SCOPE

ON-PREM DOCUMENTATION

ScopeAR Coturn on Kubernetes

Version	3.97.5
Channel	on-prem
Mode	production
Revision	3c4d851
Generated	2026-09-29

Confidential · For authorized Scope AR customers and partners

Contents

ScopeAR Coturn on Kubernetes (AKS, EKS, OpenShift, and Generic)

Overview

Prerequisites

Enable Coturn

- Defaults

What ScopeAR configures automatically

Platform guidance

- Azure Kubernetes Service (AKS)
- Amazon Elastic Kubernetes Service (EKS)
- OpenShift
- Generic Kubernetes and on-premises

Firewall and network ports

DNS and the advertised host

Credentials

Verify the installation

Common problems

Contact

- Samples
  - `scopear_k8s_onprem_sample.yaml`
  - `scopear_k8s_sample.yaml`
  - `scopear_k8s_eks_sample.yaml`

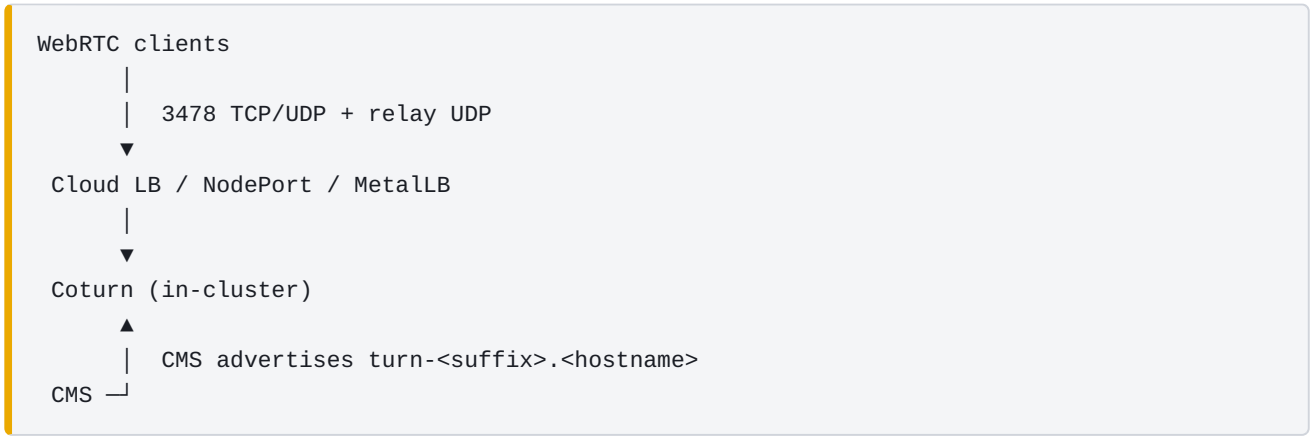
ScopeAR Coturn on Kubernetes (AKS, EKS, OpenShift, and Generic)

Overview

Coturn is ScopeAR's in-cluster TURN/STUN server. WebRTC clients use it when they cannot establish a direct peer connection (for example, when both parties are behind NAT or restrictive firewalls).

When `spec.coturn.enabled` is `true` in your ScopeAR environment manifest, `scopearctl k8s apply` deploys Coturn into your cluster and wires CMS so clients are told to use that server.

TURN traffic is **UDP** (and TCP on the signaling port). It does **not** go through Istio ingress gateways or OpenShift Routes. Your HTTP applications (CMS, Create, and so on) continue to use their usual ingress path; Coturn is exposed separately via a cloud LoadBalancer or NodePort.



Prerequisites

Before enabling Coturn, you should already have:

- A working ScopeAR Kubernetes installation managed with `scopearctl`
- A ScopeAR environment manifest (`apiVersion: k8s.scopear.com/v1`) for your cluster
- Control over DNS for the hostname clients will use (typically `turn-<suffix>.<your domain>`)
- Ability to open firewall, NSG, or security-group rules for TURN signaling and relay ports

You should have been provided the ScopeAR Kubernetes installation guide and the environment manifest reference (`k8s_scopear_env_manifest_configs.pdf`). Your ScopeAR representative can provide samples for your specific platform if needed.

If you do not have `scopearctl` or authorized image access, contact Scope AR support.

Enable Coturn

Add (or update) the `coturn` block in your ScopeAR manifest:

```
spec:
  cluster:
    provider: azure   # aws | azure | google | other
    type: aks         # eks | aks | openshift | gke | ...
  hostnames:
    - example.com
  subdomainSuffix: qa0   # optional; advertised host becomes turn-qa0.example.com

  coturn:
    enabled: true
    replicas: 1
    config:
      port: 3478
      min_port: 49152
      max_port: 49251
      username: <turn-username>
      password: <turn-password>
```

Then apply:

```
scopearctl k8s apply -f /path/to/scopear-manifest.yaml
```

To generate manifests without applying them:

```
scopearctl k8s apply -f /path/to/scopear-manifest.yaml --dry
```

Defaults

FIELD	DEFAULT
`enabled`	`false`
`auto_secret_gen`	`true`
`replicas`	`1`
`config.port`	`3478`
`config.min_port` / `max_port`	`49152` / `49251` (100 ports)

FIELD	DEFAULT
Advertised host	`turn-<subdomainSuffix>.<primary hostname>` when `config.external_host` is unset

Relay port span limit: `max_port - min_port + 1` must be **200 or fewer**. Cloud load balancers list each relay port individually; larger windows can exceed provider rule limits and fail validation.

Do **not** copy the single-node or Swarm relay range `49152–50000` (849 ports). That range is invalid for Kubernetes Coturn and is far larger than typical cloud LB limits. Keep the default window unless Scope AR advises otherwise.

What ScopeAR configures automatically

When `coturn.enabled` is `true`:

1. Coturn is deployed as a Kubernetes Service and Deployment.
2. CMS is configured so **SERVER_1** TURN settings point at the in-cluster Coturn host, port, and protocol—unless you already set a custom `cms.config.coturn.turn_external_host` (explicit values are not overwritten).
3. Credentials are stored in Secret `scopear-turn` (when `auto_secret_gen` is `true`, the default).

If multi-region TURN is already enabled (`cms.config.coturn.enable_multi_region`), SERVER_2 and later continue to use ScopeAR's shared regional proxies. Only SERVER_1 is switched to your in-cluster Coturn when Coturn is enabled.

Platform guidance

Service type and load-balancer behavior follow `cluster.provider`, optional `coturn.annotations`, and whether the install is treated as on-prem—not the cluster "type" alone.

Azure Kubernetes Service (AKS)

Set `cluster.provider: azure` (typically with `cluster.type: aks`).

ScopeAR creates a **public** Azure LoadBalancer Service for Coturn and applies an Azure DNS label for the load balancer. You still need DNS that clients will use:

- Create an **A** or **CNAME** record from `turn-<suffix>.<hostname>` to the LoadBalancer public IP (or to the Azure `*.cloudapp.azure.com` name), **or**

- Set `coturn.config.external_host` to the hostname you want advertised and ensure DNS resolves to the LB

Open your Azure NSG (and related network rules) for the ports listed under Firewall and network ports. Coturn uses `externalTrafficPolicy: Local`, so node-level network rules may also need those ports.

Amazon Elastic Kubernetes Service (EKS)

Set `cluster.provider: aws` (typically with `cluster.type: eks`).

ScopeAR creates an internet-facing **AWS Network Load Balancer (NLB)** and annotates the Service for **external-dns** so the hostname `turn-<suffix>.<hostname>` can be published automatically when external-dns is installed and allowed to manage your DNS (for example Route53).

Ensure:

- Security groups allow the signaling and relay ports below to the load balancer (and, where required by Local traffic policy, to the nodes)
- DNS for the advertised TURN hostname resolves to the NLB

OpenShift

OpenShift **Routes** are for HTTP(S) and cannot front TURN/UDP traffic. Do not try to expose Coturn through a Route.

Coturn on OpenShift uses the same Kubernetes Service rules as elsewhere, based on provider:

OPENSHIFT PLACEMENT	TYPICAL CR	COTURN SERVICE
OpenShift on AWS	`provider: aws`, `type: openshift`	LoadBalancer (AWS NLB), same pattern as EKS
On-prem / bare-metal OpenShift	`provider: other`, usually `on-prem: true`	**NodePort**, unless you set `coturn.annotations` (for example MetalLB)

DNS and firewall requirements are the same as for generic Kubernetes: the advertised hostname must reach the LB or NodePort path on UDP/TCP as required.

If the Coturn pod fails to start because of OpenShift Security Context Constraints (SCC), contact Scope AR support rather than guessing cluster SCC overrides.

Generic Kubernetes and on-premises

When `on-prem` is true and `cluster.provider` is **not** `aws` or `azure`, and you have **not** set `coturn.annotations`, Coturn is exposed as **NodePort**.

Options:

1. **MetalLB or another in-cluster load balancer** — set `coturn.annotations` to whatever your LB controller requires. When annotations are present, the Service type is **LoadBalancer** and your annotations are used as-is (they override the built-in AWS/Azure annotations).
2. **External UDP-capable load balancer** — front the Coturn NodePorts and point DNS at that external LB.
3. **Direct NodePort access** — only for tightly controlled networks; still open the full signaling and relay port set to the nodes.

Example MetalLB-style annotation override (values depend on your cluster):

```
coturn:
  enabled: true
  annotations:
    metallb.universe.tf/address-pool: public
  config:
    port: 3478
    min_port: 49152
    max_port: 49251
    username: <turn-username>
    password: <turn-password>
```

Firewall and network ports

Open inbound access to the Coturn load balancer (or nodes, when using NodePort / Local traffic policy) for:

PORT	PROTOCOL	PURPOSE
3478	TCP	TURN signaling
3478	UDP	TURN/STUN signaling
`min_port`–`max_port` (default 49152–49251)	UDP	Media relay

Firewall, NSG, and security-group changes are your responsibility and are outside what `scopearctl` applies. Opening only 3478 without the relay UDP window is a common cause of failed WebRTC media.

DNS and the advertised host

Clients connect using the hostname Coturn advertises (configured as Coturn's external IP / host). By default that is:

`turn-<subdomainSuffix>.<primary hostname>`

Examples:

- `subdomainSuffix: qa0`, hostname `example.com` → `turn-qa0.example.com`
- No suffix, hostname `example.com` → `turn.example.com`

Override with `coturn.config.external_host` when you need a different public name.

Important: DNS and the advertised host must agree. If DNS points somewhere else, or the advertised name does not resolve to the Coturn LB/NodePort path, TURN allocations fail even when the Service and pod look healthy.

Credentials

MODE	BEHAVIOR
`auto_secret_gen: true` (default)	ScopeAR creates Secret `scopear-turn` from `coturn.config.username` and `coturn.config.password`
`auto_secret_gen: false`	You must create Secret `scopear-turn` yourself **before** apply, with keys `username` and `password`

Example when managing the secret yourself:

```
kubectl create secret generic scopear-turn \
  --from-literal=username=<turn-username> \
  --from-literal=password=<turn-password> \
  -n <scopear-namespace>
```

Use unique credentials for each production environment. Do not reuse sample or shared demo passwords long-term.

CMS and Coturn both use `scopear-turn` so clients receive credentials that match the TURN server.

Verify the installation

After apply:

1. Confirm the Coturn Service exists and has an EXTERNAL-IP / hostname (LoadBalancer) or allocated NodePorts.
2. Confirm the Coturn pod is Running.
3. Confirm Secret `scopear-turn` exists in the ScopeAR namespace.
4. Confirm CMS TURN advertise settings match the DNS name clients will use (host, port 3478, protocol as configured).
5. From outside the cluster, verify reachability of:
   - TCP and UDP **3478** on the LB (or NodePort mapping)
   - At least one UDP port in the relay range (for example 49152)
6. Run a ScopeAR / WebRTC connectivity check with the TURN credentials (your ScopeAR representative can provide diagnostic guidance for your client apps).

Useful commands (adjust namespace as needed):

```
kubectl get svc scopear-coturn -n <scopear-namespace>
kubectl get pods -l app.kubernetes.io/name=scopear-coturn -n <scopear-namespace>
kubectl get secret scopear-turn -n <scopear-namespace>
```

Common problems

SYMPTOM / MISTAKE	WHAT TO CHECK
LB healthy but calls fail	DNS does not match `external_host` / advertised TURN hostname
Signaling works, media fails	Relay UDP range not open end-to-end (NSG/SG/firewall)
OpenShift Route for Coturn	Routes cannot carry TURN/UDP; use LoadBalancer, MetalLB, or NodePort + external LB
Manifest validation error on ports	Relay span greater than 200, or `max_port` less than `min_port`
Still using shared ScopeAR proxy	Custom `cms.config.coturn.turn_external_host` left set; clear it or set it to your in-cluster hostname if you want local Coturn
Missing credentials	`auto_secret_gen: false` without a pre-created `scopear-turn` Secret

Contact

For installation issues or environment-specific samples, contact Scope AR support at `support@scopear.com`. Your ScopeAR representative can also provide a tailored manifest for AKS, EKS, OpenShift, or on-premises Kubernetes.

Samples

The following sample manifests are also included as standalone YAML files under `samples/` in the documentation package.

`scopear_k8s_onprem_sample.yaml`

```
# Generated by hack/generate-docs-samples — do not edit by hand.
# Kitchen-sink example with SetDefaults() applied. Delete or change values as needed.
# Kubernetes on-premises ScopeAR environment (standalone cluster, on-prem: true).
apiVersion: k8s.scopear.com/v1
kind: ScopeAR
metadata:
  name: scopear-onprem
  namespace: scopear
spec:
  ai:
    config:
      airbrake:
        enabled: false
      helicone_api_url: https://oai.hconeai.com/v1
      istio_injection: false
      mongodb:
        host: mongodb
        port: 27017
      qdrant_url: http://qdrant:6333
      storage:
        minio_server_address: minio
        minio_server_port: 443
        minio_server_use_ssl: true
        provider: aws
      true_openai_queries: false
    enabled: false
    ingress: {}
    replicas: 1
    resources:
      limits:
        cpu: 500m
        memory: 1000Mi
      requests:
        cpu: 200m
        memory: 500Mi
  apply_ingress: false
  apply_virtual_service: true
  c2:
    api:
      airbrake:
        enabled: false
      replicas: 1
      resources:
        limits:
          cpu: 500m
          memory: 3000Mi
        requests:
          cpu: 200m
```

```
        memory: 3000Mi
    client:
      airbrake:
        enabled: false
      analytics: false
      replicas: 1
      resources:
        limits:
          cpu: "1"
          memory: 1000Mi
        requests:
          cpu: 100m
          memory: 100Mi
    config:
      cad_pipeline:
        flex_host: flex-op.scopear.com
        flex_port: 27001
        processing_mode: none
        query_time: 10000
      istio_injection: false
      job_query_time: 100
      max_upload_file_size: 2000
      mongodb:
        host: mongodb
        port: 27017
      slidemaster_enabled: true
      storage:
        aws_bucket_region: us-west-2
        minio_client_port: 443
        minio_private_address: minio
        minio_private_port: 9000
        minio_server_use_ssl: true
        storage_provider: aws
      userflow_enabled: false
    enabled: true
    ingress: {}
    model_converter:
      airbrake:
        enabled: false
      replicas: 1
      resources:
        limits:
          cpu: "2"
          memory: 4000Mi
        requests:
          cpu: 200m
          memory: 4000Mi
    project_packager:
```

```
      airbrake:
        enabled: false
      replicas: 1
      resources:
        limits:
          cpu: "2"
          memory: 5000Mi
        requests:
          cpu: 500m
          memory: 2000Mi
    webapp:
      airbrake: {}
      ingress: {}
      replicas: 1
      resources:
        limits:
          cpu: "1"
          memory: 1000Mi
        requests:
          cpu: 100m
          memory: 100Mi
  channel: stable
  cluster:
    provider: other
    type: standalone
  cms:
    api:
      replicas: 1
      resources:
        limits:
          cpu: "4"
          memory: 4Gi
        requests:
          cpu: "1"
          memory: 4Gi
    config:
      admin:
        company_name: Scope AR
        email_address: support@scopear.com
        first_name: ScopeAR
        last_name: Support
        phone_number: 780-555-1234
        support_admin_email_address: support@scopear.com
      airbrake:
        enabled: false
      auth_method: db
      coturn:
        enable_multi_region: false
```

```
    enabled: false
    label: us-east-1
    tunnel_external_host: proxy-aws-us.scopear.com
    tunnel_external_port: 443
    turn_external_host: proxy-aws-us.scopear.com
    turn_external_port: 3478
    turn_protocol: udp
  hubspot: {}
  istio_injection: false
  jwt_token_verification: true
  ldap:
    enabled: false
  log_level: info
  metabase:
    calling_dashboard_id: 206
    company_dashboard_id: 330
    enabled: false
    scenario_dashboard_id: 204
  mysql:
    aws_iam_auth_type: instance_profile_credentials
    aws_rds_region: us-west-2
    database: scopear_cms
    host: scopear-mysql.mysql.svc.cluster.local
    port: 3306
    use_iam_auth: false
    username: scopecms
  open_id:
    enabled: false
  puma_timeout: 120
  redis:
    host: scopear-redis.redis.svc.cluster.local
    port: 6379
  sentry:
    enabled: false
  smtp:
    auth_type: plain
    email_format: all
    enable_starttls_auto: true
    openssl_verify_mode: none
  storage:
    default_bucket: scopear-data
    default_region: us-west-2
    enable_multi_region: false
    file_storage: s3
    storage_size: 100Gi
    storage_ssl_verify: true
  twilio: {}
  userflow_enabled: false
```

```
      virus_scan_enabled: false
      zeus:
        enabled: false
        external_host: master.scopear.com
        external_port: 24000
    ingress: {}
    sidekiq:
      replicas: 1
      resources: {}
  coturn:
    auto_secret_gen: true
    config:
      max_port: 49251
      min_port: 49152
      password: Augmenttheworld2025!
      port: 3478
      protocol: udp
      realm: ScopeAR
      username: scopear
    enabled: false
    replicas: 1
    resources:
      limits:
        cpu: "1"
        memory: 512Mi
      requests:
        cpu: 100m
        memory: 128Mi
  dt:
    enabled: false
    ingress: {}
    istio_injection: false
    replicas: 1
    resources: {}
  env: production
  gwl:
    host: getworklink.com
    ingress: {}
    istio_injection: false
    replicas: 1
    resources: {}
    use_formatted_host: true
  hostnames:
    - scopear.example.com
  iot:
    config:
      istio_injection: false
      mongodb: {}
```

```
      mqtt:
        host: scopear-iot-mosquitto-headless
        port: 30883
        protocol: mqtts
      simulator:
        enabled: false
        ingress: {}
        port: 3001
    enabled: false
    ingress: {}
    port: 9000
    resources: {}
    ssl_enabled: true
  istioAPIVersion: networking.istio.io/v1
  istioGatewayName: istio-system/scopear-gateway
  minio:
    enabled: false
    ingress: {}
    istio_injection: false
    resources: {}
    storage_size: 100Gi
  mongodb:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 100Gi
  mysql:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 100Gi
  on-prem: true
  postgres:
    database: superset
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 25Gi
    username: scopear
  qdrant:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_class_name: default
```

```
    storage_size: 5Gi
  redis:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 10Gi
  stripPublicRegistry: false
  superset:
    config:
      istio_injection: false
      mcp_server_enabled: false
      redis_celery_db: 1
      redis_results_db: 1
    enabled: false
    ingress: {}
    replicas: 1
    resources: {}
  xmpp:
    enabled: false
    ingress: {}
    istio_injection: false
    replicas: 1
    resources: {}
  zeus:
    auto_secret_gen: true
    config:
      encryption_cert: scopear-zeus-encryption
      istio_injection: false
      port: 24000
      redis:
        database: 10
        host: scopear-redis.redis.svc.cluster.local
        port: 6379
        use_auth: false
    enabled: false
    replicas: 1
    resources:
      limits:
        cpu: "2"
        memory: 4Gi
      requests:
        cpu: 500m
        memory: 1Gi
```

`scopear_k8s_sample.yaml`

```
# Generated by hack/generate-docs-samples — do not edit by hand.
# Kitchen-sink example with SetDefaults() applied. Delete or change values as needed.
# Generic Kubernetes ScopeAR environment with full defaulted component tree.
apiVersion: k8s.scopear.com/v1
kind: ScopeAR
metadata:
  name: scopear
  namespace: scopear
spec:
  ai:
    config:
      airbrake:
        enabled: false
      helicone_api_url: https://oai.hconeai.com/v1
      istio_injection: false
      mongodb:
        host: mongodb
        port: 27017
      qdrant_url: http://qdrant:6333
      storage:
        minio_server_address: minio
        minio_server_port: 443
        minio_server_use_ssl: true
        provider: aws
      true_openai_queries: false
    enabled: false
    ingress: {}
    replicas: 1
    resources:
      limits:
        cpu: 500m
        memory: 1000Mi
      requests:
        cpu: 200m
        memory: 500Mi
  apply_ingress: false
  apply_virtual_service: true
  c2:
    api:
      airbrake:
        enabled: false
      replicas: 1
      resources:
        limits:
          cpu: 500m
          memory: 3000Mi
        requests:
          cpu: 200m
```

```
        memory: 3000Mi
    client:
      airbrake:
        enabled: false
      analytics: false
      replicas: 1
      resources:
        limits:
          cpu: "1"
          memory: 1000Mi
        requests:
          cpu: 100m
          memory: 100Mi
    config:
      cad_pipeline:
        flex_host: flex-op.scopear.com
        flex_port: 27001
        processing_mode: none
        query_time: 10000
      istio_injection: false
      job_query_time: 100
      max_upload_file_size: 2000
      mongodb:
        host: mongodb
        port: 27017
      slidemaster_enabled: true
      storage:
        aws_bucket_region: us-west-2
        minio_client_port: 443
        minio_private_address: minio
        minio_private_port: 9000
        minio_server_use_ssl: true
        storage_provider: aws
      userflow_enabled: false
    enabled: true
    ingress: {}
    model_converter:
      airbrake:
        enabled: false
      replicas: 1
      resources:
        limits:
          cpu: "2"
          memory: 4000Mi
        requests:
          cpu: 200m
          memory: 4000Mi
    project_packager:
```

```
      airbrake:
        enabled: false
      replicas: 1
      resources:
        limits:
          cpu: "2"
          memory: 5000Mi
        requests:
          cpu: 500m
          memory: 2000Mi
    webapp:
      airbrake: {}
      ingress: {}
      replicas: 1
      resources:
        limits:
          cpu: "1"
          memory: 1000Mi
        requests:
          cpu: 100m
          memory: 100Mi
  channel: stable
  cluster:
    provider: other
    type: standalone
  cms:
    api:
      replicas: 1
      resources:
        limits:
          cpu: "4"
          memory: 4Gi
        requests:
          cpu: "1"
          memory: 4Gi
    config:
      admin:
        company_name: Scope AR
        email_address: support@scopear.com
        first_name: ScopeAR
        last_name: Support
        phone_number: 780-555-1234
        support_admin_email_address: support@scopear.com
      airbrake:
        enabled: false
      auth_method: db
      coturn:
        enable_multi_region: false
```

```
    enabled: false
    label: us-east-1
    tunnel_external_host: proxy-aws-us.scopear.com
    tunnel_external_port: 443
    turn_external_host: proxy-aws-us.scopear.com
    turn_external_port: 3478
    turn_protocol: udp
  hubspot: {}
  istio_injection: false
  jwt_token_verification: true
  ldap:
    enabled: false
  log_level: info
  metabase:
    calling_dashboard_id: 206
    company_dashboard_id: 330
    enabled: false
    scenario_dashboard_id: 204
  mysql:
    aws_iam_auth_type: instance_profile_credentials
    aws_rds_region: us-west-2
    database: scopear_cms
    host: scopear-mysql.mysql.svc.cluster.local
    port: 3306
    use_iam_auth: false
    username: scopecms
  open_id:
    enabled: false
  puma_timeout: 120
  redis:
    host: scopear-redis.redis.svc.cluster.local
    port: 6379
  sentry:
    enabled: false
  smtp:
    auth_type: plain
    email_format: all
    enable_starttls_auto: true
    openssl_verify_mode: none
  storage:
    default_bucket: scopear-data
    default_region: us-west-2
    enable_multi_region: false
    file_storage: s3
    storage_size: 100Gi
    storage_ssl_verify: true
  twilio: {}
  userflow_enabled: false
```

```
      virus_scan_enabled: false
      zeus:
        enabled: false
        external_host: master.scopear.com
        external_port: 24000
    ingress: {}
    sidekiq:
      replicas: 1
      resources: {}
  coturn:
    auto_secret_gen: true
    config:
      max_port: 49251
      min_port: 49152
      password: Augmenttheworld2025!
      port: 3478
      protocol: udp
      realm: ScopeAR
      username: scopear
    enabled: false
    replicas: 1
    resources:
      limits:
        cpu: "1"
        memory: 512Mi
      requests:
        cpu: 100m
        memory: 128Mi
  dt:
    enabled: false
    ingress: {}
    istio_injection: false
    replicas: 1
    resources: {}
  env: production
  gwl:
    host: getworklink.com
    ingress: {}
    istio_injection: false
    replicas: 1
    resources: {}
    use_formatted_host: true
  hostnames:
    - scopear.example.com
  iot:
    config:
      istio_injection: false
      mongodb: {}
```

```
      mqtt:
        host: scopear-iot-mosquitto-headless
        port: 30883
        protocol: mqtts
      simulator:
        enabled: false
        ingress: {}
        port: 3001
    enabled: false
    ingress: {}
    port: 9000
    resources: {}
    ssl_enabled: true
  istioAPIVersion: networking.istio.io/v1
  istioGatewayName: istio-system/scopear-gateway
  minio:
    enabled: false
    ingress: {}
    istio_injection: false
    resources: {}
    storage_size: 100Gi
  mongodb:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 100Gi
  mysql:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 100Gi
  on-prem: true
  postgres:
    database: superset
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 25Gi
    username: scopear
  qdrant:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_class_name: default
```

```
    storage_size: 5Gi
  redis:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 10Gi
  stripPublicRegistry: false
  superset:
    config:
      istio_injection: false
      mcp_server_enabled: false
      redis_celery_db: 1
      redis_results_db: 1
    enabled: false
    ingress: {}
    replicas: 1
    resources: {}
  xmpp:
    enabled: false
    ingress: {}
    istio_injection: false
    replicas: 1
    resources: {}
  zeus:
    auto_secret_gen: true
    config:
      encryption_cert: scopear-zeus-encryption
      istio_injection: false
      port: 24000
      redis:
        database: 10
        host: scopear-redis.redis.svc.cluster.local
        port: 6379
        use_auth: false
    enabled: false
    replicas: 1
    resources:
      limits:
        cpu: "2"
        memory: 4Gi
      requests:
        cpu: 500m
        memory: 1Gi
```

`scopear_k8s_eks_sample.yaml`

```
# Generated by hack/generate-docs-samples — do not edit by hand.
# Kitchen-sink example with SetDefaults() applied. Delete or change values as needed.
# Amazon EKS ScopeAR environment with eksRoleArn placeholders.
apiVersion: k8s.scopear.com/v1
kind: ScopeAR
metadata:
  name: scopear-eks
  namespace: scopear
spec:
  ai:
    config:
      airbrake:
        enabled: false
      helicone_api_url: https://oai.hconeai.com/v1
      istio_injection: false
      mongodb:
        host: mongodb
        port: 27017
      qdrant_url: http://qdrant:6333
      storage:
        minio_server_address: minio
        minio_server_port: 443
        minio_server_use_ssl: true
        provider: aws
      true_openai_queries: false
    enabled: false
    ingress: {}
    replicas: 1
    resources:
      limits:
        cpu: 500m
        memory: 1000Mi
      requests:
        cpu: 200m
        memory: 500Mi
  apply_ingress: false
  apply_virtual_service: true
  c2:
    api:
      airbrake:
        enabled: false
      replicas: 1
      resources:
        limits:
          cpu: 500m
          memory: 3000Mi
        requests:
          cpu: 200m
```

```
        memory: 3000Mi
    client:
      airbrake:
        enabled: false
      analytics: false
      replicas: 1
      resources:
        limits:
          cpu: "1"
          memory: 1000Mi
        requests:
          cpu: 100m
          memory: 100Mi
    config:
      cad_pipeline:
        flex_host: flex-op.scopear.com
        flex_port: 27001
        processing_mode: none
        query_time: 10000
      istio_injection: false
      job_query_time: 100
      max_upload_file_size: 2000
      mongodb:
        host: mongodb
        port: 27017
      slidemaster_enabled: true
      storage:
        aws_bucket_region: us-west-2
        minio_client_port: 443
        minio_private_address: minio
        minio_private_port: 9000
        minio_server_use_ssl: true
        storage_provider: aws
      userflow_enabled: false
    eksRoleArn: <aws-role-arn>
    enabled: true
    ingress: {}
    model_converter:
      airbrake:
        enabled: false
      replicas: 1
      resources:
        limits:
          cpu: "2"
          memory: 4000Mi
        requests:
          cpu: 200m
          memory: 4000Mi
```

```
    project_packager:
      airbrake:
        enabled: false
      replicas: 1
      resources:
        limits:
          cpu: "2"
          memory: 5000Mi
        requests:
          cpu: 500m
          memory: 2000Mi
    webapp:
      airbrake: {}
      ingress: {}
      replicas: 1
      resources:
        limits:
          cpu: "1"
          memory: 1000Mi
        requests:
          cpu: 100m
          memory: 100Mi
  channel: stable
  cluster:
    provider: aws
    type: eks
  cms:
    api:
      replicas: 1
      resources:
        limits:
          cpu: "4"
          memory: 4Gi
        requests:
          cpu: "1"
          memory: 4Gi
    config:
      admin:
        company_name: Scope AR
        email_address: support@scopear.com
        first_name: ScopeAR
        last_name: Support
        phone_number: 780-555-1234
        support_admin_email_address: support@scopear.com
      airbrake:
        enabled: false
      auth_method: db
      coturn:
```

```
    enable_multi_region: false
    enabled: false
    label: us-east-1
    tunnel_external_host: proxy-aws-us.scopear.com
    tunnel_external_port: 443
    turn_external_host: proxy-aws-us.scopear.com
    turn_external_port: 3478
    turn_protocol: udp
  hubspot: {}
  istio_injection: false
  jwt_token_verification: true
  ldap:
    enabled: false
  log_level: info
  metabase:
    calling_dashboard_id: 206
    company_dashboard_id: 330
    enabled: false
    scenario_dashboard_id: 204
  mysql:
    aws_iam_auth_type: instance_profile_credentials
    aws_rds_region: us-west-2
    database: scopear_cms
    host: scopear-mysql.mysql.svc.cluster.local
    port: 3306
    use_iam_auth: false
    username: scopecms
  open_id:
    enabled: false
  puma_timeout: 120
  redis:
    host: scopear-redis.redis.svc.cluster.local
    port: 6379
  sentry:
    enabled: false
  smtp:
    auth_type: plain
    email_format: all
    enable_starttls_auto: true
    openssl_verify_mode: none
  storage:
    default_bucket: scopear-data
    default_region: us-west-2
    enable_multi_region: false
    file_storage: s3
    storage_size: 100Gi
    storage_ssl_verify: true
  twilio: {}
```

```
      userflow_enabled: false
      virus_scan_enabled: false
      zeus:
        enabled: false
        external_host: master.scopear.com
        external_port: 24000
    eksRoleArn: <aws-role-arn>
    ingress: {}
    sidekiq:
      replicas: 1
      resources: {}
  coturn:
    auto_secret_gen: true
    config:
      max_port: 49251
      min_port: 49152
      password: Augmenttheworld2025!
      port: 3478
      protocol: udp
      realm: ScopeAR
      username: scopear
    enabled: false
    replicas: 1
    resources:
      limits:
        cpu: "1"
        memory: 512Mi
      requests:
        cpu: 100m
        memory: 128Mi
  dt:
    enabled: false
    ingress: {}
    istio_injection: false
    replicas: 1
    resources: {}
  env: production
  gwl:
    host: getworklink.com
    ingress: {}
    istio_injection: false
    replicas: 1
    resources: {}
    use_formatted_host: true
  hostnames:
    - scopear.example.com
  iot:
    config:
```

```
      istio_injection: false
      mongodb: {}
      mqtt:
        host: scopear-iot-mosquitto-headless
        port: 30883
        protocol: mqtts
      simulator:
        enabled: false
        ingress: {}
        port: 3001
    enabled: false
    ingress: {}
    port: 9000
    resources: {}
    ssl_enabled: true
  istioAPIVersion: networking.istio.io/v1
  istioGatewayName: istio-system/scopear-gateway
  minio:
    enabled: false
    ingress: {}
    istio_injection: false
    resources: {}
    storage_size: 100Gi
  mongodb:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 100Gi
  mysql:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 100Gi
  on-prem: true
  postgres:
    database: superset
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 25Gi
    username: scopear
  qdrant:
    enabled: false
    istio_injection: false
    replicas: 1
```

```
    resources: {}
    storage_class_name: default
    storage_size: 5Gi
  redis:
    enabled: false
    istio_injection: false
    replicas: 1
    resources: {}
    storage_size: 10Gi
  stripPublicRegistry: false
  superset:
    config:
      istio_injection: false
      mcp_server_enabled: false
      redis_celery_db: 1
      redis_results_db: 1
    enabled: false
    ingress: {}
    replicas: 1
    resources: {}
  xmpp:
    enabled: false
    ingress: {}
    istio_injection: false
    replicas: 1
    resources: {}
  zeus:
    auto_secret_gen: true
    config:
      encryption_cert: scopear-zeus-encryption
      istio_injection: false
      port: 24000
      redis:
        database: 10
        host: scopear-redis.redis.svc.cluster.local
        port: 6379
        use_auth: false
    enabled: false
    replicas: 1
    resources:
      limits:
        cpu: "2"
        memory: 4Gi
      requests:
        cpu: 500m
        memory: 1Gi
```