

Sino Control Hub on AWS

The hub as one ECS Fargate task behind an Application Load Balancer, with Amazon RDS as its database. App Runner works too.

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

These steps have not yet been checked against a live AWS account. Tell us anything that differs.

1. Image

Push the hub image (`sino-hub:<version>` , from Sino with early access) to ECR:

```
aws ecr create-repository --repository-name sino-hub
aws ecr get-login-password | docker login --username AWS --password-
stdin <account>.dkr.ecr.<region>.amazonaws.com
docker tag sino-hub:<version> <account>.dkr.ecr.
<region>.amazonaws.com/sino-hub:<version>
docker push <account>.dkr.ecr.<region>.amazonaws.com/sino-hub:<version>
```

2. Postgres

Make an RDS for PostgreSQL instance in private subnets. Create a `sino_hub` database and role, and allow port 5432 only from the hub's security group. RDS uses TLS. The address is `jdbc:postgresql://<endpoint>:5432/sino_hub?sslmode=require` .

Prefer MySQL or MariaDB? Use RDS for MySQL, RDS for MariaDB or Aurora MySQL (port 3306) instead, with a `jdbc:mysql://<host>:3306/sino_hub?sslMode=verify-full` address (see the *Deployment Guide*, section 5). RDS certificates come from Amazon's own CA, so add `serverSslCert=` with the RDS CA bundle. The rest is the same.

3. Secrets

```
aws secretsmanager create-secret --name sino-hub/master-key --secret-string "$(openssl rand -base64 32)"
aws secretsmanager create-secret --name sino-hub/db-password --secret-string '<sino_hub role password>'
```

Back up the master key outside AWS as well. Without it the database cannot be opened.

4. ECS Service

The task definition, in outline. The execution role needs `secretsmanager:GetSecretValue` on both secrets:

```

{
  "family": "sino-hub",
  "requiresCompatibilities": ["FARGATE"],
  "networkMode": "awsvpc",
  "cpu": "512", "memory": "1024",
  "containerDefinitions": [{
    "name": "hub",
    "image": "<account>.dkr.ecr.<region>.amazonaws.com/sino-hub:
<version>",
    "portMappings": [{ "containerPort": 8080 }],
    "readOnlyRootFilesystem": true,
    "environment": [
      { "name": "SINO_HUB_DATABASE_URL", "value":
"jdbc:postgresql://<endpoint>:5432/sino_hub?sslmode=require" },
      { "name": "SINO_HUB_DATABASE_USER", "value": "sino_hub" },
      { "name": "SINO_HUB_TRUST_PROXY", "value": "true" }
    ],
    "secrets": [
      { "name": "SINO_HUB_MASTER_KEY", "valueFrom":
"arn:aws:secretsmanager:<region>:<account>:secret:sino-hub/master-key-
XXXX" },
      { "name": "SINO_HUB_DATABASE_PASSWORD", "valueFrom":
"arn:aws:secretsmanager:<region>:<account>:secret:sino-hub/db-password-
XXXX" }
    ],
    "mountPoints": [{ "sourceVolume": "tmp", "containerPath": "/tmp" }],
    "logConfiguration": { "logDriver": "awslogs", "options": {
      "awslogs-group": "/sino-hub", "awslogs-region": "<region>",
      "awslogs-stream-prefix": "hub" } }
  }],
  "volumes": [{ "name": "tmp" }]
}

```

Create the service with **desired count 1** and a deployment of `minimumHealthyPercent: 0`, `maximumPercent: 100`. An update then stops the old task before it starts the new one, so two hubs never run at once.

The load balancer:

- an HTTPS listener with an ACM certificate for `hub.example.com` ;
- a target group on port 8080 (target type `ip`), with health check path `/healthz` ;
- the default idle timeout of 60 seconds is enough: a PC's stream carries a line every 20 seconds.

The ALB appends the client's address to `X-Forwarded-For` , so

`SINO_HUB_TRUST_PROXY=true` is right. Point `hub.example.com` at the ALB. This address is permanent, because every device keeps it.

Read the setup code from CloudWatch Logs (`/sino-hub` , the line "enter this setup code").

Then open the address, claim the hub with your passkey, and sign its certificate in the Sino app (*Deployment Guide*, section 3, step 5).

5. App Runner Instead

App Runner builds the HTTPS front for you:

- Service port 8080, health check HTTP on `/healthz` .
- An auto scaling configuration with minimum and maximum size 1.
- The two secrets as runtime environment secrets from Secrets Manager.
- A VPC connector to reach RDS.

App Runner only gives an instance CPU while it is handling requests. Between requests, the hub's timers can stall: SIEM delivery, recovery-copy mirroring and clean-up. Devices' check-ins usually keep it busy enough. Choose ECS if events must reach the SIEM promptly even when no device is online.

6. Storage

Add the company's S3 in the console, under **Storage**, as an STS target. Give it an IAM user's access key and the ARN of a role that the user may assume. The role needs `s3:GetObject` , `s3:PutObject` , `s3:DeleteObject` , `s3:ListBucket` and the multipart actions on the bucket. The hub narrows each device's session to its own vault's folder, or to one shared

folder (read-only for its viewers), for the session length you set (1 hour by default, up to the role's maximum session duration).