SNO Disconnected Deployment Guide

Single Node OpenShift (SNO) Disconnected Deployment Guide

Created Date: January 14, 2026

Target Version: OpenShift Container Platform 4.16

This documentation provides a hardened, end-to-end workflow for deploying a Single Node OpenShift (SNO) cluster in a disconnected environment. We utilize the **Agent-based Installer**, which embeds configurations directly into a bootable ISO, eliminating the need for external bootstrap infrastructure during the final boot.

Working Environment Definitions

System Component	Description & Technical Role	Network Placement
Connected Bastion	RHEL 8/9 host used to mirror platform and operator data from Red Hat registries.	Internet Facing
Disconnected Bastion	RHEL 8/9 host used to build the Agent ISO and host the local mirror registry (Quay).	Air-Gapped
Target SNO Node	Physical server or VM where the OpenShift 4.16 cluster will be deployed.	Air-Gapped

Infrastructure & Network Requirements

Before beginning the implementation, ensure the following prerequisites are met within the air-gapped environment:

Requirement	Description	Specifics
Static IPs	Dedicated IP addresses for core infrastructure components.	One for Bastion
—	—	One for SNO Node
DNS Records	Resolvable records pointing to the single node IP. *.apps must point to SNO IP.	api, api-int, *.apps
—	—	registry-fqdn
NTP	Mandatory local time synchronization source.	Local NTP Server IP
Admin Credentials		

Requirement	Description	Specifics
	Red Hat Portal ID and SSH Key Pair for node access.	Pull Secret, Public SSH Key

Pre-Flight Resource Validation

Failure to meet these minimums will cause the SNO installation or Day 2 operator deployments to fail. Ensure RHCOS version matches OCP 4.16.

Category	Technical Requirement Justification	Documentation Source
SNO Node Storage	A minimum of 120 GB is required for the base RHCOS installation.	Installing on a single node
Bastion Storage	200 GB minimum accounts for platform images and common operators.	Mirroring images for disconnected installation
Registry Persistence	The Disconnected Bastion must house the Quay database and all layers permanently.	Creating a mirror registry
SNO Compute	Minimum requirements are 8 vCPUs and 16-32 GB of RAM.	SNO Preparing to Install
Installer Specs	The installer host requires space for the binary and a ~1 GB bootable ISO.	Agent-based Installer Guide

The Sneakernet Workflow

In a disconnected environment, the “Sneakernet” process is the manual method of bridging the air-gap using physical media (such as external SSDs).

Phase	Action	Requirement
Collection	Mirroring platform images and operators from Red Hat to local media.	Connected Bastion + Physical Media
Transition	Physically moving the media through security checkpoints to the air-gapped zone.	Secure Chain of Custody
Ingestion	Uploading mirrored content from media into the local disconnected registry.	Disconnected Bastion + Local Registry

---

Implementation Roadmap

Phase	Objective
Day 0: Preparation	Establish the connected staging environment and gather binaries
—	Declaratively mirror OCP images and operators for physical transfer
—	Deploy and harden a local Quay registry for air-gapped ingestion
Day 1: Installation	Define the SNO network and cluster logic via YAML manifests
—	Build the self-contained agent.iso and initiate the hardware boot
Day 2: Hardening	Resolve common air-gap and certificate-related deployment faults

Phase	Objective
—	Final verification of the environment prior to ISO execution
—	Validating cluster health, storage, and supply chain integrity
—	Implementing local storage (LVMS) and log aggregation

Appendix: Methodology & Scope

While official Red Hat documentation provides foundational technical references for individual components, this guide serves as an architectural blueprint specifically synthesized for **high-security, disconnected environments.**

The “Secret Sauce”

- **Synthesis of Fragmentation**: This guide eliminates the need to cross-reference multiple manuals by providing a single, linear assembly line for deployment.
- **Agent-Based Architecture**: Utilizing the Agent-based Installer creates a “Cluster in a Box,” reducing external infrastructure dependencies.
- **Hardened Security by Default**: This methodology prioritizes established chain-of-trust protocols using Internal CAs and explicit certificate injection into the `additionalTrustBundle`.
- **Pre-Flight Rigor**: Explicit “Go/No-Go” gates verify DNS, NTP, and Registry availability before physical provisioning begins.

Step 1: Staging the Connected Bastion

Created Date: January 14, 2026 Status: Environment Preparation

The **Connected Bastion** serves as the staging area for the entire deployment. It is the only machine with internet access, used to download the OpenShift binaries and mirror the container images required for the air-gapped installation.

Infrastructure Requirements

Component	Requirement	Role
OS	RHEL 8.x or 9.x	Base operating system for mirroring tools
Storage	200 GB+	Local cache for mirrored container images
Account	Red Hat Customer Portal	Access to registry.redhat.io and pull secrets
Tooling	oc, openshift-install, oc-mirror	Primary binaries for cluster orchestration

---

Binary Acquisition & Verification

Execute these commands to download the OpenShift 4.16.x binaries. Verifying the checksums is critical to ensure the integrity of the installer before it is moved into a high-security environment.

```
# 1. Download the OpenShift Client (oc) and Installer
wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/
       latest-4.16/openshift-client-linux.tar.gz
wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/
       latest-4.16/openshift-install-linux.tar.gz

# 2. Download the checksum file
wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/
       latest-4.16/sha256sum.txt

# 3. Verify the integrity of the binaries
sha256sum --check --ignore-missing sha256sum.txt

# 4. Extract and move to a persistent path
```

```
tar -xvf openshift-client-linux.tar.gz
tar -xvf openshift-install-linux.tar.gz
sudo mv oc kubectl openshift-install /usr/local/bin/

# 5. Verify the binaries are in your PATH and functional
oc version --client
openshift-install version
```

---

oc-mirror Plugin Installation (v2)

The `oc-mirror` plugin is required for the declarative mirroring workflow used in Step 2.

```
# 1. Download the oc-mirror v2 plugin
wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/
        latest-4.16/oc-mirror.tar.gz

# 2. Extract and install
tar -xvf oc-mirror.tar.gz
chmod +x oc-mirror
sudo mv oc-mirror /usr/local/bin/

# 3. Verify installation
oc-mirror version
```

Environment Persistence

To ensure the tooling is available across all terminal sessions on the Bastion, verify the following configuration.

Task	Action	Verification
Path Persistence	Ensure /usr/local/bin is in your secure_path	echo $PATH
Alias Setup	Optional: Alias k=kubectl for efficiency	alias k
Profile Loading	Reload .bashrc after manual path changes	source ~/.bashrc

---

Architectural Justifications & Reference Notes

Category	Technical Requirement Details	Documentation Source
Binary Choice	The client (oc) version must be equal to or newer than the target cluster version (4.16).	OCP CLI Installation
Checksum Verification	Integrity checks prevent the introduction of corrupted or tampered binaries into the air-gap.	OCP Installing on Bare Metal
oc-mirror v2	Version 2 of the plugin provides a more stable, declarative approach to mirroring compared to v1.	oc-mirror v2 Documentation

---

Step 2: Mirroring Content for the Air-Gap

Created Date: January 14, 2026 Status: Data Collection

In a disconnected environment, you must manually "mirror" all required container images from Red Hat's registries to local media. This process ensures that every component needed for the SNO installation is physically present before moving to the air-gapped site.

Storage & Workspace Verification

Before beginning the mirror, ensure the Bastion has sufficient local storage. The mirroring process requires a significant workspace for metadata and image layers.

Category	Requirement	Verification Command
Disk Space	200 GB+ available on the mirroring partition	df -h
Workspace	Write permissions for the oc-mirror-workspace directory	ls -ld .
Backend	Physical media (SSD/HDD) formatted and mounted	lsblk

ImageSetConfiguration (Step 2.1)

The `imageset-config.yaml` file defines exactly which versions of OpenShift and which Operators will be mirrored.

```yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v2alpha1
mirror:
  platform:
    channels:
    - name: stable-4.16
      type: ocp
      minVersion: 4.16.0
      maxVersion: 4.16.0
  operators:
    - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.16
      packages:
        - name: lvms-operator
        - name: cluster-logging
  additionalImages:
    - name: registry.redhat.io/ubi9/ubi:latest
```

Executing the Mirror (Step 2.2)

Use the `oc-mirror` plugin to pull the images. It is highly recommended to perform a dry run first to validate the configuration file without downloading data.

```
# 1. Perform a dry run to validate the ImageSetConfiguration
oc-mirror --config ./imageset-config.yaml file://./mirror-data --
        dry-run

# 2. Execute the actual mirror to local storage
oc-mirror --config ./imageset-config.yaml file://./mirror-data

# 3. Verify the generated 'mirror-data' directory contains the
        'v2' metadata
ls -R ./mirror-data
```

Architectural Justifications & Reference Notes

Category	Technical Requirement Details	Documentation Source
Workspace Capacity	The oc-mirror tool stores metadata and blob layers in the local workspace; insufficient space will cause a mid-process failure.	Mirroring images for disconnected installation
Dry Run Validation	Using the –dry-run flag catches syntax errors in the YAML and confirms package availability before committing to a massive download.	oc-mirror v2 Documentation
Platform Pinning	Setting minVersion and maxVersion to the same value (4.16.0) prevents the	oc-mirror Reference

Category	Technical Requirement Details	Documentation Source
	tool from mirroring every version in the stable channel.	
Operator Filtering	By listing specific packages (lvms, logging), you reduce the final mirror size by gigabytes compared to mirroring the full catalog.	Filtering operator catalogs

Step 3: Local Registry & Ingestion

Created Date: January 14, 2026 Status: Registry Configuration

Once the physical media has been moved to the air-gapped environment, the mirrored content must be “ingested” into a local registry. This registry becomes the authoritative source of truth for the SNO node, replacing the need for an internet connection to Red Hat’s servers.

Registry Deployment (Step 3.1)

We utilize the Red Hat mirror-registry tool to deploy a local Quay instance on the Disconnected Bastion.

```
# 1. Install the mirror-registry on the air-gapped host
./mirror-registry install --quayHostname <registry_fqdn> --
        quayRoot /opt/quay

# 2. Add the Registry CA to the Bastion's trusted store
sudo cp /opt/quay/quay-install/quay-config/root-ca.crt /etc/pki/
        ca-trust/source/anchors/
sudo update-ca-trust extract
```

Data Ingestion (Step 3.2)

Use the same `oc-mirror` plugin used in Step 2 to upload the images from your physical media into the local Quay registry.

```
# 1. Execute the ingestion from media to the local registry
oc-mirror --from ./mirror-data docker://<registry_fqdn>:8443
```

Credential & Trust Hardening

This is a critical phase where you combine the default Red Hat pull secret with your new local registry credentials to create a unified secret for the installer.

Task	Action	Verification
Pull Secret Merge	Use ‘jq’ or a text editor to add the local registry auth to the Red Hat pull-secret.json	cat pull-secret.json
Certificate Bundle	Ensure the additionalTrustBundle contains the full CA chain (Intermediate + Root)	openssl x509 -text
Auth Verification	Run ‘podman login’ to verify the merged secret works against the local registry	podman login :8443

Pull Secret Merge Logic

The resulting `pull-secret.json` must contain both the original Red Hat auths and the new local registry auth. A JSON syntax error here is a common cause of installation failure.

```
{
  "auths": {
    "cloud.redhat.com": { "auth": "...", "email": "..." },
    "quay.io": { "auth": "...", "email": "..." },
    "registry.redhat.io": { "auth": "...", "email": "..." },
    "<registry_fqdn>:8443": { "auth":
         "BASE64_ENCODED_CREDENTIALS" }
  }
}
```

Architectural Justifications & Reference Notes

Category	Technical Requirement Details	Documentation Source
Pull Secret Integrity	The installer requires a single unified pull secret. A manual merge must maintain valid JSON structure for the node to pull images.	Mirroring images for disconnected installation
Certificate Chains	If using an organization-issued certificate, the ssl.crt must include the full chain. If the intermediate is missing, the node will reject the registry.	Creating a mirror registry
Registry Persistence	The local registry must remain active for the life of the cluster. If the Quay service stops, the cluster will lose its ability to scale or recover.	Disconnected installation overview
Trust Bundle Injection	The additionalTrustBundle in install-config.yaml allows the RHCOS operating system to trust your internal registry during the bootstrap phase.	SNO Preparing to Install

---

Step 4: SNO Node Configurations

Created Date: January 14, 2026 Status: Manifest Definition

The Agent-based Installer requires two primary configuration files: `install-config.yaml` and `agent-config.yaml`. These files define the cluster's logical identity and the physical hardware networking respectively.

Workspace Setup

Create a dedicated directory for the installation assets. This workspace will house the manifests and eventually the generated metadata.

```
mkdir sno-config
cd sno-config
```

Logical Configuration (install-config.yaml)

This file defines the cluster name, the base domain, and the security trust bundle.

```
apiVersion: v1
baseDomain: <domain_name>
compute:
- name: worker
  replicas: 0
controlPlane:
  name: master
  replicas: 1
metadata:
  name: <cluster_name>
platform:
  none: {}
pullSecret: '<merged_pull_secret>'
sshKey: '<ssh_public_key>'
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  <registry_ca_contents>
  -----END CERTIFICATE-----
```

Physical Configuration (agent-config.yaml)

This file maps the logical cluster to the physical hardware, including static networking and disk selection.

```yaml
apiVersion: v1
kind: AgentConfig
metadata:
  name: <cluster_name>
hosts:
  - hostname: <sno_hostname>
    interfaces:
      - name: <interface_name>
        macAddress: <mac_address>
    networkConfig:
      interfaces:
        - name: <interface_name>
          type: ethernet
          state: up
          ipv4:
            enabled: true
            address:
              - ip: <sno_node_ip>
                prefix-length: 24
      dns-resolver:
        config:
          server:
            - <dns_server_ip>
      routes:
        config:
          - destination: 0.0.0.0/0
            next-hop-address: <gateway_ip>
            next-hop-interface: <interface_name>
```

Hardware & Path Precision

For a successful SNO boot in a disconnected environment, the following hardware details must be verified.

Component	Requirement	Recommendation
Interface Name	Must match the RHCOS kernel name (e.g., eno1, ens3)	Verify via ‘ip link’ on a Live ISO if unsure
MAC Address	Must match the physical NIC intended for PXE/ Boot	Double-check against physical labels or BIOS
Installation Disk	The target drive for the RHCOS operating system	Use /dev/sda or /dev/ nvme0n1 consistently
Disk Pathing	Persistence across reboots in varied hardware	Use /dev/disk/by-id/ for unambiguous identification

Architectural Justifications & Reference Notes

Category	Technical Requirement Details	Documentation Source
Interface Mapping	In Agent-based installs, if the interface name in networkConfig does not match the hardware, the static IP will not bind, causing the node to be unreachable.	Agent-based Installer Guide
Disk Selection	The installer defaults to the first available disk. Forcing a specific disk path prevents accidental overwrites of existing data drives.	SNO Preparing to Install

replicas: 0

Category	Technical Requirement Details	Documentation Source
	In a Single Node OpenShift deployment, worker replicas must be set to 0 as the master node handles both roles.	Installing SNO Overview
additionalTrustBundle	This field is mandatory for disconnected installs. Without it, the node cannot verify the TLS certificate of the local Quay registry.	Disconnected installation mirroring

---

Step 5: ISO Generation & Hardware Boot

Created Date: January 14, 2026 Status: Physical Provisioning

The final phase of the installation process involves converting the YAML configurations into a bootable `agent.iso`. This file is self-contained, meaning it includes all the logic required to reach out to your local registry and provision the Single Node OpenShift cluster.

Generating the Image (Step 5.1)

Run the installer from your `sno-config` directory. This command consumes the `install-config.yaml` and `agent-config.yaml` files.

```
# 1. Generate the bootable agent.iso
openshift-install agent create image --dir ./sno-config
```

```
# 2. Verify the image creation in the directory
ls -lh ./sno-config/agent.iso
```

Media Verification & Transfer

Before booting the target hardware, ensure the ISO has been transferred to the boot media (USB or Virtual Media) without corruption.

Verification Task	Command / Action	Importance
ISO Checksum	sha256sum ./sno-config/agent.iso	Detects local filesystem corruption before the transfer
Media Integrity	Compare checksum of the ISO on the USB to the source	Prevents boot failures due to faulty physical hardware
Boot Mode	Ensure BIOS is set to UEFI (unless using Legacy)	Alignment with the RHCOS partition table requirement

---

Sensitive Data & Cleanup

The generation process creates several metadata files in the `./sno-config` directory. These files contain sensitive credentials that must be managed according to your security posture.

File Path	Sensitivity Level	Justification
auth/kubeadmin-password	High	Initial cluster administrator password
auth/kubeconfig	High	Full administrator access to the cluster API
metadata.json	Medium	Contains cluster ID and infrastructure metadata

---

Initiating the Boot (Step 5.2)

Mount the `agent.iso` via the server's Out-of-Band management (iDRAC/iLO) or insert the physical USB. Once the server starts, it will automatically initiate the "Agent" flow.

```
# 1. Monitor the installation progress from the Bastion
export KUBECONFIG=./sno-config/auth/kubeconfig
openshift-install agent wait-for install-complete --dir ./sno-
        config
```

Architectural Justifications & Reference Notes

Category	Technical Requirement Details	Documentation Source
Self-Contained ISO	The agent.iso includes the 'Ignition' configuration. Once booted, it requires no further manual input from the administrator.	Agent-based Installer Guide
ISO Verification	Corrupted ISOs often manifest as 'Kernel Panic' or 'SquashFS errors' mid-boot; verification saves hours of physical troubleshooting.	OCP Installing on Bare Metal
credential-cleanup	After the cluster is verified 'Ready', the local sno-config directory should be archived or secured to prevent unauthorized access.	Security and Compliance Guide
wait-for-completion	This command tracks the transition from the bootstrap phase	SNO Installing on a single node

Category	Technical Requirement Details	Documentation Source
	to the final production state of the SNO control plane.	

Pre-Flight Deployment Checklist

Created Date: January 14, 2026 Status: Final Verification

Before executing the `agent.iso` on the physical hardware, use this checklist to ensure the air-gapped environment is fully prepared. Failure to verify these items often results in installation hangs that are difficult to debug mid-process.

Environment & Infrastructure Gates

Category	Checkpoint Item	Verification Method
DNS	api.. resolves to SNO IP	dig or nslookup
DNS	*.apps.. resolves to SNO IP	dig or nslookup
DNS	resolves to Bastion IP	dig or nslookup
NTP	Local NTP Server is reachable from SNO segment	ping
Time Sync	Bastion clock is synchronized with Local NTP	chronyc sources -v
Connectivity	Port 8443 is open on Disconnected Bastion	telnet 8443

Hardware & Manifest Alignment

Ensure the physical server characteristics match the logic defined in your `agent-config.yaml` and `install-config.yaml`.

Component	Checkpoint Item	Verification Method
Interface	MAC Address matches agent-config.yaml exactly	Physical label or BIOS check
Interface	Kernel name (eno1/ ens3) matches agent-config.yaml	Verified via Live ISO/ Hardware spec
Storage	Target disk is “clean” (no existing partitions)	wipefs (if re-using hardware)
Resources	Minimum 8 vCPUs and 16 GB to 32 GB RAM available	BIOS/Hardware specification
Firmware	Boot mode set to UEFI	BIOS Boot settings

Registry & Supply Chain Integrity

Category	Checkpoint Item	Verification Method
Trust	Registry CA (and Intermediate) is in additionalTrustBundle	Review install-config.yaml
Auth	Pull Secret contains valid auth for local registry	jq .auths pull-secret.json
Content	oc-mirror report confirms all images are present	Review mirror metadata
Media	agent.iso checksum matches source on Bastion	sha256sum /dev/sdX

Architectural Justifications & Reference Notes

Category	Technical Requirement Details	Documentation Source
NTP Accuracy	OpenShift’s etcd and certificate verification mechanisms require sub-second clock synchronization between the node and its time source.	SNO Preparing to Install
MAC/Interface	The Agent-based installer uses the MAC address to bind the static IP; if the interface name is wrong, the network stack will not initialize.	Agent-based Installer Guide
Wildcard DNS	The *.apps record is required for the OpenShift Ingress Controller to route traffic to the console and user applications.	OCP Networking Overview
UEFI Requirement	RHCOS uses a specific partition layout that requires UEFI firmware for modern secure boot and disk management capabilities.	OCP Installing on Bare Metal

Post-Installation Verification Checklist

Created Date: January 14, 2026 Status: Cluster Validation

Once the `wait-for install-complete` command returns successfully, use this checklist to verify that the cluster is healthy and that the air-gapped configuration is correctly routing image requests to your local registry.

Core Cluster Health

Category	Checkpoint Item	Verification Command
Nodes	SNO node is in Ready status	oc get nodes
Operators	All ClusterOperators are Available: True	oc get co
Version	Cluster version matches 4.16.x	oc get clusterversion
Certificates	No pending CSRs require approval	oc get csr

Air-Gap & Registry Validation

This section ensures the “Secret Sauce” of the disconnected installation (the image redirection) is functioning as intended.

Category	Checkpoint Item	Verification Command
ICSP	ImageContentSourcePolicy is present and active	oc get icsp
Registry	Internal image-registry operator is Available	oc get co image-registry
Storage	Internal registry has a valid storage backend	oc get configs.imageregistry.operator.openshift.io/ cluster -o yaml
Redirection	Pods are pulling from	oc get pods -A -o jsonpath=‘{.items[*].spec.containers[*].imag

Persistence & Storage Readiness

Category	Checkpoint Item	Verification Command
StorageClass	A default StorageClass is present (after LVMS install)	oc get sc
PVs	Persistent Volumes can be bound to claims	oc get pvc -A
Etcd	Etcd is healthy and synchronized	oc get pods -n openshift-etcd

Architectural Justifications & Reference Notes

Category	Technical Requirement Details	Documentation Source
ICSP Role	The ImageContentSourcePolicy is the mechanism that instructs the node to transparently swap ‘quay.io’ links for your local registry FQDN.	Disconnected installation mirroring
Registry Storage	In a Single Node OpenShift (SNO) deployment, the internal registry cannot use ‘shared’ storage like NFS easily; it must be configured for ‘EmptyDir’ or a local PV.	Image Registry Operator
CSR Monitoring	While the Agent-installer handles most certificate signing, manual approval of Kubelet CSRs is	OCP Node Management

Category	Technical Requirement Details	Documentation Source
	occasionally required if the node name changes during boot.	
etcd Quorum	On a single node, etcd acts as a standalone database. High disk I/O latency on the SNO node is the leading cause of etcd instability.	SNO Performance and Scalability

---

Day 2: Post-Installation & Operational Health

Created Date: January 14, 2026 Status: Post-Install / Day 2 Operations

Once the Single Node OpenShift (SNO) cluster is “Ready,” the focus shifts to operationalizing the environment. In a disconnected solution, this involves finalizing local storage for persistent data and configuring cluster logging for auditability.

Day 2 Operational Overview

Task	Description
Storage Provisioning	Configuring the **LVM Storage (LVMS)** operator to provide persistent volumes (PVs) using the remaining local disk space.
Audit & Logging	Deploying the **Cluster Logging** operator to aggregate system and application logs for troubleshooting and compliance.
Cert-Manager Setup	(Optional) Deploying cert-manager to automate the management and issuance of certificates within the cluster.

Day 2 Reference Script

> **Note**: These commands assume the required operators were included in your initial mirror (Step 2). Ensure your `KUBECONFIG` is still exported.

```
# 1. Verify CatalogSource Initialization
# CRITICAL: The local catalog MUST show 'READY' before proceeding.
# If it shows 'PENDING' or 'IMAGEPULLBACKOFF', check your Registry CA trust.
oc get catalogsource -n openshift-marketplace

# 2. Prepare Namespaces
# Operators require specific namespaces to be created before configuration
oc create namespace openshift-storage
oc create namespace openshift-logging

# 3. Verify Operator Availability
# Once the catalog is ready, ensure the mirrored operators are visible in the hub
oc get packagemanifests -n openshift-marketplace | grep -E 'lvms|logging'

# 4. Configure Local Storage (LVMS)
# Create the LVMCluster resource to initialize local disk provisioning
cat <<EOF | oc apply -f -
apiVersion: lvm.topolvm.io/v1alpha1
kind: LVMCluster
metadata:
  name: lvmcluster-sno
  namespace: openshift-storage
spec:
  storage:
    deviceClasses:
      - name: vg1
        thinPoolConfig:
          name: thin-pool-1
          overprovisionRatio: 10
          sizePercent: 90
EOF

# 5. Verify Storage Readiness
# Wait 1-2 minutes, then verify the StorageClass is created and
        marked as default
oc get sc
```

Architectural Justifications & Reference Notes

Category	Technical Requirement Details	Documentation Source
CatalogSource Status	In air-gapped sites, the `CatalogSource` can take several minutes to pull the index image from your local registry. Installation of operators will fail if the status is not `READY`.	oc-mirror Documentation
Namespace Scoping	Operators like Logging and Storage should be isolated in their own namespaces (`openshift-logging`, `openshift-storage`) to maintain resource boundaries.	LVMS Operator Documentation
Disk Readiness	For LVMS to work, the target disk must be “clean” (no existing partitions or filesystems) outside of the main OS partition.	LVMS Operator Documentation
Resource Constraints	SNO nodes often have limited CPU/ RAM. Monitor the overhead of the Logging operator, as	SNO Performance and Scalability

Category	Technical Requirement Details	Documentation Source
	it can consume significant resources during high-burst log events.	
Disconnected Updates	Any future "Day 2" operators must be mirrored via the same `oc-mirror` process established in Step 2 to maintain the air-gapped supply chain integrity.	oc-mirror Documentation

---

Next Step: Since we've finalized the Day 2 operations, is there any other file in the repository you'd like me to review for "Secret Sauce" improvements or technical gaps?

Troubleshooting & FAQ

Created Date: January 14, 2026 Status: Maintenance & Support

Deployment in disconnected environments introduces unique failure points related to certificate trust, DNS resolution, and image mirroring. This guide covers the most common issues encountered when using the Agent-based Installer for Single Node OpenShift (SNO).

Agent Boot & Installation Logs

If the `wait-for install-complete` command hangs or the node fails to reach the "Ready" state, you must inspect the logs directly from the SNO node via SSH or the local console.

Log Source	Command	Purpose
Agent Installer	journalctl -u agent-installer -f	

Log Source	Command	Purpose
		Primary log for the initial bootstrap and image pull
Assisted Service	journalctl -u assisted-service	Tracks the orchestration of the installation steps
Kubelet	journalctl -u kubelet -f	Monitors the health of the Kubernetes node agent
Pod Logs	oc logs -n	Diagnostic data for specific system operators

Common Failure Scenarios

Symptom	Probable Cause	Resolution
ImagePullBackOff	Registry trust failure	Ensure the Registry CA is in additionalTrustBundle
ImagePullBackOff	Malformed pull secret	Verify JSON structure of the merged pull-secret.json
Node Not Found	DNS resolution failure	Verify api.. points to the SNO IP
etcd Degraded	Time drift or disk latency	Check chronyd status and disk I/O wait times
Static IP Missing	Interface name mismatch	Ensure agent-config.yaml matches the kernel device name

Certificate & Time Drift Management

In a disconnected environment, the lack of an external time source can cause the cluster's internal certificate authority to drift, leading to cluster-wide authentication failures.

- **Clock Skew**: If the SNO node clock differs from the Registry Bastion by more than a few seconds, the bootstrap process may reject the Registry's TLS certificate.
- **Certificate Expiry**: If the SNO node is powered down for more than 30 days, the Kubelet certificates may expire. Upon reboot, you may need to manually approve the CSRs (`oc get csr`) to restore node connectivity.
- **NTP Recovery**: If `chronyd` cannot reach the local NTP server, manually set the date on the SNO node using `date -s "YYYY-MM-DD HH:MM:SS"` to allow initial certificate validation to proceed.

Architectural Justifications & Reference Notes

Category	Technical Requirement Details	Documentation Source
Log Access	During the bootstrap phase, the standard 'oc' commands may not work. Accessing logs via 'journalctl' is the only way to debug pull failures.	OCP Troubleshooting Guide
Certificate Drift	In air-gapped sites, local NTP is the heartbeat of the cluster. Without it, the etcd quorum and API security will eventually collapse.	SNO Performance and Scalability
Pull Secret Logic	A single typo in the BASE64 string of a manually merged pull secret will	Disconnected installation mirroring

Category	Technical Requirement Details	Documentation Source
	prevent the node from authenticated with the local Quay registry.	
Interface Predictability	RHCOS uses predictable network naming. If the BIOS or a hardware change alters the naming (e.g., eno1 to eno2), the Agent config will fail.	Agent-based Installer Guide