

Sino Control Hub on Kubernetes

The hub as one pod behind your cluster's ingress, with your Postgres, MySQL or MariaDB: AKS, EKS, GKE or on-premises

Sino Privacy Vault (<https://sinosecure.app>)

1. Before You Install

- **The image and the chart.** The hub image (`sino-hub:<version>`) and its Helm chart (the `sino-hub` folder of the deploy bundle) come from Sino with early access. Push the image to your registry.
- **A database and an account for the hub.** Postgres, MySQL 8.0.16 or later, or MariaDB 10.5 or later. The hub makes its tables at start. Use TLS: `?sslmode=require` on a Postgres address, `?sslMode=verify-full` on a MySQL or MariaDB one (see the *Deployment Guide*, section 5).
- **An ingress controller with HTTPS, and a DNS name for the hub.** The name is permanent: every enrollment link and every device holds it.

2. Install

```

kubectl create namespace sino-hub
kubectl -n sino-hub create secret generic sino-hub-master --from-
literal=master_key="$(openssl rand -base64 32)"
kubectl -n sino-hub create secret generic sino-hub-db --from-
literal=password='<the hub account password>'

helm install sino-hub ./sino-hub -n sino-hub \
  --set image.repository=registry.example.com/sino-hub --set image.tag=
<version> \
  --set masterKey.existingSecret=sino-hub-master \
  --set
database.url='jdbc:postgresql://postgres.example.internal:5432/sino_hub?
sslmode=require' \
  --set database.existingSecret=sino-hub-db \
  --set ingress.host=hub.example.com --set ingress.className=nginx

kubectl -n sino-hub logs deploy/sino-hub | grep "setup code"

```

Open <https://hub.example.com>, enter the setup code and register your passkey. You are now the owner. Then sign the hub's certificate in the Sino app, as in the *Deployment Guide*, section 3, step 5.

Back up the master key Secret somewhere outside the cluster. Without it the database cannot be opened, and the hub's keys and every stored credential are lost.

3. What the Chart Sets Up

Resource	Notes
Deployment	One replica, <code>Recreate</code> : the hub's timers, rate limits and setup code live in its process, so two copies must never run at once. Kubernetes restarts it if it stops. It runs as a non-root user (10001) with a read-only root filesystem, no capabilities and no service account token.
Probes	<code>GET /healthz</code> , which answers 200 while the database does.

Secrets	The master key and database password are mounted as files (<code>..._FILE</code> settings), not put in the environment.
Service and Ingress	HTTP inside the cluster, HTTPS at the ingress.
NetworkPolicy (optional)	Traffic in only from the ingress controller's namespace. Traffic out only to DNS, 443 (storage, SIEM, Entra ID) and the database port. Add 6514 to <code>networkPolicy.extraEgressPorts</code> for syslog.
PersistentVolumeClaim (only with <code>sqlite.enabled</code>)	For test clusters without a database server. It is kept when the chart is uninstalled.

Settings are listed in the chart's `values.yaml`. The chart refuses to install without the master key Secret and a database.

4. Ingress Notes

- **Request size.** Devices send audit uploads of up to about 1 MB, and the hub accepts up to 4 MB. With ingress-nginx, add the annotation `nginx.ingress.kubernetes.io/proxy-body-size: 4m`.
- **Streams.** PCs keep a streamed answer open for up to 3 minutes, with a line every 20 seconds. ingress-nginx passes it as it comes (the hub sends `X-Accel-Buffering: no`); keep `proxy-read-timeout` at 60 seconds or more.
- **Client addresses.** `trustProxy: true` makes rate limits count the address the ingress appends to `X-Forwarded-For`. ingress-nginx, Traefik and the cloud load balancers append it. If nothing in front sets the header, set `trustProxy: false`. Otherwise a client could dodge the limits with a forged header.
- **Certificates.** With cert-manager, add `cert-manager.io/cluster-issuer: <issuer>` to `ingress.annotations`. It makes the `ingress.tls.secretName` Secret for you.

5. Upgrade and Uninstall

`helm upgrade` replaces the pod, so the hub is down for a few seconds. Devices retry. `helm uninstall` leaves the database, the Secrets and any data volume in place.