SCOPE

ON-PREM DOCUMENTATION

scopearctl User Guide

Version	3.99.1
Channel	on-prem
Mode	production
Revision	49f2343
Generated	2026-10-20

Confidential · For authorized Scope AR customers and partners

Contents

Scopearctl User Guide: Simplifying ScopeAR Configuration Management

- Introduction
  - Purpose of the Document
  - Overview of Scopearctl
- Scopearctl at a Glance
  - Key Functionality
  - Benefits
- Installation Guide
  - Prerequisites
  - Installation via Script
- Using Scopearctl
  - Configuration Files
  - Deployment Process
  - Handling Multiple Environments
- Advanced Features and Customizations
  - Custom Scripts and Extensions
  - Security and Compliance
- Troubleshooting and Support
  - Common Issues and Resolutions
  - Contact
- Samples
  - `scopear_k8s_onprem_sample.yaml`
  - `scopear_k8s_sample.yaml`
  - `scopear_k8s_eks_sample.yaml`
  - `scopear_swarm_env_sample.yaml`

Scopearctl User Guide: Simplifying ScopeAR Configuration Management

Introduction

Purpose of the Document

This document provides a comprehensive guide on how to install, use, and understand the purpose and benefits of “scopearctl,” a tool designed to streamline configuration management across Kubernetes assets in Scopear environments.

Overview of Scopearctl

Scopearctl is a command-line tool that simplifies the management of configurations across multiple deployments within Scopear Kubernetes (K8s) environments.

Scopearctl at a Glance

Key Functionality

- **Single Configuration File Management:** Centralizes the configuration process into a single file, replacing the need for multiple, disparate configuration files.
- **Programmatic Injection:** Automates the process of injecting configurations into various deployment settings.
- **Logical Manifest Apply:** Deploys the manifest to a K8s cluster in a logical order for maximum dependency success.

Benefits

- Reduces configuration complexity and human error.
- Speeds up deployment and update cycles.
- Improves consistency across environments.

Installation Guide

Prerequisites

- A supported operating system: Linux or macOS
- Network access to `https://release.scopear.com`
- Sufficient permissions to install software (may require `sudo` access)

Installation via Script

Run the Script

The script accepts several optional parameters to customize the installation process:

- `--os=<os>`: Specify the operating system (default: detected automatically).
- `--version=<version>`: Specify the version to install (default: latest).
- `--channel=<channel>`: Specify the release channel (default: stable).
- `--docsdir=<docsdir>`: Specify the directory for documentation (default: current directory).
- `--tmpdir=<tmpdir>`: Specify the temporary directory for downloads (default: `/tmp/scopearctl`).
- `--installdir=<installdir>`: Specify the installation directory (default: `/usr/local/bin`).
- `--nodocs`: Skip downloading and setting up documentation.
- `--nodownload`: Skip downloading the binary and only perform installation steps.

Example Command:

```
curl -sSL https://release.scopear.com/scopearctl/install | sh -s -- --version=1.0.0 --installdir=/opt/scopearctl
```

Install Script Steps

- **Validation:** Checks if the operating system is either Linux or macOS. If not, it exits with an error.
- **Download:** Constructs the download URL based on the specified version and channel, then downloads the archive to the temporary directory.
- **Checksum Verification:** If checksum verification is enabled, the script downloads the checksum file and verifies the integrity of the downloaded archive.
- **Extraction:** The downloaded archive is extracted to the temporary directory.
- **Documentation:** If documentation setup is enabled, the script downloads the documentation files to the specified directory.
- **Installation:** The binary is moved to the specified installation directory, and execution permissions are set.
- **Cleanup:** Temporary files and directories are removed.

Example Install Scenarios

- **Install the Latest Version:**

  ```
  curl -sSL https://release.scopear.com/scopearctl/install | sh
  ```

- **Install a Specific Version:**

```
curl -sSL https://release.scopear.com/scopearctl/install | sh -s -- --version=3.37.2
```

- **Install Without Documentation:**

```
curl -sSL https://release.scopear.com/scopearctl/install | sh -s -- --nodocs
```

- **Install to Custom Directories:**

```
curl -sSL https://release.scopear.com/scopearctl/install | sh -s -- \
--installdir=/opt/scopearctl \
--docsdir=/opt/scopearctl/docs \
--tmpdir=/opt/scopearctl/tmp
```

Using Scopearctl

Configuration Files

- **Structure and Format:** The unified configuration file enables the end user to set unique configurations to almost all available settings ScopeAR has. Using higher-order stanzas, configs are nested under their respective application.
- **Best Practices:** It is recommended to have a separate file per environment and store those files in a version control system such as Git.

Deployment Process

1. **Set Kubeconfig Context:** Ensure that your kubeconfig context is set to the proper cluster.

Running k8s apply command

The command accepts several optional parameters to customize the installation process:

```
scopearctl k8s apply -h

 ▄▄▄▄▄▄  ▄▄▄▄▄  ▄▄▄▄▄  ▄▄▄▄▄  ▄▄▄▄▄▄  ▄▄▄▄  ▄▄▄▄▄
 ▀▀  ▀▀  ▀▀  ▀▀  ▀▀  ▀▀ ▀▀  ▀▀  ▀▀  ▀▀ ▀▀ ▀▀  ▀▀
 ▀▀▀▀▀▀  ▀▀  ▀▀  ▀▀  ▀▀ ▀▀▀▀▀  ▀▀▀▀▀▀ ▀▀▀▀▀
    ▀▀ ▀▀  ▀▀  ▀▀  ▀▀ ▀▀  ▀▀  ▀▀  ▀▀ ▀▀ ▀▀  ▀▀
 ▀▀▀▀▀▀  ▀▀▀▀▀  ▀▀▀▀▀  ▀▀  ▀▀  ▀▀▀▀▀▀ ▀▀  ▀▀ ▀▀  ▀▀

┌─ Build info ──────────────────────────────────────────┐
│                                                       │
│ ┌─ Version ─┐ ┌─ Plugin Revision ─┐ ┌─ Channel ─┐ │
│ │ 3.37.2    │ │ da6326c           │ │ stable    │ │
│ └───────────┘ └───────────────────┘ └───────────┘ │
│                                                       │
└───────────────────────────────────────────────────────┘

Generates & deploys ScopeAR components manifests. Configured by your environment manifest.

Usage:
  scopearctl k8s apply [flags]

Examples:
scopearctl k8s apply -f manifest.yaml

Flags:
      --dry               Generate manifests only.
      --exclude strings   List of components to exclude from apply
  -h, --help              help for apply
  -f, --manifest string   ScopeAR CR file.
      --only strings      List of components to apply; e.g. --only scopear
  -o, --out string        Write the manifests to the given path (default "__gen__")
      --skip-delays       Disabled manifest apply delays

Global Flags:
      --debug                 Set log level to debug
      --kubeconfig string     Path to kubeconfig (default "/Users/sgraham/.kube/config")
  -v, --verbose count         Increase logging verbosity (use multiple times for more
output)
```

Example Command:

```
scopearctl k8s apply -f /path/to/config.yaml
```

Apply Script Steps

- **Defaulting:** The tool sets default values for any configuration item that is left unset in your environment manifest.
- **Validation:** The tool checks if the values set in your ScopeAR environment manifest. If errors exist, it will return them all and exit the process.
- **YAML Manifest Generation:** Scopearctl programmatically injects the single configuration file into multiple K8s assets and outputs them to a local directory, default `./__gen__`.
- **Apply Generated Manifests to Cluster:** If the `--dry` flag is not set and the user has the appropriate cluster context authorization, the tool will apply the generated manifest similar to the way `kubectl apply -f` would.

Example Usage Scenarios

- **Apply the configs set in your ScopeAR manifest:**

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

- **Only generate yaml manifests:**

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

- **Set a custom kubeconfig file:**

```
scopearctl k8s apply \
  -f /path/to/scopear-manifest.yaml \
  --kubeconfig /path/to/custom-config
```

Handling Multiple Environments

- **Environment Specific Configurations:** It is best practice to create a configuration file per environment and use a logical file naming scheme to easily differentiate between configs.
- **Version Control Integration:** Covers integration with version control systems for tracking changes.

Advanced Features and Customizations

Custom Scripts and Extensions

- **Integration with Other Tools and Scripts:** Scopearctl can be easily installed into any CI/CD environment either in custom scripting or by running raw Linux commands to download and install. These installations can be controlled by downloading specific versions and installing into permissioned paths. You can read about the installation above.

Security and Compliance

- **Compliance Checks:** Scopearctl is scanned continuously for vulnerable dependencies and patched regularly.
- **Security Best Practices:** Security vulnerabilities rated “High” or above are patched before releasing any new versions, ensuring you only get the safest builds possible.

Troubleshooting and Support

Common Issues and Resolutions

- **Installation:** Ensure you have `curl`, `shasum`, and `tar` installed. Verify you have sufficient permissions to install to the specified directory (use `sudo` if necessary). Check for network issues if the download fails.
- **Misconfigurations:** These can be common. Simply edit the config file and re-apply the changes. Kubernetes will handle most of the changes by default; however, if it’s just a `ConfigMap` change, the pod that uses the config might need to be restarted in the environment.
- **Apply issues:** Ensure you have the correct expected version of `scopearctl` installed, you can find it by running `scopearctl help`. Verify you have sufficient permissions to apply manifests to the K8s cluster in current context. Check for network issues if the connection fails.

Contact

For any issues or further assistance, please contact the `scopearctl` support team at `devops@scopear.com`.

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:
    enabled: true
    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:
    enabled: true
    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:
    enabled: true
    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_swarm_env_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.
# Docker Swarm on-premises ScopeAR environment (channel: on-prem).
apiVersion: swarm.scopear.com/v1
kind: ScopeAR
metadata:
  name: scopear-swarm
  namespace: scopear
spec:
  ai:
    ai:
      resources: {}
    config:
      airbrake: {}
      helicone_api_url: https://oai.hconeai.com/v1
      mongodb:
        database: ai
        host: mongodb
        port: 27017
      qdrant_url: http://qdrant:6333
      storage:
        minio_server_address: minio.swarm.localdomain
        minio_server_port: 443
        minio_server_use_ssl: true
        provider: minio
      true_openai_queries: false
    data_volume: /opt/scopear/data/ai
    enabled: false
  channel: on-prem
  comms:
    coturn:
      enabled: false
      resources: {}
      turn_max_port: 50000
      turn_min_port: 49152
      turn_password: Augmenttheworld2025!
      turn_port: 3478
      turn_username: scopear
    zeus:
      config:
        redis:
          data_volume: /opt/scopear/data/redis
          resources: {}
      enabled: true
      resources: {}
  gwl:
    android_package_name: com.scopear.RemoteAR
    deferred_deep_linking_firebase_domain: scopear.page.link
```

```
    ios_app_store_id: "1099950714"
    microsoft_store_id: 9nblggh4v339
    replicas: 1
    resources: {}
  hostnames:
    - scopear.example.com
  iot:
    config:
      cert_volume: /opt/scopear/data/iot/certs
      mqtt:
        cert_volume: /opt/scopear/data/mosquitto/certs
        port: 30883
    enabled: false
    resources: {}
  logLevel: info
  on-prem: true
  secrets: {}
  secrets_path: /opt/scopear/secrets
  stripPublicRegistry: true
  superset:
    config:
      cms_db_user: superset
      mcp_server_enabled: false
      redis_celery_db: 3
      redis_results_db: 2
    enabled: false
    postgres:
      data_volume: /opt/scopear/data/postgres
      resources: {}
    resources: {}
  traefik:
    data_volume: /opt/scopear/data/traefik
    tls:
      acme_storage: /etc/traefik/acme.json
      provider: static
  web:
    cms:
      data_volume: /opt/scopear/data/cms
      replicas: 1
      resources: {}
    config:
      admin: {}
      airbrake: {}
      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: {}
      ldap: {}
      log_level: info
      metabase: {}
      mysql: {}
      open_id: {}
      redis: {}
      sentry: {}
      smtp:
        auth_type: plain
        email_format: all
        enable_starttls_auto: true
        openssl_verify_mode: none
      storage:
        file_storage: s3
      twilio: {}
      zeus:
        enabled: false
        external_host: master.scopear.com
        external_port: 24000
    minio:
      console_enabled: true
      data_volume: /opt/scopear/data/minio
      resources: {}
    mysql:
      data_volume: /opt/scopear/data/mysql
      resources: {}
    nginx:
      resources: {}
    redis:
      data_volume: /opt/scopear/data/redis
      resources: {}
    sidekiq:
      replicas: 1
      resources: {}
  wlc2:
    api:
      replicas: 1
      resources: {}
    client:
      replicas: 1
      resources: {}
    config:
```

```
    cad_pipeline:
      processing_mode: none
      query_time: 10000
    mongodb: {}
    storage:
      bucket_name: wlc2
  model_converter:
    replicas: 1
    resources: {}
  mongodb:
    data_volume: /opt/scopear/data/mongodb
    database: wlc2
    enabled: true
    replicas: null
    resources: {}
  project_packager:
    replicas: 1
    resources: {}
  resources: {}
  webapp:
    replicas: 1
    resources: {}
```