Manage storage in NetApp Console

NetApp Console local deployment

NetApp
August 10, 2026

Table of Contents

Manage storage in NetApp Console

Learn about fleet management in NetApp Console local deployment

Fleet management in NetApp Console local deployment is the practice of organizing storage systems into logical groups so you can apply consistent policies, control access, and monitor health across related clusters. Instead of managing each cluster independently, fleet management lets you treat your fleet as a common pool of resources to support your data storage goals.

As your storage environment grows, managing individual clusters becomes impractical. Fleet management solves this by giving you a structured way to group related systems, delegate responsibilities, and ensure every cluster operates under the same standards.

Why fleet management matters

Fleet management addresses three core challenges:

- **Simplification through abstraction** - Fleet management abstracts away the complexity of managing individual storage infrastructure. Instead of dealing with cluster-by-cluster configuration, you manage your storage estate as a unified whole, reducing operational overhead and decision fatigue.
- **Consistency and scalability** - Without clear organization, different teams make different decisions for similar workloads, leading to fragmented configurations. Fleet management lets you apply standards across dozens of systems at once, ensuring new workloads start from the same baseline across teams and fleets.
- **Governance and control** - As teams grow, you need clear boundaries for who can manage what. Fleet management provides those boundaries through organizational structure and access control, ensuring that storage decisions remain governed and auditable.

How fleets organize your resources

Your organization's resource hierarchy consists of folders and fleets:

For a detailed overview of the hierarchy and access model, see Learn about folders and fleets in NetApp Console local deployment.

- **Organization** - The top level representing your company.
- **Folders** - Help you organize related fleets. For example, you might create a folder for each region or business unit, then create fleets within each folder. Folders can contain other folders or fleets. When you assign a role at the folder level, members inherit that role for all fleets within the folder.
- **Fleets** - Group the storage systems you manage together. You associate storage systems with fleets and assign members to fleets to grant them access.

Folders are an organizational tool and are not visible to members who do not have IAM permissions such as Organization admin, Folder admin, or Fleet admin roles. Members access fleets, not folders.

Common fleet organization patterns

How you organize fleets depends on your operational model:

- **By environment** - Create separate fleets for production, development, and test workloads. This keeps production storage under tighter policies while allowing more flexibility in lower environments.
- **By business unit** - Group clusters that serve a specific department, such as finance, engineering, or HR, into a dedicated fleet. You can then assign storage management responsibilities to team members within that business unit without exposing other fleets to them.
- **By location or data center** - When clusters are distributed across sites, organizing by location lets you monitor site-level health, apply region-specific policies, and track capacity consumption per site.
- **By application tier** - Group clusters that host a specific class of workload, such as databases, file shares, or virtual machines, so you can apply consistent standards tuned to that workload type across the entire fleet.

Use folders to add another level of organization. For example, create a folder for each region and then create environment-based fleets within each regional folder.

How fleet management enables access control

Fleets and folders serve as boundaries for user access and administrative responsibility:

- **Folder-level access** - When you assign a role at the folder level, the member inherits that role for all fleets and resources within the folder. This lets you delegate administrative responsibilities without granting access to the entire organization.
- **Fleet-level access** - When you assign a role at the fleet level, the member has access only to the storage systems within that fleet. This lets you delegate storage management to team members or application owners without exposing unrelated parts of your infrastructure.

Console local deployment provides fleet-level health and performance monitoring so you can see the status of all clusters in a fleet from a single view, identify issues early, and act before they affect workloads.

How storage management works

Storage management is the practice of defining storage standards, applying them consistently, and operating workloads so they stay aligned over time. In Console local deployment, those standards are expressed as storage class policies and storage classes, scoped to fleets, and enforced through provisioning workflows governed by RBAC and audit logging.

Console local deployment connects four concepts into one workflow:

- **Fleets** define the scope where you manage storage. A fleet groups systems so you can control access, apply standards consistently, and manage resources as a unit.
- **Storage class policies** define standards as reusable building blocks (performance, capacity, security, data protection).
- **Storage classes** express workload intent. A storage class is a named template assembled from one or more policies.
- **Provisioning workflows** create storage within guardrails. Different workflows make configuration decisions differently, but all can enforce storage classes and remain governed by RBAC and audit logging.

Provisioning approaches

Console local deployment supports three provisioning approaches, all governed by RBAC, audit logging, and storage class standards:

- **Manual provisioning** - You select the target system and specify configuration details directly. Use this when you need explicit control over system selection and configuration.
- **Automated provisioning** - You provide workload inputs and optionally select a storage class. Console local deployment recommends best-fit placement and applies class-driven settings to reduce manual decisions.
- **AI-assisted (agentic) provisioning** - You describe what you need in natural language. Console local deployment proposes a plan constrained by RBAC and storage classes, then provisions only after you review and approve.

Where to go next

- To understand storage standards, see Learn about storage classes and policies in NetApp Console local deployment.
- To create and manage fleets, see Manage folders and fleets.
- Provision storage manually
- Provision storage automatically
- Provision storage with AI assistance

Standardize storage behavior

Learn about storage classes and policies in NetApp Console local deployment

A storage class is a reusable template that defines how storage should behave. When you provision storage using a storage class, NetApp Console local deployment enforces the standards encoded in that class automatically without requiring administrators to make individual configuration decisions for every workload.

Storage classes are built from storage class policies, which define individual dimensions of storage behavior. Together, they let you capture your organization's storage standards once and apply them consistently across fleets.

What storage class policies are

A storage class policy defines one dimension of storage behavior. Policies are reusable building blocks that you combine to create storage classes.

Common policy types include:

- **Performance policies** - Define latency targets, IOPS, or throughput requirements.
- **Capacity policies** - Define provisioning mode, threshold alerts, or growth limits.
- **Security policies** - Define encryption, compliance, or access control requirements.
- **Data protection policies** - Define snapshot schedules, replication targets, or retention periods.

You define policies independently, then combine them when you build a storage class. This lets you reuse the same performance policy across multiple classes, or update a capacity policy in one place and apply that change to every class that uses it.

What storage classes are

A storage class is a named template assembled from one or more policies. It captures your organization's storage standards for a specific type of workload.

For example, you might create a **Production Database** storage class that combines high-performance, high-availability, and strict data protection policies, or a **Development File Share** storage class that combines moderate performance, flexible capacity, and basic data protection policies.

Storage administrators define storage classes to encode enterprise requirements. All provisioning activity, whether done manually, through guided workflows, or with automation, draws from the library of classes those administrators have approved.

How storage classes enforce standards

When you provision storage, you select a storage class rather than configuring individual settings. Console local deployment uses the policies in that class to identify which clusters in your fleet have the capacity and performance headroom to support the workload, recommend the best placement option, and apply class-driven settings automatically at provisioning time.

After provisioning, Console local deployment continuously evaluates each workload against its storage class standards.

Storage class drift and compliance

When a workload no longer matches its storage class intent, Console local deployment flags it for investigation and remediation.

Issues fall into two categories:

- **Configuration drift**: Storage settings governed by the storage class policy no longer match the intended configuration. This can occur when changes are made directly on the ONTAP cluster or outside standard provisioning workflows.
- **Performance non-compliance**: The workload's runtime behavior no longer meets the storage class performance intent (for example, sustained pressure near a QoS limit or elevated tail latency).

An analysis view explains what changed and why. Guided remediation actions (for example, **Fix it**) may be available to help restore compliance. Some remediations can be applied in place, while others may require aligning the workload to a different storage class to restore the intended behavior.

Where to investigate

You can typically investigate drift and non-compliance from one of these locations (availability depends on your deployment and permissions):

- **Storage classes:** Drift / Compliance
- **Alerts:** Analyze

For step-by-step instructions, see Remediate storage class drift.

End-to-end workflow

The following steps describe the process for using storage classes to standardize and govern storage in your environment:

1. **Create storage class policies** - Policies define the individual dimensions of storage behavior (performance targets, capacity settings, security requirements, data protection schedules). Create policies before building a storage class.
2. **Create a storage class** - Assemble a storage class by combining one policy from each applicable category. You can use a predefined storage class or create a custom one for your organization's standards.
3. **Associate the storage class with a fleet** - After you create a storage class, associate it with the fleet where it will be used for provisioning and compliance monitoring.
4. **Provision storage using the storage class** - When provisioning a volume or LUN, select a storage class instead of configuring individual settings. Console local deployment uses the class to recommend the best-fit cluster placement, then provisions with the settings defined in the class.
5. **Monitor workloads for drift** - Console local deployment continuously evaluates each workload against the standards encoded in its storage class. If configuration drift or performance non-compliance occurs, it flags the issue and may raise an alert.
6. **Remediate drift to restore compliance** - When drift or performance non-compliance is detected, you can follow guided steps to correct the issue manually, let Console local deployment resolve common drift conditions automatically, or disassociate a workload from its storage class if you need to change its settings.

How storage classes work with fleets

Storage classes are associated with fleets. When you associate a storage class with a fleet, that class becomes available for all provisioning within the fleet. This ensures that every workload provisioned in the fleet follows the same standards, making fleets the primary way to enforce consistent storage behavior across related clusters.

For details on how to organize fleets, see Learn about fleet management in NetApp Console local deployment.

Where to go next

- To understand fleet organization, see Learn about fleet management in NetApp Console local deployment.
- To create policies and storage classes, see Manage storage classes and policies.
- Provision storage manually
- Provision storage automatically
- Provision storage with AI assistance
- To analyze and remediate drift, see Remediate storage class drift

Understanding storage class policies in NetApp Console local deployment

Storage class policies are the building blocks you use to construct a storage class. Each policy controls a specific aspect of storage behavior. When you combine policies into a storage class, you create a reusable template that the NetApp Console local deployment enforces automatically at provisioning time and monitors continuously throughout a workload's lifecycle.

Policies as building blocks

A storage class is composed of one or more policies, each governing a different dimension of storage behavior. Storage administrators define policies to encode enterprise standards—performance expectations, capacity thresholds, data placement rules—and then assemble those policies into storage class templates. When a workload is provisioned using a storage class, it inherits all of the policies in that class. This design separates the responsibility of defining standards from the act of provisioning, so storage can be provisioned consistently by automated workflows without needing to make individual configuration decisions.

Types of storage class policies

NetApp Console local deployment includes four types of policies that you can use to construct storage classes:

Performance policy

A performance policy sets expected IOPS/TB, peak IOPS/TB, absolute minimum IOPS, and expected latency targets for a workload. The Console local deployment uses these values to recommend clusters with sufficient headroom at provisioning time and to detect IOPS or latency drift after provisioning.

Capacity policy

A capacity policy controls how a volume consumes and manages space. It sets space reservation (thick, thin, or system default), autogrow behavior, FabricPool tiering, and whether to provision NAS workloads as FlexVol or FlexGroup volumes.

Security policy

A security policy sets the encryption, ransomware protection, and FIPS requirements for a workload. Each attribute can be set to a required value or left as "any" to accept the system default.

Data protection policy

A data protection policy sets the number of snapshots retained at each interval to ensure workloads using the policy get a consistent backup schedule.

Default policies

The Console local deployment includes system-defined policies across all four policy types. You can use these policies as-is when you create a storage class, or copy them as a starting point for custom policies tailored to your organization’s requirements.

Performance policies

Name	Type	Expected IOPS/TB	Peak IOPS/TB	Absolute min IOPS	Expected latency (ms)	Summary
Extreme for Database Data	System defined	12,288	24,576	2,000	1	Use for transactional database data volumes requiring sustained high IOPS and sub-millisecond latency.
Extreme for Database Logs	System defined	22,528	45,056	4,000	1	Use for database transaction log volumes requiring the highest IOPS floor to prevent write bottlenecks.

Name	Type	Expected IOPS/TB	Peak IOPS/TB	Absolute min IOPS	Expected latency (ms)	Summary
Extreme for Database Shared Data	System defined	16,384	32,768	2,000	1	Use for shared database volumes accessed by multiple hosts requiring high throughput and consistent latency.
Extreme Performance	System defined	6,144	12,288	1,000	1	Use for HPC, AI/ML, and analytics workloads requiring high burst IOPS and predictable low latency.
Performance	System defined	2,048	4,096	500	2	Use for virtualized and enterprise applications requiring reliable performance without extreme IOPS demands.
Value	System defined	128	512	75	17	Use for cost-sensitive workloads such as home directories, file shares, and archives.

Capacity policies

Name	Type	Space reserve	Autogrow mode	Tiering policy	Volume type (NAS)	Summary
default	System defined	System default	System default	none	System default	Use for general workloads where platform default capacity behavior is acceptable.
production	System defined	System default	grow	none	System default	Use for production workloads that must expand automatically to prevent out-of-space failures.

Security policies

Name	Type	Encryption	Ransomware protection	FIPS	Summary
default	System defined	any	any	any	Use for workloads where platform default security behavior is acceptable.
anti-ransomware	System defined	any	on	any	Use for workloads containing sensitive or business-critical data that require active ransomware protection.
production	System defined	enabled	any	any	Use for production workloads where encryption is required for compliance.

Data protection policies

Name	Type	Hourly snapshots	Daily snapshots	Weekly snapshots	Monthly snapshots	Summary
default	System defined	6	2	2	0	Use for standard workloads requiring short-term recovery points without long-term retention overhead.
3month-hourly	System defined	23	6	4	3	Use for regulated or business-critical workloads requiring granular intraday recovery points and three months of historical retention.

Review predefined storage class policies in NetApp Console local deployment

NetApp Console local deployment includes system-defined policies across all four policy types. Use this reference to identify which policy best fits your workload requirements when creating or customizing a storage class.

Performance policies

Extreme Performance

Use for HPC, AI/ML, and analytics workloads requiring high burst IOPS and predictable low latency.

Extreme for Database Data

Use for transactional database data volumes requiring sustained high IOPS and sub-millisecond latency.

Extreme for Database Logs

Use for database transaction log volumes requiring the highest IOPS floor to prevent write bottlenecks.

Extreme for Database Shared Data

Use for shared database volumes accessed by multiple hosts requiring high throughput and consistent latency.

Performance

Use for virtualized and enterprise applications requiring reliable performance without extreme IOPS demands.

Value

Use for cost-sensitive workloads such as home directories, file shares, and archives.

Capacity policies

Autogrow capacity

Use for production workloads that must expand automatically to prevent out-of-space failures.

Security policies

Encryption at-rest
Use for production workloads where encryption is required for compliance.

Ransomware protection
Use for workloads containing sensitive or business-critical data that require active ransomware protection.

Standard security
Use for workloads where platform default security behavior is acceptable.

Data protection policies

default
Use for standard workloads requiring short-term recovery points without long-term retention overhead.

3month-hourly
Use for regulated or business-critical workloads requiring granular intra-day recovery points and three months of historical retention.

30-day recovery
Use for standard workloads requiring short-term recovery points without long-term retention overhead.

90-day recovery
Use for regulated or business-critical workloads requiring granular intra-day recovery points and three months of historical retention.

Create storage classes in NetApp Console local deployment

Create a storage class in the NetApp Console local deployment to standardize how storage is provisioned and governed across your fleets. A storage class is a reusable template that bundles storage class policies—such as performance targets, capacity behavior, security requirements, and data protection schedules—into a single named definition.

When users (or automation) provision a volume or LUN using a storage class, NetApp Console local deployment applies the class's policies automatically. After provisioning, Console local deployment continuously monitors workloads against the storage class to detect drift and can guide or automate remediation to restore compliance.

Before you begin

- Discover at least one ONTAP cluster so Console local deployment has targets for placement recommendations.
- Ensure you have the Organization admin role (or a role that grants storage class management permissions in your environment).
- Create (or identify) the storage class policies you plan to use—performance, capacity, security, and data protection—because storage classes are assembled from policies.

Create a storage class

You must have the Organization admin or Folder or fleet admin role to add a storage class.

Steps

1. Select **Storage > Management**.
2. Select **Storage classes**.
3. Select **Add**.
4. Under **Basic information**, enter the following:
   - **Storage class name**: Enter a unique name for the storage class.
   - **Description**: (Optional) Enter a description of the storage class.
   - **Recommended apps**: (Optional) Enter a list of recommended apps for the storage class.
5. Under **Storage policies**, select one or more policies to associate with the storage class.
6. Select **Add**.

Customize an existing storage class

You must have the Organization admin or Folder or fleet admin role to customize an existing storage class.

Steps

1. Select **Storage > Management**.
2. Select **Storage classes**.
3. For the storage class you want to customize, select **…**, and then select **Duplicate**.
4. Under **Basic information**, update the following as required:
   - **Storage class name**: Enter a unique name for the storage class.
   - **Description**: (Optional) Enter a description of the storage class.
   - **Recommended apps**: (Optional) Enter a list of recommended apps for the storage class.
5. Under **Storage policies**, select one or more policies to associate with the storage class.
6. Select **Add**.

Edit a user-defined storage class

You must have the Organization admin or Folder or fleet admin role to edit a user-defined storage class.

Steps

1. Select **Storage > Management**.
2. Select **Storage classes**.
3. For the storage class you want to edit, select **…**, and then select **Edit**.
4. Under **Basic information**, edit the following as required:
   - **Storage class name**: Enter a unique name for the storage class.
   - **Description**: (Optional) Enter a description of the storage class.
   - **Recommended apps**: (Optional) Enter a list of recommended apps for the storage class.
5. Under **Storage policies**, select one or more policies to associate with the storage class.

6. Select **Edit**.

Delete a user-defined storage class

You must have the Organization admin or Folder or fleet admin role to delete a user-defined storage class.

Steps

1. Select **Storage > Management**.
2. Select **Storage classes**.
3. For the storage class you want to delete, select **…**, and then select **Delete**.
4. Select **Delete**.

   Note that this action cannot be undone. If you delete a storage class that is currently in use, any volumes or LUNs provisioned with that class will continue to operate but will no longer be associated with the deleted class.

Provision storage in NetApp Console local deployment

Provision volumes and LUNs in NetApp Console local deployment to make storage resources available for your workloads. During provisioning, you can optionally select a storage class to apply predefined storage policies and organizational standards.

Required access roles

Super admin, Organization admin, Folder or fleet admin, or Storage admin. Learn about access roles.

Provision a volume

Steps

1. Select **Storage > Management**.
2. Select **Fleets**.
3. Select the fleet you want to provision into.
4. Select **Add**.
5. From the menu, select **Volume**.
6. Under **Volume details**:
   a. Enter a volume name.
   b. Enter capacity (and choose units if needed).
   c. (Optional) Select a storage class to apply storage class policies. If you don’t select one, the volume is provisioned without storage class policies.
7. Under **System details**:
   a. Select a system from the eligible systems for the selected fleet.
   b. Select a storage VM.
8. Under **Host (NAS) details**, choose how hosts access the volume:
   a. Export by NFS (export policies control which NFS hosts can access the volume)
   b. Share by SMB/CIFS

9. Select **Add**.

Provision a LUN

Steps

1. Select **Storage > Management**.
2. Select **Fleets**.
3. Select the fleet you want to provision into.
4. Select **Add**.
5. From the menu, select **LUNs**.
6. Under **LUN storage details**:
   a. Enter a prefix name.
   b. Enter the number of LUNs to create with that prefix.
   c. Enter the capacity per LUN (and choose units if needed).
   d. (Optional) Select a storage class to apply storage class policies. If you don’t select one, the LUNs are provisioned without storage class policies.
7. Under **System details**:
   a. Select a system (if prompted).
   b. Select a storage VM.
8. Under **Host (SAN) details**, choose how hosts access the volume:
   a. Choose the host operation system (for example, Windows or Linux) to set recommended defaults for LUN alignment and block size.
   b. Choose the host mapping group to control which initiators can access the LUNs.
9. Select **Add**.

Copyright information

Trademark information