ScienceLogic

Skylar AI Installation Guide

Version 2.8.0

Table of Contents

Skylar AI On-Premises Installation Guide 1

Skylar Analytics Cluster Requirements 2

Skylar Analytics Cluster Requirements 2

Skylar Advisor Cluster Requirements 3

Required Infrastructure Components 3

Optional Cluster Items 3

Registry Access Setup 4

Pre-flight Validation 4

Step 1: Get Harbor CLI Secret 4

Step 2: Sign In to the Helm Registry 5

Step 3: Install the OpenTelemetry Operator 5

Step 4: Download the Skylar AI Chart 6

Step 5: Determine Scaling Profiles 6

Availability Profiles 6

Installation Process 7

Step 1: Analytics Configuration File Setup 7

Ingress Controller Annotations and Known Issues 7

Proxy Configuration 8

Adding Streaming Telemetry 8

Adding Prometheus Monitoring 9

Step 2: Skylar Advisor Configuration (If Applicable) 9

Step 3: Configure the Harbor Password 9

Step 4: Configure the TLS Certificate 10

Step 5: Deploying the Skylar AI Platform 10

Configure the TLS Certificate 11

Validation and Access 11

Verify Deployment Status 11

Check Ingress and User Interface 12

Useful Commands 12

Troubleshooting 12

Registry Authentication Issues 12

Skylar Analytics and Skylar Advisor Sizing Guide **13**
Sizing for Skylar Analytics 14
With High Availability 14
Without High Availability 14
Additional Sizing for Adding Skylar Advisor to Skylar Analytics 14
With High Availability 14
Without High Availability 15
Largest Single Pod Sets Minimum Node Size 15
Data Visualization (DataViz) Sizing 15
Additional Sizing for Adding Streaming Telemetry 16
Troubleshooting and Security for Skylar AI **17**
Troubleshooting the Installation Process 18
Common Issues 18
Image Pull Failures 18
TLS Certificate Issues 18
DNS Resolution Issues 18
Ingress Controller Issues 18
Pod Startup Failures 18
Database Connection Issues 18
Large Upload/Download Issues 19
Cloud-Specific Issues 19
AWS 19
GCP 19
Azure 19
Useful Commands 19
Deployment Status Commands 19
Infrastructure Verification Commands 19
Registry Authentication Issues 20
TLS Certificate Issues 20
Ingress Controller Issues 20
Security Considerations 20
Security Best Practices 20

Third-Party Security Responsibilities 21
Backup and Recovery 21
Critical Data Components 21
ClickHouse Data Volumes 21
PostgreSQL Data Volumes 21
Backup Strategy Options 22
Volume-Level Backups 22
Application-Level Backups 22
Hybrid Approach 22
Backup Verification 22
Regular Testing Requirements 22
Monitoring and Alerting 22
Configuration Backup 23
Helm and Kubernetes Configuration 23
Support 23

Chapter

1

Skylar AI On-Premises Installation Guide

Overview

This chapter details the Kubernetes cluster requirements for Skylar Analytics and Skylar Advisor. It also includes requirements for only Skylar Analytics if you are currently not using Skylar Advisor. The second half of this chapter covers how to install Skylar AI in an on-premises environment.

This chapter covers the following topics:

Skylar Analytics Cluster Requirements 2

Optional Cluster Items 3

Registry Access Setup 4

Installation Process 7

Validation and Access 11

Troubleshooting 12

Skylar Analytics Cluster Requirements

IMPORTANT: Before beginning the installation process, verify that your environment has the **required** versions of Kubernetes and the container runtime (where relevant) listed in the following sections.

Skylar Analytics Cluster Requirements

- **Kubernetes**: Version 1.32 or later (Skylar Analytics only). Any container runtime will work.

  You can run the following command to verify your environment:

  ```
  kubectl version
  ```

  NOTE: Kubernetes upgrades must be performed one minor version at a time. If you are planning a cluster upgrade, see the Kubernetes upgrade documentation.

- **Storage**: Configured EBS StorageClass capable of dynamic PV/PVC provisioning. Storage cannot be NFS-backed, as this is not supported by ScienceLogic databases. The amount of storage varies by environment; for more information see the *Sizing Guide*.
- **Networking**: Currently, only IPv4 is supported. IPv6 is not supported at this time.
- **CPU**: OS must have full x86-64-v2 support. If using Hypervisor, ScienceLogic recommends that you have CPU Host Passthrough enabled to achieve this.
- **External Networking**:
  - The Skylar AI Kubernetes cluster will need Internet access to **https://registry.scilo.tools/**. This connection can be through a proxy if needed (additional configuration required).
  - 443 connectivity from the Skylar One instance to the Skylar AI Kubernetes cluster.
- **Deployment Options**:
  - Self-hosted Kubernetes clusters, such as bare metal or VMware
  - Cloud-managed Kubernetes services (EKS, GKE, AKS)

- **Availability Options**:
    - By default, Skylar Analytics will deploy the Postgres, Clickhouse-Keeper, OpenTelemetry Collector, and Tenant services in a high-availability pattern.
    - For the smallest resource footprint possible, Skylar AI deployments can be configured to use the minimum replicas required for each service.
    - Additionally affinity zone and node scheduling requirements for the Postgres and Tenant services may be relaxed from *hard* to *preferred*. For more information, see *Availability Profiles*.
- **CLI Packages**: Customer will need kubectl access to the cluster along with Helm and Oras packages installed into the command-line interface (CLI).
- **Licensing**: To pull application images, you need an active Skylar AI license. Please work with Sciencelogic to ensure that you are properly licensed for Skylar AI. Also, make sure that you have access to the **skylar** Harbor project before beginning deployment.

Skylar Advisor Cluster Requirements

> **IMPORTANT:** If you are installing only Skylar Analytics, you can skip this topic.

- **Kubernetes**: Version 1.35 or later with a container runtime that supports OCI image volume, such as ContainerD 2.1 or cri-o 1.32 or later runtime.
- **Storage**: Read Write Many (RWX) storage class.
- **GPU driver version**: 580.105.08 with CUDA 13.x

Required Infrastructure Components

Ingress or Gateway API controller: All ingress controllers supported, such as Traefik and nginx-ingress. If using Gateway API, your controller must support listenersets.

DNS and TLS Requirements:

- **FQDN**: A fully qualified domain name pointing to the load balancer or node IPs. FQDN must be resolvable by the cluster nodes DNS servers. Skylar One appliances must be able to resolve the Skylar AI FQDN for telemetry and data streaming functionality.
- **TLS Certificate**: Valid (can be self-signed) TLS certificate for the FQDN, provided as a Kubernetes secret.

Optional Cluster Items

Load Balancing: A load balancer solution to distribute traffic across worker nodes. This can be a physical load balancer like F5 or something virtual like HAProxy with floating IP or MetalLB.

Prometheus stack: Skylar AI components emit Prometheus metrics that can be used to determine application health. The metrics can be exported to an internal Prometheus endpoint then consumed my Skylar One with Prometheus PowerPack.

Backup Solution: Backups of Kubernetes manifests and volumes should be available. This is helpful for Disaster Recovery (DR) situations and for version upgrades.

Registry Access Setup

Pre-flight Validation

Perform all of the following validations before continuing to Step 1:

1. Verify the Kubernetes version; the minimum is version 1.32 (Skylar Analytics only) or 1.3.5 (Skylar Advisor and Skylar Analytics):

   ```
   kubectl version
   ```

2. Verify the container runtime version; ContainerD 2.1 or later is required:

   ```
   containerd --version
   ```

3. For Skylar Advisor, verify that the GPU Operator is running:

   ```
   kubectl get pods -n gpu-operator
   ```

4. For Skylar Advisor, verify that GPUs are visible to Kubernetes:

   ```
   kubectl describe nodes | grep -A5 'nvidia.com/gpu'
   ```

5. Verify that storage classes are available:

   ```
   kubectl get storageclass
   ```

6. Test the registry login:

   ```
   helm registry login registry.scilo.tools
   ```

7. Verify that the ingress controller is running (adjust the namespace for your controller):

   ```
   kubectl get pods -n ingress-nginx
   ```

8. When all of the checks have passed, continue to Step 1, below.

Step 1: Get Harbor CLI Secret

Use the Harbor command line interface (CLI) secret to authenticate to Harbor.

To get the Harbor CLI secret:

1. Go to *https://registry.scilo.tools*.
2. Click **[Login with OIDC Provider]**.
3. On the next page, click **[Customer Login]**.
4. Enter the Salesforce credentials that ScienceLogic provided.
5. After you log in, click your name in the upper right corner and select *User Profile*.
6. In the **User Profile** section, copy the CLI secret. You will use the secret in the following steps.

Step 2: Sign In to the Helm Registry

Use the CLI secret you copied in Step 1 to sign in to the Helm registry.

To sign in to the Helm registry:

1. Run the following command:

```
helm registry login registry.scilo.tools --username <your-harbor-username>
```

Step 3: Install the OpenTelemetry Operator

To install the OpenTelemetry operator:

1. Run the following commands to install the OpenTelemetry operator with a custom image:

```
# Install OpenTelemetry Operator with custom image
helm install opentelemetry-operator
oci://registry.scilo.tools/skylar/opentelemetry-helm-
charts/opentelemetry-operator \
  --version 0.114.0 \
  --namespace opentelemetry-operator \
  --create-namespace \
  --set
"manager.collectorImage.repository=registry.scilo.tools/skylar/sl-
otelcol" \
  --set "manager.collectorImage.tag=0.16" \
  --set admissionWebhooks.certManager.enabled=false \
  --set admissionWebhooks.autoGenerateCert.enabled=true
```

If you want to use cert-manager for certificate management, omit the following lines from the command:

```
--set admissionWebhooks.certManager.enabled=false \
```

```
--set admissionWebhooks.autoGenerateCert.enabled=true
```

Step 4: Download the Skylar AI Chart

Use Helm to pull the Skylar AI chart from the Open Container Initiative (OCI) registry.

To download the Skylar AI chart:

1. Pull the chart. You must be logged in to the registry from Step 2.

   ```
   helm pull oci://registry.scilo.tools/skylar/skylar-charts
   ```

2. Decompress the chart. Replace *x.x.x* with the version you downloaded in the previous step. Use `ls` to find the correct `.tgz` file name.

   ```
   tar -xvf skylar-charts-x.x.x.tgz
   ```

3. Create a directory for your configuration files:

   ```
   mkdir skylar-ai-configs
   ```

Step 5: Determine Scaling Profiles

ScienceLogic sizes deployments based on the Skylar One datapoints-per-minute (DPM) rate, which depends on the number of devices. Work with ScienceLogic to determine the size you need to deploy.

This guide uses the Micro sizing file, which supports up to 5,000 DPM and requires few resources.

To copy the sizing file, run the following command:

```
cp skylar-charts/sizing/analytics-micro.yaml skylar-ai-configs/
```

Availability Profiles

Skylar AI can be configured for high availability (HA). The default deployment requires three nodes so that pods can run on distinct hosts. You can split these nodes across data centers if latency is low.

For a proof of concept (POC), ScienceLogic recommends the `no-ha.yaml` profile to keep resource requirements low.

Profile	Override File	Minimum Nodes	Scheduling
HA (default)	None. This profile is the default.	3	3 replicas on distinct nodes
Soft HA	`soft-ha.yaml`	1	3 replicas, which can run on a single node
No HA	`no-ha.yaml`	1	1 replica

If you choose `soft-ha.yaml` or `no-ha.yaml`, copy the file into the configuration directory. If you use the default profile, you do not need to copy a file.

To copy an availability profile:

1. Run the command for the profile you want to use:

```
cp skylar-charts/availability/no-ha.yaml skylar-ai-configs/
```

```
cp skylar-charts/availability/soft-ha.yaml skylar-ai-configs/
```

Installation Process

Step 1: Analytics Configuration File Setup

Prepare the Helm override files with details specific to your environment. The decompressed `skylar-charts` directory contains an `examples` directory.

If you are deploying Skylar Analytics, copy **examples/override-analytics.yaml** and populate the copy.

To set up the Skylar Analytics override file:

1. Copy the example file:

```
cp skylar-charts/examples/override-analytics.yaml skylar-ai-configs/
```

2. If you want to use the Gateway API instead of ingress, copy the Gateway API example:

```
cp skylar-charts/examples/override-analytics-gateway-api.yaml skylar-ai-configs/
```

3. Edit the override file:

```
vi skylar-ai-configs/override-analytics.yaml
```

Ingress Controller Annotations and Known Issues

The `override-analytics.yaml` file contains ingress-nginx annotations. Annotations differ by controller, so the file includes only the ingress-nginx annotations as an example. Use them as a guide to implement the equivalent annotations for your ingress controller.

Traefik: ScienceLogic has done limited testing with Traefik and found that the nginx ingress translation layer works for this application. ScienceLogic suggests this approach if you use Traefik. For more information, see the Traefik ingress-nginx documentation.

If using middleware, apply this manifest for URL re-writes:

```
apiVersion: traefik.io/v1alpha1
kind: Middleware
metadata:
  name: skylar-auth-sl1embed-rewrite
  namespace: skylar-production
spec:
  replacePath:
```

```
    path: /skylar-auth/sl1embed
---
apiVersion: traefik.io/v1alpha1
kind: Middleware
metadata:
  name: skylar-auth-sl1embed-headers
  namespace: skylar-production
spec:
  headers:
    customResponseHeaders:
      Content-Security-Policy: "frame-ancestors *;"
```

nginx-ingress (F5): nginx-ingress requires a master/minion configuration. If you use nginx-ingress, set `mergeableIngress: true`.

ingress-nginx: The controller must allow snippet annotations. Use the following Helm chart overrides:

```
--set controller.allowSnippetAnnotations=true
```

```
--set controller.config.annotations-risk-level=Critical
```

Proxy Configuration

If you use a proxy for outbound internet traffic, uncomment the sections in `override-analytics.yaml` that set the `HTTP_PROXY`, `HTTPS_PROXY`, and `NO_PROXY` environment variables.

Adding Streaming Telemetry

To enable the Streaming Telemetry product, set the following flags in `override-analytics.yaml`:

```
global:
```

```
  enableOtelUi: true
```

```
  enableStreamingTelemetry: true
```

```
  alertBuilderEnabled: true
```

```
  otel:
```

```
    collector:
```

```
      featureFlag: true
```

Adding Prometheus Monitoring

To enable Prometheus monitoring, set the following flag in `override-analytics.yaml`. This flag exposes Prometheus metrics endpoints on Skylar AI services that your existing Prometheus deployment can scrape.

```
global:

  enableMonitoring: true
```

Step 2: Skylar Advisor Configuration (If Applicable)

If you are also deploying Skylar Advisor, populate `examples/override-advisor.yaml` as well. Skylar Advisor also requires a fourth override file that specifies the GPU-specific tolerations and resources. Work directly with ScienceLogic to create this configuration.

The `sizing` directory contains example files. For example, `sizing/l40s-rtx6000-combined.yaml` and `sizing/h100-resources.yaml` are good starting points.

To set up the Skylar Advisor override files:

1. Copy the example files:

   ```
   cp skylar-charts/examples/override-advisor.yaml skylar-ai-configs/

   cp skylar-charts/sizing/l40s-rtx6000-combined.yaml skylar-ai-configs/
   ```

2. Edit `override-advisor.yaml` with the details for your cluster:

   ```
   vi skylar-ai-configs/override-advisor.yaml
   ```

Step 3: Configure the Harbor Password

Use the commands below to create the secret used for ImagePulls. Concepts like ExternalSecretsOperator and Vault are fully supported. Just configure your provider to create the following secrets instead:

```
kubectl create namespace skylar-production

kubectl -n skylar create secret docker-registry harbor-creds \
  --docker-server=registry.scilo.tools \
  --docker-username='your-email@sciencelogic.com' \
  --docker-password='HarborCLISecret'

kubectl -n skylar create secret generic harbor-creds-mountable \
  --from-literal=HARBOR_USERNAME='your-email@sciencelogic.com' \
```

```
  --from-literal=HARBOR_PASSWORD='HarborCLISecret' \
  --from-literal=HARBOR_REGISTRY=registry.scilo.tools
```

Step 4: Configure the TLS Certificate

> **NOTE:** You can skip this step if you have automated certificate management, such as cert-manage, or deployment processes that handle TLS configuration.

Create a TLS secret with certificate and key provided by your Certificate Authority of choice. You may use self-signed if you prefer. Place certificates in the **skylar-ai-configs/certs** directory.

```
mkdir skylar-ai-configs/certs
cd skylar-ai-configs/certs

# Below OpenSSL command is just an example of self signed cert. Recom-
mended to use real
cert signed by Certificate Authority
openssl req -x509 -newkey rsa:2048 -nodes -sha256 -days 825 \
  -keyout tls.key -out tls.crt \
  -subj "/CN=on-prem-advisor.ai.sciencelogic.biz" \
  -addext "subjectAltName=DNS:example.com" \
  -addext "basicConstraints=critical,CA:TRUE"

chmod 600 tls.key

kubectl create secret tls tls-certificate \
  --cert=tls.crt --key=tls.key \
  --namespace=sky \
  --dry-run=client -o yaml | kubectl apply -f -
```

Step 5: Deploying the Skylar AI Platform

To deploy the Skylar AI platform:

1. Deploy Skylar AI with the scaling profile and your custom overrides:

```
helm upgrade --install skylar-production \
  oci://registry.scilo.tools/skylar/skylar-charts \
  --namespace skylar-production \
  --values skylar-ai-configs/analytics-micro.yaml \
  --values skylar-ai-configs/override-analytics.yaml
```

2. If you are enabling Skylar Advisor, add the following options to the command. The GPU file name is a sample. You can name the file to match the GPU you deploy.

```
  --values skylar-ai-configs/override-advisor.yaml
  --values skylar-ai-configs/l40s-rtx6000-combined.yaml
```

3. If you enable the soft HA or no HA option, add the following option to the command:

```
--values skylar-ai-configs/no-ha.yaml
```

4. Save the command that you used. You will use the same command for future upgrades.

```
vi skylar-ai-configs/upgrade.txt
```

5. Paste the Helm upgrade command into the text file.

Configure the TLS Certificate

Create a TLS secret with your certificate and private key.

> **NOTE:** Skip this step if you have automated certificate management, such as cert-manager, or deployment processes that handle TLS configuration.

Create the TLS secret with your certificate:

```
kubectl create secret tls skylar-tls-secret \
  --cert=path/to/your/certificate.crt \
  --key=path/to/your/private.key \
  --namespace=skylar-production
```

Validation and Access

Verify Deployment Status

To verify the deployment status:

1. Check that all pods are running:

```
kubectl get pods -n skylar-production
```

2. Check the volumes:

```
kubectl get pvc -n skylar-production
```

Check Ingress and User Interface

The following commands show the configured domain name. Use that domain name to access the user interface (UI).

To check the ingress configuration:

1. Verify the ingress configuration:

```
kubectl get ingress -n skylar-production
```

2. If you use the Gateway API, verify that the HTTP routes and listener sets appear:

```
kubectl get httproutes
```

```
kubectl get listenerset
```

Useful Commands

Use the following commands to check the deployment status and troubleshoot:

1. Check the deployment status:

```
helm status skylar-prod -n skylar-production
```

2. View pod logs for troubleshooting:

```
kubectl logs <pod-name> -n skylar-production
```

3. Describe a pod to view detailed information:

```
kubectl describe pod <pod-name> -n skylar-production
```

Troubleshooting

Registry Authentication Issues

If you encounter image pull errors, try the following:

- ***Verify CLI Secret***. Ensure that the CLI secret has not expired in Harbor.
- ***Test Registry Login***. Re-authenticate with `helm registry login`.
- ***Check Secret Format***. Verify that you created the Kubernetes secret correctly.
- ***Namespace Access***. Ensure that the secret exists in the correct namespace.

Chapter

2

Skylar Analytics and Skylar Advisor Sizing Guide

Overview

Skylar Analytics and Advisor are deployed on a Kubernetes infrastructure. The sizings in this chapter below cover the necessary resources for only the *applications*, not the entirety of the Kubernetes cluster that hosts the application.

Additional resources are required for system and cluster-level services, such as DNS, networking, security. As a result, *do not size a node to these requirements alone*. ScienceLogic cannot determine the additional overhead needed for cluster components; this must be accounted for separately.

NOTE: Not all components are currently highly available (HA). Only critical services support HA today, with additional components for HA support coming in future releases.

Node Storage Capacity: Around 350gbs per node of internal storage for images and Kubernetes-related functions. Ensure the root directory has most of this space provisioned. Storage below is referencing Attached storage (EBS).

This chapter covers the following topics:

Sizing for Skylar Analytics 14

Additional Sizing for Adding Skylar Advisor to Skylar Analytics 14

Largest Single Pod Sets Minimum Node Size 15

Data Visualization (DataViz) Sizing 15

Additional Sizing for Adding Streaming Telemetry 16

Sizing for Skylar Analytics

With High Availability

Deployment	Datapoints per Minute (DPM) Based on Cluster DPM	Pods	CPU	Memory	Storage
Micro (POC)	0 - 5,000	35	9.66	50.9 Gi	211 Gi
Small	5,000 - 30,000	39	21.39	100.1 Gi	766 Gi
Medium	30,000 - 215,000	39	39.16	271.4 Gi	976 Gi
Large	215,000 - 300,000	39	65.67	503.1 Gi	1.03 Ti
Extra Large	300,000 - 900,000	39	111.61	879.2 Gi	5.77 Ti

Without High Availability

Deployment	Datapoints per Minute (DPM)	Pods	CPU	Memory	Storage
Micro (POC)	0 - 5,000	32	8.86	50.3 Gi	207 Gi
Small	5,000 - 30,000	32	18.63	76.2 Gi	506 Gi
Medium	30,000 - 215,000	32	34.67	228.3 Gi	704 Gi
Large	215,000 - 300,000	32	58.30	426.6 Gi	768 Gi
Extra Large	300,000 - 900,000	32	101.95	795.8 Gi	4.70 Ti

Additional Sizing for Adding Skylar Advisor to Skylar Analytics

Add the following resources to the existing provisioned Skylar Analytics resources.

The storage numbers are for backend services and include 20 GB reserved for the Skylar Advisor Corpus. The Corpus can be much larger, so you must account for additional storage specifically for the Corpus.

With High Availability

GPU Profile	+ Pods	+ CPU	+ Memory	GPU	+ Storage
4 GPU (l40s/RTX)	32	74.52	407.8 Gi	4	394 Gi
8 GPU (h100)	38	99.82	532.0 Gi	8	394 Gi

Without High Availability

GPU Profile	+ Pods	+ CPU	+ Memory	GPU	+ Storage
l40s-rtx6000-4gpu	30	69.92	371.0 Gi	4	394 Gi
h100-resources	36	95.22	495.2 Gi	8	394 Gi

Largest Single Pod Sets Minimum Node Size

At least one node must be large enough to accommodate the ClickHouse database. This requirement is generally not an issue for smaller deployments, but it becomes an important consideration for Medium deployments and larger.

For example, a Medium deployment requires a ClickHouse node with at least 18.52 cores and 147.4 Gi of memory. After you set aside the ClickHouse node, you can size the remaining nodes evenly.

Deployment	Largest Pod	CPU	Memory
Micro	n/a	n/a	n/a
Small	ClickHouse	4.71	18.5 Gi
Medium	ClickHouse	18.52	147.4 Gi
Large	ClickHouse	36.91	294.5 Gi
Extra Large	ClickHouse	73.72	588.9 Gi

Data Visualization (DataViz) Sizing

The previous sizing tables are for the Skylar AI application. Skylar AI can be multi-tenant, and the sizing recommendations change as you add tenants. Add the following requirements *on top of* the requirements in the previous tables, per tenant.

Tenant Size (Based on Tenant DPM)	DataViz Resourcing
Micro	5 CPU and 32 GB RAM
Small	5 CPU and 32 GB RAM
Medium	5 CPU and 64 GB RAM
Large	5 CPU and 64 GB RAM
Extra Large	8 CPU and 128 GB RAM

Additional Sizing for Adding Streaming Telemetry

If you are including Streaming Telemetry, add the following resources to the existing provisioned Skylar Analytics resources. The requirements are the same for every deployment size, with and without HA.

CPU	Memory	Storage
1.67	2.7 Gi	None

Chapter

3

Troubleshooting and Security for Skylar AI

Overview

This chapter includes a list of common issues with the installation process for Skylar AI. This chapter also include security considerations, backup and recovery procedures, and a list of potentials topics where you might need to contact Skylar AI Enablement.

This chapter covers the following topics:

Troubleshooting the Installation Process *18*

Security Considerations *20*

Backup and Recovery *21*

Support *23*

Troubleshooting the Installation Process

> **IMPORTANT:** The information in this section is relevant for both Skylar Analytics and Skylar Advisor deployments.

Common Issues

Image Pull Failures

- Verify that the registry secret is correctly configured.
- Check that the CLI secret is still valid in Harbor.
- Ensure the namespace has access to the registry secret.
- Verify that the Helm registry login was successful.

TLS Certificate Issues

- Verify that the TLS secret contains a valid certificate and key.
- Check that certificate matches the FQDN.
- Ensure the certificate is not expired.

DNS Resolution Issues

- Verify that the FQDN points to the correct load balancer IP.
- Check the DNS propagation if recently updated.

Ingress Controller Issues

- Verify that the ingress controller is running and healthy.
- Check the ingress controller logs for errors.
- Ensure that the ingress class name matches your controller.
- Verify that the ingress annotations are compatible with your controller.

Pod Startup Failures

- Check the resource constraints and storage availability.

Database Connection Issues

- Ensure that the ClickHouse and PostgreSQL pods are healthy.

Large Upload/Download Issues

- Verify that the ingress controller supports large request bodies (256 MB and up).
- Check the proxy timeout configurations.
- Ensure that the proper buffering settings are applied.

Cloud-Specific Issues

AWS

- Check the IAM permissions for EBS/EFS access.

GCP

- Verify the service account permissions for persistent disks.

Azure

- Ensure the proper RBAC for storage resources.

Useful Commands

Deployment Status Commands

To check the deployment status:

```
helm status skylar-prod -n skylar-production
```

To view pod logs for troubleshooting:

```
kubectl logs <pod-name> -n skylar-production
```

To describe the pod for detailed information

```
kubectl describe pod <pod-name> -n skylar-production
```

Infrastructure Verification Commands

To check the ingress controller status (adjust the namespace based on your setup):

```
kubectl get pods -n ingress-nginx
```

To check persistent volume claims:

```
kubectl get pvc -n skylar-production
```

To test registry connectivity:

```
kubectl run test-registry --image=registry.scilo.tools/skylar/skylar-
charts:latest \ --dry-run=client -o yaml | kubectl apply -f -
```

Registry Authentication Issues

If you encounter image pull errors:

- Make sure the CLI secret has not expired in Harbor.
- Re-authenticate with the Helm registry login.
- Verify that the Kubernetes secret was created correctly.
- Make sure the secret exists in the correct namespace.

TLS Certificate Issues

If you encounter TLS-related problems:

- Verify that the certificate is valid for your FQDN.
- Make sure the certificate is in PEM format.
- Include intermediate certificates if required.
- Verify that the private key matches the certificate.

Ingress Controller Issues

If ingress resources are not working:

- Verify that the ingress controller is running.
- Make sure the ingress class name matches your controller.
- Check the ingress controller logs for configuration errors.
- Make sure the backend services are healthy and have endpoints.
- Make sure the ingress annotations are supported by your controller.

Security Considerations

> **IMPORTANT:** The information in this section is relevant for both Skylar Analytics and Skylar Advisor deployments.

Security Best Practices

- **Registry Credentials**: Store CLI secrets securely and rotate them regularly.
- **TLS Certificates**: Ensure certificates are from trusted CAs and renewed before expiration.

- **Network Policies**: Consider implementing Kubernetes network policies to restrict inter-pod communication.
- **RBAC**: Configure appropriate role-based access controls for the Skylar AI namespace.
- **Ingress Security**: Configure appropriate security headers and rate limiting on your ingress controller.
- **Secrets Management**: Use external secret management solutions, such as HashiCorp Vault, or AWS Secrets Manager for production.
- **Cloud Security**: Follow cloud provider security best practices.
- **Third-Party Components**: Monitor security advisories for included third-party components.

Third-Party Security Responsibilities

While ScienceLogic validates and updates third-party chart versions with each Skylar AI release, customers should:

- **Monitor Security Advisories**: Stay informed about security issues in included third-party components.
- **Plan for Updates**: Be prepared to upgrade Skylar AI when security patches are available.
- **Vulnerability Assessment**: Include third-party components in security scanning and assessment processes.
- **Risk Management**: Understand the security posture of all included dependencies.

Backup and Recovery

> **IMPORTANT:** The information in this section is relevant for both Skylar Analytics and Skylar Advisor deployments.

Critical Data Components

Skylar AI stores critical data in the following persistent volumes, which must be backed up regularly.

ClickHouse Data Volumes

- Contains analytics data, metrics, and time-series information.
- Recommended backup frequency: Daily with 30-day retention.

PostgreSQL Data Volumes

- Contains application metadata, user data, and configuration.
- Recommended backup frequency: Daily with 30-day retention.

Backup Strategy Options

This section covers the different types of backup strategies you can use with Skylar AI.

Volume-Level Backups

- Use cloud provider snapshot capabilities, such as AWS EBS, GCP Persistent Disks, or Azure Disks.
- Implement Kubernetes volume snapshots using CSI drivers.
- Consider third-party backup solutions like Velero for comprehensive cluster backup.

Application-Level Backups

- Database-native backup tools for ClickHouse and PostgreSQL.
- Export application configuration and secrets.
- Backup Helm chart values and deployment configurations.

Hybrid Approach

- Combine volume snapshots for fast recovery with application-level backups for granular restore options.
- Implement both local and off-site backup storage for disaster recovery.

Backup Verification

Regular Testing Requirements

- Monthly: Test backup restoration procedures in non-production environment.
- Quarterly: Full disaster recovery simulation.
- Document and validate recovery time objectives (RTO) and recovery point objectives (RPO).

Monitoring and Alerting

- Monitor backup job completion and success rates.
- Alert on backup failures or missing backups.
- Verify backup accessibility and integrity.

Configuration Backup

Helm and Kubernetes Configuration

- Export and store Helm values files.
- Backup Kubernetes secrets and ConfigMaps.
- Maintain version-controlled infrastructure as code.

> **CAUTION:** Store configuration backups containing secrets in encrypted storage with restricted access. Never commit secrets to version control systems.

For detailed backup implementation guidance, consult your cloud provider's backup documentation and the Kubernetes Backup Best Practices.

Support

> **IMPORTANT:** The information in this section is relevant for both Skylar Analytics and Skylar Advisor deployments.

For deployment assistance, configuration guidance, or troubleshooting support, contact Skylar AI Enablement. They can provide:

- Environment-specific override file templates.
- Scaling recommendations based on your requirements.
- Custom configuration for enterprise integrations.
- Cloud-specific deployment guidance.
- Registry access troubleshooting.
- TLS certificate configuration assistance.
- Ingress controller configuration guidance.
- Post-deployment optimization.
- Third-party component guidance (limited to integration aspects only).

> **NOTE:** Support for third-party components, such as Bitnami charts, is limited to integration and configuration guidance. For issues specific to these components, consult their respective documentation and support channels.

© 2003 - 2026, ScienceLogic, Inc.

All rights reserved.

ScienceLogic™, the ScienceLogic logo, and ScienceLogic's product and service names are trademarks or service marks of ScienceLogic, Inc. and its affiliates. Use of ScienceLogic's trademarks or service marks without permission is prohibited.

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information herein, the information provided in this document may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described herein at any time without notice.



800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010