, yang berisi semua gambar layanan aplikasi Wickr Enterprise terkait akan disertakan.

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

Selama instalasi, perlu untuk mengatur argumen baris `--airgap-bundle` perintah setelah ekstraksi, jika tidak, proses mengikuti prosedur instalasi standar.

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

Memperbarui klaster tertanam AirGapped

Untuk memperbarui cluster AirGapped Tertanam, selesaikan langkah-langkah berikut.

1. Unduh paket cluster tertanam baru dari Replicated, dan transfer ke mesin host menggunakan metode transfer data standar Anda untuk lingkungan airgapped Anda. Setelah bundel baru ada di mesin host, ekstrak tarball:

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

2. Jalankan pembaruan menggunakan bundel biner dan airgap baru:

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

3. Mulai konsol admin KOTS, dan masuk ke URL yang disediakan menggunakan metode standar Anda untuk mengakses konsol admin KOTS

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

4. Setelah masuk ke Konsol Admin KOTS, temukan Pembaruan Terbaru yang Tersedia di sebelah kiri di bawah Versi, lalu tekan tombol Buka Riwayat Versi.
5. Pilih Terapkan untuk versi baru di bawah Pembaruan yang Tersedia. Berjalan melalui layar:
   1. Ubah opsi konfigurasi apa pun, gulir ke bawah, lalu pilih Berikutnya.
   2. Verifikasi tidak ada pemeriksaan preflight yang gagal, pilih Berikutnya: Konfirmasi dan terapkan.
   3. Pilih Deploy.

Catatan tambahan tentang cluster tertanam Wickr Enterprise

- NAMESPACE : Tidak seperti kebanyakan instalasi Wickr Enterprise, instalasi cluster tertanam menginstal aset Wickr ke namespace kotsadm di kubernetes dan bukan wickr. Ubah skrip atau perintah apa pun yang telah Anda simpan yang digunakan `-n wickr` untuk kubectl, helm, atau utilitas lainnya untuk digunakan. `-n kotsadm`
- Berinteraksi dengan Cluster Kubernetes: Dari mesin host, gunakan `./wickr-enterprise-ha` biner untuk membuat shell dengan variabel yang sesuai yang disetel untuk berinteraksi dengan instalasi Kubernetes dengan menjalankan. ./wickr-enterprise-ha shell Ini akan menyediakan utilitas kubectl di dalam PATH shell dan mengatur konfigurasi kube yang sesuai ke instalasi lokal.

# Pemecahan masalah instalasi cluster tertanam Wickr

Semua contoh langkah pemecahan masalah ini mengasumsikan Anda memiliki akses shell ke instance yang menjalankan instalasi Wickr Embedded Cluster dan telah menjalankan `./wickr-enterprise-ha shell` perintah untuk dapat berinteraksi dengan instalasi Kubernetes secara langsung.

## Masalah umum

Tambahkan Tombol Node Hilang Dari Layar Manajemen Cluster

Pemasangan Airgapped

Jika Anda sedang dalam instalasi airgap, silakan hubungi Wickr Support untuk bantuan dalam memperbaiki perilaku ini.

Instalasi Standar

Jika lisensi Anda menyertakan hak Multi-Node Cluster Tertanam, lakukan sinkronisasi lisensi untuk mendapatkan versi terbaru. Jika Anda tidak yakin atau tidak memiliki hak ini, silakan hubungi Dukungan Wickr.

Untuk melakukan sinkronisasi lisensi, selesaikan langkah-langkah berikut.

1. Arahkan ke panel kontrol KOTS.
2. Pada halaman Dasbor, cari bagian lisensi di area kanan atas halaman.
3. Di bagian ini, di sudut kanan atas, Anda akan melihat hyperlink Lisensi Sinkronisasi. Pilih hyperlink.

4. Setelah lisensi disinkronkan, pembaruan UI dan Terakhir disinkronkan beberapa detik yang lalu muncul.
5. Pilih Redeploy dari bagian Versi pada halaman dasbor KOTS.
6. Setelah redeploy selesai, navigasikan kembali ke manajemen Cluster dan Anda dapat menambahkan node.

## Masalah peningkatan

Upgrade Terjebak pada Upgrade Cluster

Jika pemutakhiran Anda macet di Upgrade Cluster, ini kemungkinan berarti beberapa pod tidak dihentikan dengan tepat. Masuk ke instance dan gunakan `./wickr-enteprise-ha shell` perintah untuk masuk ke lingkungan shell untuk mengelola instalasi kubernetes.

1. Identitas pod yang masih berjalan:

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

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

   > **Note**
   >
   > Jika salah satu pod yang sedang berjalan adalah `embedded-cluster-upgrade-XXXXXXXXXXXXX-xxxxx` `kotsadm-xxxxxxx` atau atau serupa, jangan menghapusnya karena pod ini diperlukan untuk melakukan upgrade.

3. Pastikan tidak ada pod yang berjalan yang tersisa.

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

Prosedur ini harus memungkinkan upgrade cluster untuk melanjutkan dengan upgrade Wickr.

Aplikasi Tidak Diperbarui Selama Upgrade Cluster Dan Tidak Dapat Menyebarkan Versi Baru

Jika aplikasi tetap pada versi lama setelah upgrade, versi baru mungkin dalam keadaan tidak konsisten.

Periksa catatan instalasi Kubernetes:

1. Buka shell Kubernetes dari installer.

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

2. Jalankan perintah kubectl berikut:

```
kubectl get installations
```

3. Output akan terlihat seperti ini:

```
      [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. Hapus instalasi yang gagal.

```
kubectl delete installation 20251113180133
```

5. Coba jalankan upgrade lagi melalui panel Admin KOTS.

### RabbitMQ Pod Gagal dengan baris log **`Error while waiting for Mnesia tables: {timeout_waiting_for_tables}`**

Rahasia dan penyimpanan RabbitMQ tidak sinkron. Ini biasanya terjadi ketika beberapa instance RabbitMQ berjalan dan menyebabkan pemilihan pemimpin atau kesalahan kuorum. Untuk memperbaikinya, hapus layanan RabbitMQ dan volume penyimpanannya, lalu gunakan kembali.

Untuk menghapus RabbitMQ yang gagal, selesaikan langkah-langkah berikut.

1. Hapus Statefulset RabbitMQ.

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

2. Hapus pod RabbitMQ yang tersisa. Jika ada beberapa pod RabbitMQ-X yang berjalan, keluarkan perintah ini beberapa kali memperbarui nilai RabbitMQ-X agar sesuai dengan nama pod tambahan.

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

3. Hapus yang sesuai PVCs. Jika ada beberapa pod yang berjalan, keluarkan perintah ini beberapa kali untuk memperbarui data-RabbitMQ-X agar sesuai dengan pod yang sesuai.

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

4. Periksa apakah ada pod yang tersisa, ini seharusnya tidak menghasilkan apa-apa jika berhasil.

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

5. Periksa apakah ada yang tersisa PVCs, ini seharusnya tidak menghasilkan apa-apa jika berhasil.

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

6. Menerapkan ulang melalui Panel Admin KOTS.

Untuk informasi pemecahan masalah lainnya, lihat Pemecahan Masalah.

# Riwayat dokumen

Tabel berikut menjelaskan rilis dokumentasi untuk Wickr Enterprise Automated Install Guide.

| Perubahan | Deskripsi | Tanggal |
| --- | --- | --- |
| Pengaturan keamanan | Pengaturan keamanan telah ditambahkan. Untuk informasi selengkapnya, lihat Pengaturan keamanan. | Agustus 26, 2025 |
| Instalasi Multi-Node | Instalasi Multi-Node telah ditambahkan. Untuk informasi selengkapnya, lihat Instalasi Multi-Node. | Agustus 26, 2025 |
| Memanggil pengaturan ingress | Pengaturan masuk panggilan telah ditambahkan. Untuk informasi selengkapnya, lihat Memanggil setelan ingress. | Agustus 26, 2025 |
| Opsi penyebaran otomatis | Opsi penyebaran otomatis telah ditambahkan. Untuk informasi selengkapnya, lihat Menginstal Wickr Enterprise. | Februari 23, 2024 |
| Port ke daftar yang diizinkan | Port TCP/8443 telah ditambahkan ke daftar yang diizinkan. Untuk informasi selengkapnya, lihat Persyaratan. | Februari 12, 2024 |
| Menghancurkan sumber daya dan Port untuk daftar yang diizinkan | Petunjuk tentang cara menghancurkan sumber daya telah ditambahkan. Untuk informasi selengkapnya, lihat Menghancurkan sumber daya. Selain itu, port ke allowlist | 17 Agustus 2023 |

| | | |
|---|---|---|
| | telah ditambahkan. Untuk informasi selengkapnya, lihat Persyaratan. | |
| Rilis awal | Rilis awal Panduan Instalasi Otomatis Wickr Enterprise | 4 Agustus 2023 |

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.