SCOPE

ON-PREM DOCUMENTATION

ScopeAR WorkLink Create OpenShift Installation

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 WorkLink Create Openshift Installation

Overview

Setting Up Prerequisites

- Preparing for installation
- Loading Images into the Openshift Registry
- Authorize with CMS
- MongoDB
  - Helm Chart
- Object Storage
  - AWS

Configuring Scope AR WorkLink Create

- Rendering Kubernetes Manifests

Generating Kustomization

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

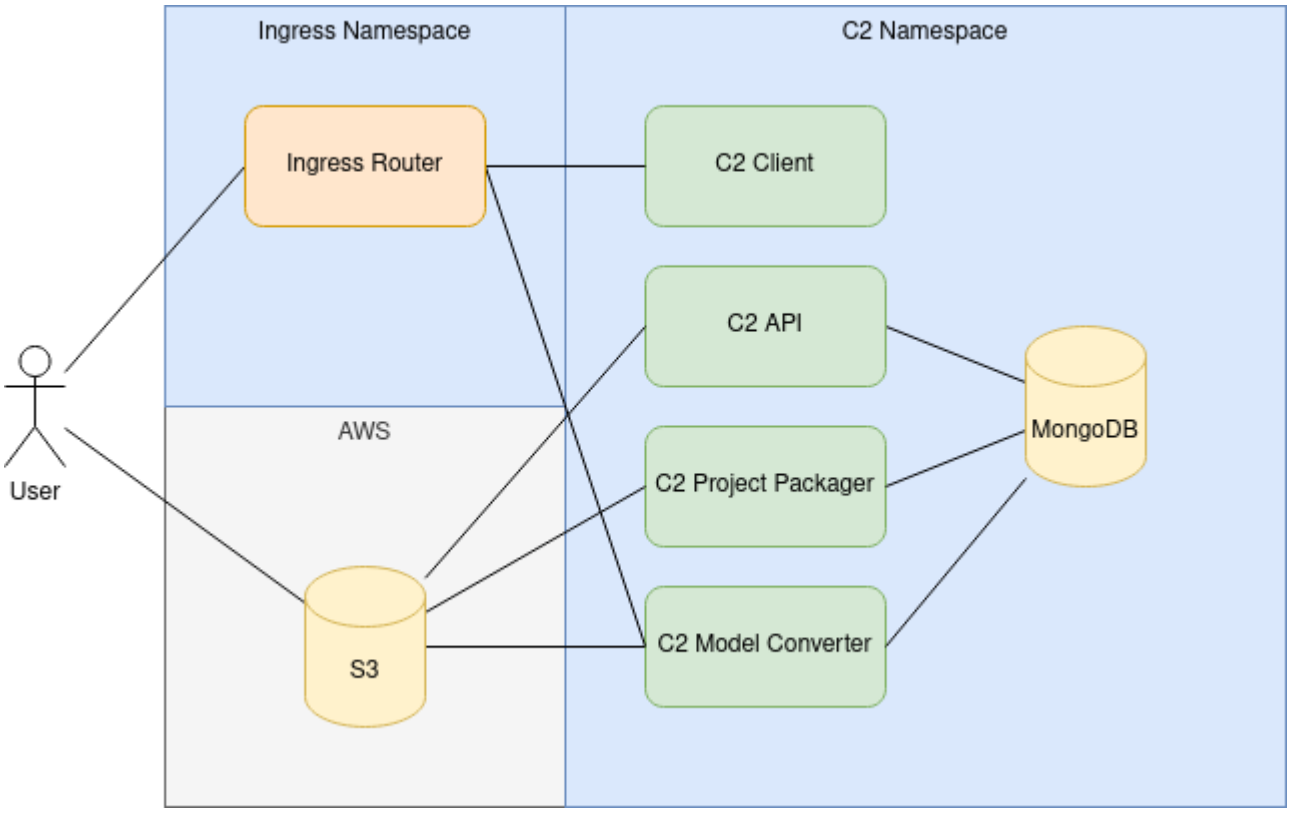
ScopeAR WorkLink Create Openshift Installation

Overview

Scope AR WorkLink Create consists of 4 containers:

- WorkLink Create Client (Single Page Application asset server)
- WorkLink Create API (Create back-end API)
- WorkLink Create Project Packager (facilitates project import/export)
- WorkLink Create Model Converter (facilitates conversion of CAD models)

The API, Project Packager and Model Converter containers communicate with a MongoDB instance and an S3-compatible object store to persist state.



Network Architecture

Setting Up Prerequisites

Preparing for installation

The following environment variables should be set in your shell before proceeding:

```
NAMESPACE=<namespace>
REGISTRY=<openshift_registry_url>
REPO=$REGISTRY/$NAMESPACE
```

Loading Images into the Openshift Registry

Login to Docker Hub:

`docker login docker.io`

Note: You must log into Docker hub with an account that is authorized to pull ScopeAR images. If you do not have access to such an account, please contact Scope AR support.

Login to OpenShift cluster:

`oc login -u <username> -p <password>`

Login to Image Registry:

`docker login <registry_url> -u <username> -p $(oc whoami -t)`

Pull Scope AR WorkLink Create images:

```
docker pull docker.io/scopear/docker-wlc2-api:openshift
docker pull docker.io/scopear/docker-wlc2-client:openshift
docker pull docker.io/scopear/docker-wlc2-mc:openshift
docker pull docker.io/scopear/docker-wlc2-package-processor:openshift
```

Re-tag images for OpenShift internal registry:

```
docker tag docker.io/scopear/docker-wlc2-api:openshift $REPO/api:latest
docker tag docker.io/scopear/docker-wlc2-client:openshift $REPO/client:latest
docker tag docker.io/scopear/docker-wlc2-mc:openshift $REPO/mc:latest
docker tag docker.io/scopear/docker-wlc2-package-processor:openshift $REPO/package-
processor:latest
```

Push Scope AR WorkLink Create images to the OpenShift internal registry:

```
docker push $REPO/api:latest
docker push $REPO/client:latest
docker push $REPO/mc:latest
docker push $REPO/package-processor:latest
```

Authorize with CMS

WorkLink Create requires the Scope AR CMS JWT Secret Token to authenticate itself with the CMS.

Once you've obtained the CMS JWT Secret Token, you must add it as a Kubernetes secret to allow it to be picked up by the WorkLink Create API container:

```
kubectl create secret generic -n <namespace> scopear-c2 \
        --from-literal=cms_jwt_secret=<<REDACTED>>
```

Note: Replace `<<REDACTED>>` with the your CMS instance's secret token.

MongoDB

The WorkLink Create API server requires a mongodb-compatible database as it's primary data store.

Helm Chart

The simplest way to obtain one is to install the bitnami mongodb helm chart:

```
helm upgrade --install \
        --namespace <namespace> \
        scopear-c2-mongodb bitnami/mongodb \
        --set auth.username=scopear-c2 \
        --set auth.database=scopear-c2 \
        --set podSecurityContext.enabled=false \
        --set containerSecurityContext.enabled=false
```

Note: `podSecurityContext` and `containerSecurityContext` are disabled as they can cause issues with Openshift's default security policies.

Object Storage

WorkLink Create uses the S3 protocol for object storage.

AWS

1. Create S3 bucket

2. Create IAM policy with full access to the newly created S3 bucket
3. Create IAM user bound to newly created IAM policy
4. Obtain programmatic access keys (`AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY`)
5. Create kubernetes secret for WorkLink Create S3 authentication:

```
kubectl create secret -n <namespace> generic \
        scopear-c2-aws-credentials \
        --from-literal=AWS_ACCESS_KEY_ID=<<REDACTED>> \
        --from-literal=AWS_SECRET_ACCESS_KEY=<<REDACTED>>
```

Note: Replace `<<REDACTED>>` with the values obtained in step 4.

Finally, you'll need to update the CORS configuration of the newly created S3 bucket. The CORS configuration should look like:

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "HEAD",
            "GET",
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "null",
            "https://<<WorkLink Create Client Hostname>>"
        ],
        "ExposeHeaders": [
            "x-amz-server-side-encryption",
            "x-amz-request-id",
            "x-amz-id-2"
        ],
        "MaxAgeSeconds": 3000
    }
]
```

Note: Replace `<<C2 Client Hostname>>` with the hostname of the WorkLink Create client.

Configuring Scope AR WorkLink Create

The following example can be used as a reference to construct your WorkLink Create config yaml:

```
apiVersion: k8s.scopear.com/v1alpha1
kind: C2
metadata:
  name: scopear-c2
  namespace: c2
spec:
  aws:
    region: us-east-1
  api:
    host: c2-api.example.com
    route:
      tls:
        termination: edge
  client:
    host: c2.example.com
    route:
      tls:
        termination: edge
  cms:
    host: cms.example.scopear.com
  pixyz:
    local:
      flex:
        host: pixyz-flex.example.com
        port: 27000
        tls: true
  storage:
    provider: AWS
    bucket: scopear-c2
  kustomization:
    images:
    - name: docker.io/scopear/docker-wlc2-api
      newName: image-registry.openshift-image-registry.svc:5000/c2/api
    - name: docker.io/scopear/docker-wlc2-client
      newName: image-registry.openshift-image-registry.svc:5000/c2/client
    - name: docker.io/scopear/docker-wlc2-mc
      newName: image-registry.openshift-image-registry.svc:5000/c2/mc
    - name: docker.io/scopear/docker-wlc2-project-packager
      newName: image-registry.openshift-image-registry.svc:5000/c2/project-packager
```

Note: If you do not have a flex license server with a valid Pixyz license, you may enter bogus values for these fields. All WorkLink Create functionality unrelated to CAD file conversion should continue to work

as normal. If a Pixyz license is desired for the CAD conversion functionality, please contact Scope AR sales/support for assistance.

Rendering Kubernetes Manifests

You can render the kubernetes manifests for the given WorkLink Create config by invoking:

```
./scopearctl c2 manifests -f <filename>
```

Or apply them directly to your cluster:

```
./scopearctl c2 manifests -f <filename> | kubectl apply -f -
```

Generating Kustomization

If further configuration of the resources is required, you can output a kustomization instead of outputting the Kubernetes resource manifests.

To generate the kustomization, you'll need to use the `scopearctl` command line tool:

```
./scopearctl c2 kustomization -f <filename>
```

You may then make any desired modifications to the `overlay` directory, and use the `kustomize` CLI to build the Kubernetes manifests:

```
kustomize build overlay
```

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
```