SCOPE

ON-PREM DOCUMENTATION

ScopeAR Kubernetes 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

Technical Document: ScopeAR Kubernetes Installation via `scopearctl`

- Getting started
- ScopeAR CR environment manifest
- Samples
  - `scopear_k8s_onprem_sample.yaml`
  - `scopear_k8s_sample.yaml`
  - `scopear_k8s_eks_sample.yaml`

Technical Document: ScopeAR Kubernetes Installation via `scopearctl`

This document details how to manage ScopeAR components on a Kubernetes cluster using `scopearctl k8s` command.

Getting started

Before beginning the installation, you will need to obtain the following:

- `scopearctl` executable for your operating system and arch e.g. `linux/amd64`. This is the primary deployment method defined in this document.
- A ScopeAR manifest (`apiVersion: k8s.scopear.com/v1`) file edited to your environemnt specifics.
- A K8s cluster that you can access via the Kubernetes API (optional)

ScopeAR CR environment manifest

The ScopeAR manifest API has a comprehensive reference document associated. You should have been provided `k8s_scopear_env_manifest_configs.pdf` along with this document.

Your ScopeAR rep can provide samples for your specifc use case as needed.

Usage Instructions

1. **Run the Script**

   The script 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/scopear-manifest.yaml`

1. **Detailed 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 envirnoment manifest. If error 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
```

Troubleshooting

- Ensure you have the correct expected version of `scopearctl` installed.
- 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`.

By following this guide, you should be able to successfully download and install `scopearctl` using the script from the online address.

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
```