Use FPolicy for file monitoring and management on SVMs

ONTAP 9

NetApp
March 08, 2024

Table of Contents

Use FPolicy for file monitoring and management on SVMs

Understand FPolicy

What the two parts of the FPolicy solution are

FPolicy is a file access notification framework that is used to monitor and manage file access events on storage virtual machines (SVMs) through partner solutions. Partner solutions help you address various use cases such as data governance and compliance, ransomware protection, and data mobility.

Partner solutions include both Netapp supported 3rd party Solutions and NetApp products Workload Security and Cloud Data Sense.

There are two parts to an FPolicy solution. The ONTAP FPolicy framework manages activities on the cluster and sends notifications to Partner Application (aka External FPolicy Servers). External FPolicy servers process notifications sent by ONTAP FPolicy to fulfill customer use cases.

The ONTAP framework creates and maintains the FPolicy configuration, monitors file events, and sends notifications to external FPolicy servers. ONTAP FPolicy provides the infrastructure that allows communication between external FPolicy servers and storage virtual machine (SVM) nodes.

The FPolicy framework connects to external FPolicy servers and sends notifications for certain file system events to the FPolicy servers when these events occur as a result of client access. The external FPolicy servers process the notifications and send responses back to the node. What happens as a result of the notification processing depends on the application and whether the communication between the node and the external servers is asynchronous or synchronous.

What synchronous and asynchronous notifications are

FPolicy sends notifications to external FPolicy servers via the FPolicy interface. The notifications are sent either in synchronous or asynchronous mode. The notification mode determines what ONTAP does after sending notifications to FPolicy servers.

- **Asynchronous notifications**

  With asynchronous notifications, the node does not wait for a response from the FPolicy server, which enhances overall throughput of the system. This type of notification is suitable for applications where the FPolicy server does not require that any action be taken as a result of notification evaluation. For example, asynchronous notifications are used when the storage virtual machine (SVM) administrator wants to monitor and audit file access activity.

  If an FPolicy server operating in asynchronous mode experiences a network outage, FPolicy notifications generated during the outage are stored on the storage node. When the FPolicy server comes back online, it is alerted of the stored notifications and can fetch them from the storage node. The length of time the notifications can be stored during an outage is configurable up to 10 minutes.

  Beginning with ONTAP 9.14.1, FPolicy allows you to set up a persistent store to capture file access events for asynchronous non-mandatory policies in the SVM. Persistent stores can help decouple client I/O

processing from FPolicy notification processing to reduce client latency. Synchronous (mandatory or non-mandatory) and asynchronous mandatory configurations are not supported.

- **Synchronous notifications**

  When configured to run in synchronous mode, the FPolicy server must acknowledge every notification before the client operation is allowed to continue. This type of notification is used when an action is required based on the results of notification evaluation. For example, synchronous notifications are used when the SVM administrator wants to either allow or deny requests based on criteria specified on the external FPolicy server.

Synchronous and asynchronous applications

There are many possible uses for FPolicy applications, both asynchronous and synchronous.

Asynchronous applications are ones where the external FPolicy server does not alter access to files or directories or modify data on the storage virtual machine (SVM). For example:

- File access and audit logging
- Storage resource management

Synchronous applications are ones where data access is altered or data is modified by the external FPolicy server. For example:

- Quota management
- File access blocking
- File archiving and hierarchical storage management
- Encryption and decryption services
- Compression and decompression services

FPolicy persistent stores

Beginning with ONTAP 9.14.1, FPolicy allows you to set up a persistent store to capture file access events for asynchronous non-mandatory policies in the SVM. Persistent stores can help decouple client I/O processing from FPolicy notification processing to reduce client latency. Synchronous (mandatory or non-mandatory) and asynchronous mandatory configurations are not supported.

This feature is only available in FPolicy external mode. The partner application you use needs to support this feature. You should work with your partner to ensure this FPolicy configuration is supported.

Best practices

Cluster administrators need to configure a volume for the persistent store on each SVM where FPolicy is enabled. When configured, a persistent store captures all matching FPolicy events, which are further processed in the FPolicy pipeline and sent to the external server.

The persistent store remains as it was when the last event was received when there is an unexpected reboot or FPolicy is disabled and enabled again. After a takeover operation, new events will be stored and processed by the partner node. After a giveback operation, the persistent store resumes processing any unprocessed events that might remain from when the node takeover occurred. Live events would be given priority over

unprocessed events.

If the persistent store volume moves from one node to another in the same SVM, the notifications that are yet to be processed will also move to the new node. You will need to re-run the `fpolicy persistent-store create` command on either node after the volume is moved to ensure the pending notification are delivered to the external server.

The persistent store volume is setup on a per SVM basis. For each FPolicy enabled SVM you will need to create a persistent store volume.

Create the persistent store volume on the node with LIFs that expect maximum traffic to be monitored by Fpolicy.

If the notifications accumulated in the persistent store exceed the size of the volume provisioned, FPolicy will start dropping the incoming notification with appropriate EMS messages.

The persistent Store volume name and the junction-path specified at the time of volume creation should match.

Have the snapshot policy set to `none` for that volume instead of `default`. This is to ensure that there is no accidental restore of the snapshot leading to loss of current events and to prevent possible duplicate event processing.

Make the persistent store volume inaccessible for external user protocol access (CIFS/NFS) to avoid accidental corruption or deletion of the persisted event records. To achieve this, after enabling FPolicy, unmount the volume in ONTAP to remove the junction path, this makes it inaccessible for the user protocol access.

For more information, see Create persistent stores.

FPolicy configuration types

There are two basic FPolicy configuration types. One configuration uses external FPolicy servers to process and act upon notifications. The other configuration does not use external FPolicy servers; instead, it uses the ONTAP internal, native FPolicy server for simple file blocking based on extensions.

- **External FPolicy server configuration**

  The notification is sent to the FPolicy server, which screens the request and applies rules to determine whether the node should allow the requested file operation. For synchronous policies, the FPolicy server then sends a response to the node to either allow or block the requested file operation.

- **Native FPolicy server configuration**

  The notification is screened internally. The request is allowed or denied based on file extension settings configured in the FPolicy scope.

  Note: File extension requests that are denied are not logged.

When to create a native FPolicy configuration

Native FPolicy configurations use the ONTAP internal FPolicy engine to monitor and block file operations based on the file's extension. This solution does not require external FPolicy servers (FPolicy servers). Using a native file blocking configuration is appropriate when this simple solution is all that is needed.

Native file blocking enables you to monitor any file operations that match configured operation and filtering events and then deny access to files with particular extensions. This is the default configuration.

This configuration provides a means to block file access based only on the file's extension. For example, to block files that contain `mp3` extensions, you configure a policy to provide notifications for certain operations with target file extensions of `mp3`. The policy is configured to deny `mp3` file requests for operations that generate notifications.

The following applies to native FPolicy configurations:

- The same set of filters and protocols that are supported by FPolicy server-based file screening are also supported for native file blocking.

- Native file blocking and FPolicy server-based file screening applications can be configured at the same time.

  To do so, you can configure two separate FPolicy policies for the storage virtual machine (SVM), with one configured for native file blocking and one configured for FPolicy server-based file screening.

- The native file blocking feature only screens files based on the extensions and not on the content of the file.

- In the case of symbolic links, native file blocking uses the file extension of the root file.

Learn more about FPolicy: Native File Blocking.

When to create a configuration that uses external FPolicy servers

FPolicy configurations that use external FPolicy servers to process and manage notifications provide robust solutions for use cases where more than simple file blocking based on file extension is needed.

You should create a configuration that uses external FPolicy servers when you want to do such things as monitor and record file access events, provide quota services, perform file blocking based on criteria other than simple file extensions, provide data migration services using hierarchical storage management applications, or provide a fine-grained set of policies that monitor only a subset of data in the storage virtual machine (SVM).

Roles that cluster components play with FPolicy implementation

The cluster, the contained storage virtual machines (SVMs), and data LIFs all play a role in an FPolicy implementation.

- **cluster**

  The cluster contains the FPolicy management framework and maintains and manages information about all FPolicy configurations in the cluster.

- **SVM**

  An FPolicy configuration is defined at the SVM level. The scope of the configuration is the SVM, and it only operates on SVM resources. One SVM configuration cannot monitor and send notifications for file access requests that are made for data residing on another SVM.

  FPolicy configurations can be defined on the admin SVM. After configurations are defined on the admin SVM, they can be seen and used in all SVMs.

- **data LIFs**

  Connections to the FPolicy servers are made through data LIFs belonging to the SVM with the FPolicy configuration. The data LIFs used for these connections can fail over in the same manner as data LIFs used for normal client access.

How FPolicy works with external FPolicy servers

After FPolicy is configured and enabled on the storage virtual machine (SVM), FPolicy runs on every node on which the SVM participates. FPolicy is responsible for establishing and maintaining connections with external FPolicy servers (FPolicy servers), for notification processing, and for managing notification messages to and from FPolicy servers.

Additionally, as part of connection management, FPolicy has the following responsibilities:

- Ensures that file notification flows through the correct LIF to the FPolicy server.
- Ensures that when multiple FPolicy servers are associated with a policy, load balancing is done when sending notifications to the FPolicy servers.
- Attempts to reestablish the connection when a connection to an FPolicy server is broken.
- Sends the notifications to FPolicy servers over an authenticated session.
- Manages the passthrough-read data connection established by the FPolicy server for servicing client requests when passthrough-read is enabled.

How control channels are used for FPolicy communication

FPolicy initiates a control channel connection to an external FPolicy server from the data LIFs of each node participating on a storage virtual machine (SVM). FPolicy uses control channels for transmitting file notifications; therefore, an FPolicy server might see multiple control channel connections based on SVM topology.

How privileged data access channels are used for synchronous communication

With synchronous use cases, the FPolicy server accesses data residing on the storage virtual machine (SVM) through a privileged data access path. Access through the privileged path exposes the complete file system to the FPolicy server. It can access data files to collect information, to scan files, read files, or write into files.

Because the external FPolicy server can access the entire file system from the root of the SVM through the privileged data channel, the privileged data channel connection must be secure.

How FPolicy connection credentials are used with privileged data access channels

The FPolicy server makes privileged data access connections to cluster nodes by using a specific Windows user credential that is saved with the FPolicy configuration. SMB is the only supported protocol for making a privileged data access channel connection.

If the FPolicy server requires privileged data access, the following conditions must be met:

- A SMB license must be enabled on the cluster.
- The FPolicy server must run under the credentials configured in the FPolicy configuration.

When making a data channel connection, FPolicy uses the credential for the specified Windows user name. Data access is made over the admin share ONTAP_ADMIN$.

What granting super user credentials for privileged data access means

ONTAP uses the combination of the IP address and the user credential configured in the FPolicy configuration to grant super user credentials to the FPolicy server.

Super user status grants the following privileges when the FPolicy server accesses data:

- Avoid permission checks

  The user avoids checks on files and directory access.

- Special locking privileges

  ONTAP allows read, write, or modify access to any file regardless of existing locks. If the FPolicy server takes byte range locks on the file, it results in immediate removal of existing locks on the file.

- Bypass any FPolicy checks

  Access does not generate any FPolicy notifications.

How FPolicy manages policy processing

There might be multiple FPolicy policies assigned to your storage virtual machine (SVM); each with a different priority. To create an appropriate FPolicy configuration on the SVM, it is important to understand how FPolicy manages policy processing.

Each file access request is initially evaluated to determine which policies are monitoring this event. If it is a monitored event, information about the monitored event along with interested policies is passed to FPolicy where it is evaluated. Each policy is evaluated in order of the assigned priority.

You should consider the following recommendations when configuring policies:

- When you want a policy to always be evaluated before other policies, configure that policy with a higher priority.
- If the success of requested file access operation on a monitored event is a prerequisite for a file request that is evaluated against another policy, give the policy that controls the success or failure of the first file operation a higher priority.

  For example, if one policy manages FPolicy file archiving and restore functionality and a second policy manages file access operations on the online file, the policy that manages file restoration must have a higher priority so that the file is restored before the operation managed by the second policy can be allowed.

- If you want all policies that might apply to a file access operation to be evaluated, give synchronous policies a lower priority.

You can reorder policy priorities for existing policies by modifying the policy sequence number. However, to have FPolicy evaluate policies based on the modified priority order, you must disable and reenable the policy with the modified sequence number.

What the node-to-external FPolicy server communication process is
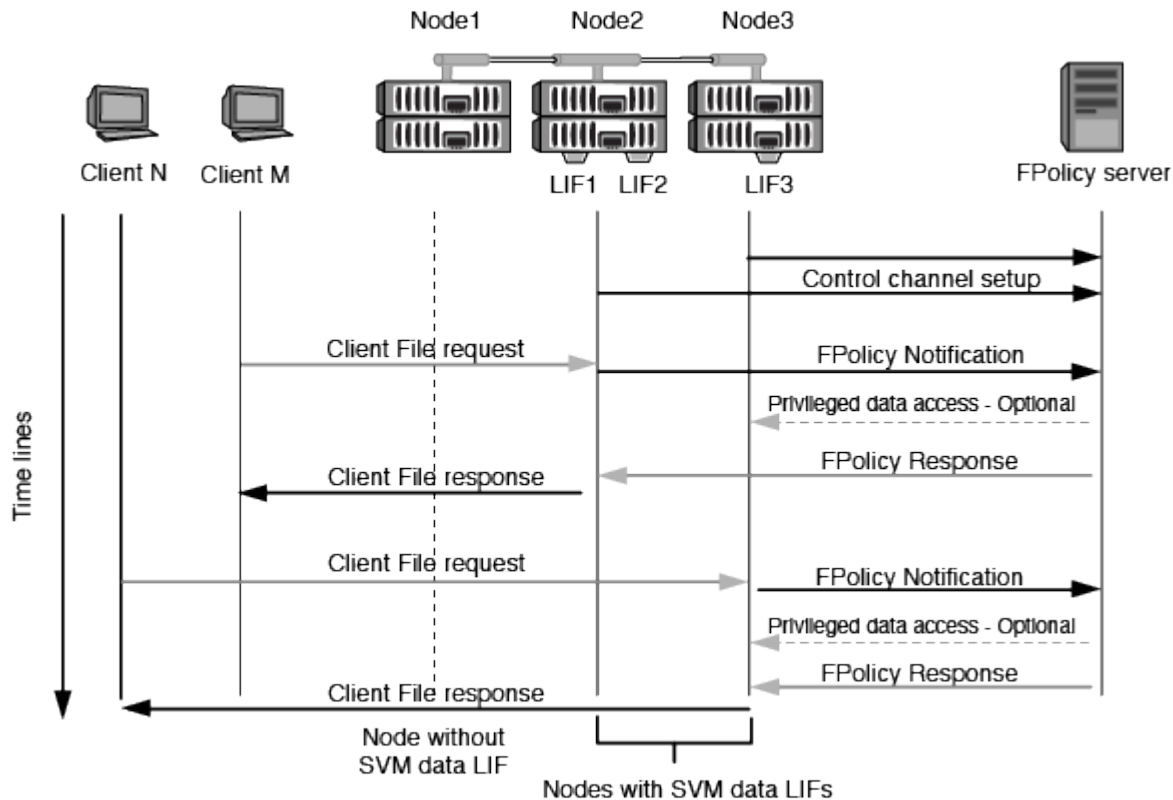
To properly plan your FPolicy configuration, you should understand what the node-to-external FPolicy server communication process is.

Every node that participates on each storage virtual machine (SVM) initiates a connection to an external FPolicy server (FPolicy server) using TCP/IP. Connections to the FPolicy servers are set up using node data LIFs; therefore, a participating node can set up a connection only if the node has an operational data LIF for the SVM.

Each FPolicy process on participating nodes attempts to establish a connection with the FPolicy server when the policy is enabled. It uses the IP address and port of the FPolicy external engine specified in the policy configuration.

The connection establishes a control channel from each of the nodes participating on each SVM to the FPolicy server through the data LIF. In addition, if IPv4 and IPv6 data LIF addresses are present on the same participating node, FPolicy attempts to establish connections for both IPv4 and IPv6. Therefore, in a scenario where the SVM extends over multiple nodes or if both IPv4 and IPv6 addresses are present, the FPolicy server must be ready for multiple control channel setup requests from the cluster after the FPolicy policy is enabled on the SVM.

For example, if a cluster has three nodes—Node1, Node2, and Node3—and SVM data LIFs are spread across only Node2 and Node3, control channels are initiated only from Node2 and Node3, irrespective of the distribution of data volumes. Say that Node2 has two data LIFs—LIF1 and LIF2—that belong to the SVM and that the initial connection is from LIF1. If LIF1 fails, FPolicy attempts to establish a control channel from LIF2.

How FPolicy manages external communication during LIF migration or failover

Data LIFs can be migrated to data ports in the same node or to data ports on a remote node.

When a data LIF fails over or is migrated, a new control channel connection is made to the FPolicy server. FPolicy can then retry SMB and NFS client requests that timed out, with the result that new notifications are sent to the external FPolicy servers. The node rejects FPolicy server responses to original, timed-out SMB and NFS requests.

How FPolicy manages external communication during node failover

If the cluster node that hosts the data ports used for FPolicy communication fails, ONTAP breaks the connection between the FPolicy server and the node.

The impact of cluster failover to the FPolicy server can be mitigated by configuring the failover-policy to migrate the data port used in FPolicy communication to another active node. After the migration is complete, a new connection is established using the new data port.

If the failover-policy is not configured to migrate the data port, the FPolicy server must wait for the failed node to come up. After the node is up, a new connection is initiated from that node with a new Session ID.

> (i) The FPolicy server detects broken connections with the keep-alive protocol message. The timeout for purging the session ID is determined when configuring FPolicy. The default keep-alive timeout is two minutes.

How FPolicy services work across SVM namespaces

ONTAP provides a unified storage virtual machine (SVM) namespace. Volumes across the cluster are joined together by junctions to provide a single, logical file system. The FPolicy server is aware of the namespace topology and provides FPolicy services across the namespace.

The namespace is specific to and contained within the SVM; therefore, you can see the namespace only from the SVM context. Namespaces have the following characteristics:

- A single namespace exists in each SVM, with the root of the namespace being the root volume, represented in the namespace as slash (/).
- All other volumes have junction points below the root (/).
- Volume junctions are transparent to clients.
- A single NFS export can provide access to the complete namespace; otherwise, export policies can export specific volumes.
- SMB shares can be created on the volume or on qtrees within the volume, or on any directory within the namespace.
- The namespace architecture is flexible.

  Examples of typical namespace architectures are as follows:

    ◦ A namespace with a single branch off of the root
    ◦ A namespace with multiple branches off of the root
    ◦ A namespace with multiple unbranched volumes off of the root

How FPolicy passthrough-read enhances usability for hierarchical storage management

Passthrough-read enables the FPolicy server (functioning as the hierarchical storage management (HSM) server) to provide read access to offline files without having to recall the file from the secondary storage system to the primary storage system.

When an FPolicy server is configured to provide HSM to files residing on a SMB server, policy-based file migration occurs where the files are stored offline on secondary storage and only a stub file remains on primary storage. Even though a stub file appears as a normal file to clients, it is actually a sparse file that is the same size of the original file. The sparse file has the SMB offline bit set and points to the actual file that has been migrated to secondary storage.

Typically when a read request for an offline file is received, the requested content must be recalled back to primary storage and then accessed through primary storage. The need to recall data back to primary storage has several undesirable effects. Among the undesirable effects is the increased latency to client requests caused by the need to recall the content before responding to the request and the increased space consumption needed for recalled files on the primary storage.

FPolicy passthrough-read allows the HSM server (the FPolicy server) to provide read access to migrated, offline files without having to recall the file from the secondary storage system to the primary storage system. Instead of recalling the files back to primary storage, read requests can be serviced directly from secondary storage.

> ⓘ Copy Offload (ODX) is not supported with FPolicy passthrough-read operation.

Passthrough-read enhances usability by providing the following benefits:

- Read requests can be serviced even if the primary storage does not have sufficient space to recall requested data back to primary storage.

- Better capacity and performance management when a surge of data recall might occur, such as if a script or a backup solution needs to access many offline files.

- Read requests for offline files in Snapshot copies can be serviced.

  Because Snapshot copies are read-only, the FPolicy server cannot restore the original file if the stub file is located in a Snapshot copy. Using passthrough-read eliminates this problem.

- Policies can be set up that control when read requests are serviced through access to the file on secondary storage and when the offline file should be recalled to primary storage.

  For example, a policy can be created on the HSM server that specifies the number of times the offline file can be accessed in a specified period of time before the file is migrated back to primary storage. This type of policy avoids recalling files that are rarely accessed.

How read requests are managed when FPolicy passthrough-read is enabled

You should understand how read requests are managed when FPolicy passthrough-read is enabled so that you can optimally configure connectivity between the storage virtual machine (SVM) and the FPolicy servers.

When FPolicy passthrough-read is enabled and the SVM receives a request for an offline file, FPolicy sends a notification to the FPolicy server (HSM server) through the standard connection channel.

After receiving the notification, the FPolicy server reads the data from the file path sent in the notification and sends the requested data to the SVM through the passthrough-read privileged data connection that is established between the SVM and the FPolicy server.

After the data is sent, the FPolicy server then responds to the read request as an ALLOW or DENY. Based on whether the read request is allowed or denied, ONTAP either sends the requested information or sends an error message to the client.

Plan the FPolicy configuration

Requirements, considerations, and best practices for configuring FPolicy

Before you create and configure FPolicy configurations on your SVMs, you need to be aware of certain requirements, considerations, and best practices for configuring FPolicy.

FPolicy features are configured either through the command line interface (CLI) or through REST APIs.

Requirements for setting up FPolicy

Before you configure and enable FPolicy on your storage virtual machine (SVM), you need to be aware of certain requirements.

- All nodes in the cluster must be running a version of ONTAP that supports FPolicy.
- If you are not using the ONTAP native FPolicy engine, you must have external FPolicy servers (FPolicy servers) installed.
- The FPolicy servers must be installed on a server accessible from the data LIFs of the SVM where FPolicy policies are enabled.

  > (i) Beginning with ONTAP 9.8, ONTAP provides a client LIF service for outbound FPolicy connections with the addition of the `data-fpolicy-client` service. Learn more about LIFs and service policies.

- The IP address of the FPolicy server must be configured as a primary or secondary server in the FPolicy policy external engine configuration.
- If the FPolicy servers access data over a privileged data channel, the following additional requirements must be met:
  - SMB must be licensed on the cluster.

    Privileged data access is accomplished using SMB connections.

  - A user credential must be configured for accessing files over the privileged data channel.
  - The FPolicy server must run under the credentials configured in the FPolicy configuration.
  - All data LIFs used to communicate with the FPolicy servers must be configured to have `cifs` as one of the allowed protocols.

    This includes the LIFs used for passthrough-read connections.

- Beginning with ONTAP 9.14.1, FPolicy allows you to set up a persistent store to capture file access events for asynchronous non-mandatory policies in the SVM. Persistent stores can help decouple client I/O processing from FPolicy notification processing to reduce client latency. Synchronous (mandatory or non-

mandatory) and asynchronous mandatory configurations are not supported.

Best practices and recommendations when setting up FPolicy

When setting up FPolicy on storage virtual machines (SVMs), get familiar with the general configuration best practices and recommendations to ensure that your FPolicy configuration provides robust monitoring performance and results that meet your requirements.

For specific guidelines related to performance, sizing, and configuration, work with your FPolicy partner application.

Policy configuration

Configuration of the FPolicy external engine, events, and scope for SVMs can improve your overall experience and security.

- Configuration of the FPolicy external engine for SVMs:
  - Providing additional security comes with a performance cost. Enabling Secure Sockets Layer (SSL) communication has a performance effect on accessing shares.
  - The FPolicy external engine should be configured with more than one FPolicy server to provide resiliency and high availability of FPolicy server notification processing.
- Configuration of FPolicy events for SVMs:

  Monitoring file operations influences your overall experience. For example, filtering unwanted file operations on the storage side improves your experience. NetApp recommends setting up the following configuration:

  - Monitoring the minimum types of file operations and enabling the maximum number of filters without breaking the use case.
  - Using filters for getattr, read, write, open, and close operations. The SMB and NFS home directory environments have a high percentage of these operations.
- Configuration of FPolicy scope for SVMs:

  Restrict the scope of the policies to the relevant storage objects, such as shares, volumes, and exports, instead of enabling them across the entire SVM. NetApp recommends checking the directory extensions. If the `is-file-extension-check-on-directories-enabled` parameter is set to `true`, directory objects are subjected to the same extension checks as regular files.

Network configuration

Network connectivity between the FPolicy server and the controller should be of low latency. NetApp recommends separating FPolicy traffic from client traffic by using a private network.

In addition, you should place external FPolicy servers (FPolicy servers) in close proximity to the cluster with high-bandwidth connectivity to provide minimal latency and high-bandwidth connectivity.

> (i) For a scenario in which the LIF for FPolicy traffic is configured on a different port to the LIF for client traffic, the FPolicy LIF might fail over to the other node because of a port failure. As a result, the FPolicy server becomes unreachable from the node which causes the FPolicy notifications for file operations on the node to fail. To avoid this issue, verify that the FPolicy server can be reached through at least one LIF on the node to process FPolicy requests for the file operations performed on that node.

Hardware configuration

You can have the FPolicy server on either a physical server or a virtual server. If the FPolicy server is in a virtual environment, you should allocate dedicated resources (CPU, network, and memory) to the virtual server.

The cluster node-to-FPolicy server ratio should be optimized to ensure that FPolicy servers are not overloaded, which can introduce latencies when the SVM responds to client requests. The optimal ratio depends on the partner application for which the FPolicy server is being used. NetApp recommends working with partners to determine the appropriate value.

Multiple-policy configuration

The FPolicy policy for native blocking has the highest priority, irrespective of the sequence number, and decision-altering policies have a higher priority than others. Policy priority depends on the use case. NetApp recommends working with partners to determine the appropriate priority.

Size considerations

FPolicy performs in-line monitoring of SMB and NFS operations, sends notifications to the external server, and waits for a response, depending on the mode of external engine communication (synchronous or asynchronous). This process affects the performance of SMB and NFS access and CPU resources.

To mitigate any issues, NetApp recommends working with partners to assess and size the environment before enabling FPolicy. Performance is affected by several factors including the number of users, workload characteristics, such as operations per user and data size, network latency, and failure or server slowness.

Monitor performance

FPolicy is a notification-based system. Notifications are sent to an external server for processing and to generate a response back to ONTAP. This round-trip process increases latency for client access.

Monitoring the performance counters on the FPolicy server and in ONTAP gives you the capability to identify bottlenecks in the solution and to tune the parameters as necessary for an optimal solution. For example, an increase in FPolicy latency has a cascading effect on SMB and NFS access latency. Therefore, you should monitor both workload (SMB and NFS) and FPolicy latency. In addition, you can use quality-of-service policies in ONTAP to set up a workload for each volume or SVM that is enabled for FPolicy.

NetApp recommends running the `statistics show -object workload` command to display workload statistics. In addition, you should monitor the following parameters:

- Average, read, and write latencies
- Total number of operations
- Read and write counters

You can monitor the performance of FPolicy subsystems by using the following FPolicy counters.

> (i) You must be in diagnostic mode to collect statistics related to FPolicy.

Steps

1. Collect FPolicy counters:

   a. `statistics start -object fpolicy -instance` *instance_name* `-sample-id` *ID*

   b. `statistics start -object fpolicy_policy -instance` *instance_name* `-sample-id`

```
        ID
```

2. Display FPolicy counters:

    a. `statistics show -object fpolicy –instance` *`instance_name`* `-sample-id` *`ID`*

    b. `statistics show -object fpolicy_server –instance` *`instance_name`* `-sample-id` *`ID`*

The `fpolicy` and `fpolicy_server` counters give you information on several performance parameters which are described in the following table.

Counters	Description
"fpolicy" counters	
aborted_requests	Number of screen requests for which processing is aborted on the SVM
event_count	List of events resulting in notification
max_request_latency	Maximum screen requests latency
outstanding_requests	Total number of screen requests in process
processed_requests	Total number of screen requests that went through fpolicy processing on the SVM
request_latency_hist	Histogram of latency for screen requests
requests_dispatched_rate	Number of screen requests dispatched per second
requests_received_rate	Number of screen requests received per second
"fpolicy_server" counters	
max_request_latency	Maximum latency for a screen request
outstanding_requests	Total number of screen requests waiting for response
request_latency	Average latency for screen request
request_latency_hist	Histogram of latency for screen requests
request_sent_rate	Number of screen requests sent to FPolicy server per second
response_received_rate	Number of screen responses received from FPolicy server per second

Manage FPolicy workflow and dependency on other technologies

NetApp recommends disabling an FPolicy policy before making any configuration changes. For example, if you want to add or modify an IP address in the external engine configured for the enabled policy, first disable the policy.

If you configure FPolicy to monitor NetApp FlexCache volumes, NetApp recommends that you not configure FPolicy to monitor read and getattr file operations. Monitoring these operations in ONTAP requires the retrieval of inode-to-path (I2P) data. Because I2P data cannot be retrieved from FlexCache volumes, it must be retrieved from the origin volume. Therefore, monitoring these operations eliminates the performance benefits that FlexCache can provide.

When both FPolicy and an off-box antivirus solution are deployed, the antivirus solution receives notifications

first. FPolicy processing starts only after antivirus scanning is complete. It is important that you size antivirus solutions correctly because a slow antivirus scanner can affect overall performance.

Passthrough-read upgrade and revert considerations

There are certain upgrade and revert considerations that you must know about before upgrading to an ONTAP release that supports passthrough-read or before reverting to a release that does not support passthrough-read.

Upgrading

After all nodes are upgraded to a version of ONTAP that supports FPolicy passthrough-read, the cluster is capable of using the passthrough-read functionality; however, passthrough-read is disabled by default on existing FPolicy configurations. To use passthrough-read on existing FPolicy configurations, you must disable the FPolicy policy and modify the configuration, and then reenable the configuration.

Reverting

Before reverting to a version of ONTAP that does not support FPolicy passthrough-read, you must meet the following conditions:

- Disable all the policies using passthrough-read, and then modify the affected configurations so that they do not use passthrough-read.
- Disable FPolicy functionality on the cluster by disabling every FPolicy policy on the cluster.

Before reverting to a version of ONTAP that does not support persistent stores, ensure that none of the Fpolicy policies have a configured persistent store. If a persistent store is configured, the revert will fail.

What the steps for setting up an FPolicy configuration are

Before FPolicy can monitor file access, an FPolicy configuration must be created and enabled on the storage virtual machine (SVM) for which FPolicy services are required.

The steps for setting up and enabling an FPolicy configuration on the SVM are as follows:

1. Create an FPolicy external engine.

   The FPolicy external engine identifies the external FPolicy servers (FPolicy servers) that are associated with a specific FPolicy configuration. If the internal "native" FPolicy engine is used to create a native file-blocking configuration, you do not need to create an FPolicy external engine.

2. Create an FPolicy event.

   An FPolicy event describes what the FPolicy policy should monitor. Events consist of the protocols and file operations to monitor, and can contain a list of filters. Events use filters to narrow the list of monitored events for which the FPolicy external engine must send notifications. Events also specify whether the policy monitors volume operations.

3. Create an FPolicy policy.

   The FPolicy policy is responsible for associating, with the appropriate scope, the set of events that need to be monitored and for which of the monitored events notifications must be sent to the designated FPolicy server (or to the native engine if no FPolicy servers are configured). The policy also defines whether the FPolicy server is allowed privileged access to the data for which it receives notifications. An FPolicy server

needs privileged access if the server needs to access the data. Typical use cases where privileged access is needed include file blocking, quota management, and hierarchical storage management. The policy is where you specify whether the configuration for this policy uses an FPolicy server or the internal "native" FPolicy server.

A policy specifies whether screening is mandatory. If screening is mandatory and all FPolicy servers are down or no response is received from the FPolicy servers within a defined timeout period, then file access is denied.

A policy's boundaries are the SVM. A policy cannot apply to more than one SVM. However, a specific SVM can have multiple FPolicy policies, each with the same or different combination of scope, event, and external server configurations.

4. Configure the policy scope.

   The FPolicy scope determines which volumes, shares, or export-policies the policy acts on or excludes from monitoring. A scope also determines which file extensions should be included or excluded from FPolicy monitoring.

   > (i) Exclude lists take precedence over include lists.

5. Enable the FPolicy policy.

   When the policy is enabled, the control channels and, optionally, the privileged data channels are connected. The FPolicy process on the nodes on which the SVM participates begin monitoring file and folder access and, for events that match configured criteria, sends notifications to the FPolicy servers (or to the native engine if no FPolicy servers are configured).

   > (i) If the policy uses native file blocking, an external engine is not configured or associated with the policy.

Plan the FPolicy external engine configuration

Plan the FPolicy external engine configuration

Before you configure the FPolicy external engine (external engine), you must understand what it means to create an external engine and which configuration parameters are available. This information helps you to determine which values to set for each parameter.

Information that is defined when creating the FPolicy external engine

The external engine configuration defines the information that FPolicy needs to make and manage connections to the external FPolicy servers (FPolicy servers), including the following information:

- SVM name
- Engine name
- The IP addresses of the primary and secondary FPolicy servers and the TCP port number to use when making the connection to the FPolicy servers
- Whether the engine type is asynchronous or synchronous

- How to authenticate the connection between the node and the FPolicy server

  If you choose to configure mutual SSL authentication, then you must also configure parameters that provide SSL certificate information.

- How to manage the connection using various advanced privilege settings

  This includes parameters that define such things as timeout values, retry values, keep-alive values, maximum request values, sent and receive buffer size values, and session timeout values.

The `vserver fpolicy policy external-engine create` command is used to create an FPolicy external engine.

What the basic external engine parameters are

You can use the following table of basic FPolicy configuration parameters to help you plan your configuration:

Type of information	Option
SVM Specifies the SVM name that you want to associate with this external engine. Each FPolicy configuration is defined within a single SVM. The external engine, policy event, policy scope, and policy that combine together to create an FPolicy policy configuration must all be associated with the same SVM.	`-vserver vserver_name`
Engine name Specifies the name to assign to the external engine configuration. You must specify the external engine name later when you create the FPolicy policy. This associates the external engine with the policy. The name can be up to 256 characters long. (i) The name should be up to 200 characters long if configuring the external engine name in a MetroCluster or SVM disaster recovery configuration. The name can contain any combination of the following ASCII-range characters: • `a` through `z` • `A` through `Z` • `0` through `9` • "_", "-", and "."	`-engine-name engine_name`

Primary FPolicy servers Specifies the primary FPolicy servers to which the node sends notifications for a given FPolicy policy. The value is specified as a comma-delimited list of IP addresses. If more than one primary server IP address is specified, every node on which the SVM participates creates a control connection to every specified primary FPolicy server at the time the policy is enabled. If you configure multiple primary FPolicy servers, notifications are sent to the FPolicy servers in a round-robin fashion. If the external engine is used in a MetroCluster or SVM disaster recovery configuration, you should specify the IP addresses of the FPolicy servers at the source site as primary servers. The IP addresses of the FPolicy servers at the destination site should be specified as secondary servers.	`-primary-servers IP_address,...`
Port number Specifies the port number of the FPolicy service.	`-port integer`
Secondary FPolicy servers Specifies the secondary FPolicy servers to which to send file access events for a given FPolicy policy. The value is specified as a comma-delimited list of IP addresses. Secondary servers are used only when none of the primary servers are reachable. Connections to secondary servers are established when the policy is enabled, but notifications are sent to secondary servers only if none of the primary servers are reachable. If you configure multiple secondary servers, notifications are sent to the FPolicy servers in a round-robin fashion.	`-secondary-servers IP_address,...`
External engine type Specifies whether the external engine operates in synchronous or asynchronous mode. By default, FPolicy operates in synchronous mode. When set to `synchronous`, file request processing sends a notification to the FPolicy server, but then does not continue until after receiving a response from the FPolicy server. At that point, request flow either continues or processing results in denial, depending on whether the response from the FPolicy server permits the requested action. When set to `asynchronous`, file request processing sends a notification to the FPolicy server, and then continues.	`-extern-engine-type external_engine_type` The value for this parameter can be one of the following: • `synchronous` • `asynchronous`

SSL option for communication with FPolicy server
 	
Specifies the SSL option for communication with the FPolicy server. This is a required parameter. You can choose one of the options based on the following information:
 	
• When set to `no-auth`, no authentication takes place.
 	
   The communication link is established over TCP.
 	
• When set to `server-auth`, the SVM authenticates the FPolicy server using SSL server authentication.
 	
• When set to `mutual-auth`, mutual authentication takes place between the SVM and the FPolicy server; the SVM authenticates the FPolicy server, and the FPolicy server authenticates the SVM.
 	
   If you choose to configure mutual SSL authentication, then you must also configure the `-certificate-common-name`, `-certificate-serial`, and `-certifcate-ca` parameters.
	`-ssl-option {no-auth`
	`|server-auth|mutual-auth}`

Certificate FQDN or custom common name Specifies the certificate name used if SSL authentication between the SVM and the FPolicy server is configured. You can specify the certificate name as an FQDN or as a custom common name. If you specify `mutual-auth` for the `-ssl-option` parameter, you must specify a value for the `-certificate-common-name` parameter.	`-certificate-common` `-name text`
Certificate serial number Specifies the serial number of the certificate used for authentication if SSL authentication between the SVM and the FPolicy server is configured. If you specify `mutual-auth` for the `-ssl-option` parameter, you must specify a value for the `-certificate-serial` parameter.	`-certificate-serial text`
Certificate authority Specifies the CA name of the certificate used for authentication if SSL authentication between the SVM and the FPolicy server is configured. If you specify `mutual-auth` for the `-ssl-option` parameter, you must specify a value for the `-certificate-ca` parameter.	`-certificate-ca text`

What the advanced external engine options are

You can use the following table of advanced FPolicy configuration parameters as you plan whether to customize your configuration with advanced parameters. You use these parameters to modify communication behavior between the cluster nodes and the FPolicy servers:

Type of information	Option
Timeout for canceling a request Specifies the time interval in hours (`h`), minutes (`m`), or seconds (`s`) that the node waits for a response from the FPolicy server. If the timeout interval passes, the node sends a cancel request to the FPolicy server. The node then sends the notification to an alternate FPolicy server. This timeout helps in handling an FPolicy server that is not responding, which can improve SMB/NFS client response. Also, canceling requests after a timeout period can help in releasing system resources because the notification request is moved from a down/bad FPolicy server to an alternate FPolicy server. The range for this value is `0` through `100`. If the value is set to `0`, the option is disabled and cancel request messages are not sent to the FPolicy server. The default is `20s`.	`-reqs-cancel-timeout integer[h|m|s]`
Timeout for aborting a request Specifies the timeout in hours (`h`), minutes (`m`), or seconds (`s`) for aborting a request. The range for this value is `0` through `200`.	`-reqs-abort-timeout ` `` `integer[h|m|s]`
Interval for sending status requests Specifies the interval in hours (`h`), minutes (`m`), or seconds (`s`) after which a status request is sent to the FPolicy server. The range for this value is `0` through `50`. If the value is set to `0`, the option is disabled and status request messages are not sent to the FPolicy server. The default is `10s`.	`-status-req-interval integer[h|m|s]`
Maximum outstanding requests on the FPolicy server Specifies the maximum number of outstanding requests that can be queued on the FPolicy server. The range for this value is `1` through `10000`. The default is `500`.	`-max-server-reqs integer`

Timeout for disconnecting a nonresponsive FPolicy server Specifies the time interval in hours (`h`), minutes (`m`), or seconds (`s`) after which the connection to the FPolicy server is terminated. The connection is terminated after the timeout period only if the FPolicy server's queue contains the maximum allowed requests and no response is received within the timeout period. The maximum allowed number of requests is either `50` (the default) or the number specified by the `max-server-reqs-` parameter. The range for this value is `1` through `100`. The default is `60s`.	`-server-progress` `-timeout integer[h|m|s]`
Interval for sending keep-alive messages to the FPolicy server Specifies the time interval in hours (`h`), minutes (`m`), or seconds (`s`) at which keep-alive messages are sent to the FPolicy server. Keep-alive messages detect half-open connections. The range for this value is `10` through `600`. If the value is set to `0`, the option is disabled and keep-alive messages are prevented from being sent to the FPolicy servers. The default is `120s`.	`-keep-alive-interval-` `integer[h|m|s]`
Maximum reconnect attempts Specifies the maximum number of times the SVM attempts to reconnect to the FPolicy server after the connection has been broken. The range for this value is `0` through `20`. The default is `5`.	`-max-connection-retries` `integer`
Receive buffer size Specifies the receive buffer size of the connected socket for the FPolicy server. The default value is set to 256 kilobytes (Kb). When the value is set to 0, the size of the receive buffer is set to a value defined by the system. For example, if the default receive buffer size of the socket is 65536 bytes, by setting the tunable value to 0, the socket buffer size is set to 65536 bytes. You can use any non-default value to set the size (in bytes) of the receive buffer.	`-recv-buffer-size` `integer`

Send buffer size Specifies the send buffer size of the connected socket for the FPolicy server. The default value is set to 256 kilobytes (Kb). When the value is set to 0, the size of the send buffer is set to a value defined by the system. For example, if the default send buffer size of the socket is set to 65536 bytes, by setting the tunable value to 0, the socket buffer size is set to 65536 bytes. You can use any non-default value to set the size (in bytes) of the send buffer.	`-send-buffer-size integer`
Timeout for purging a session ID during reconnection Specifies the interval in hours (`h`), minutes (`m`), or seconds (`s`) after which a new session ID is sent to the FPolicy server during reconnection attempts. If the connection between the storage controller and the FPolicy server is terminated and reconnection is made within the `-session-timeout` interval, the old session ID is sent to FPolicy server so that it can send responses for old notifications. The default value is set to 10 seconds.	`-session-timeout [integerh][integerm][integers]`

Additional information about configuring FPolicy external engines to use SSL authenticated connections

You need to know some additional information if you want to configure the FPolicy external engine to use SSL when connecting to FPolicy servers.

SSL server authentication

If you choose to configure the FPolicy external engine for SSL server authentication, before creating the external engine, you must install the public certificate of the certificate authority (CA) that signed the FPolicy server certificate.

Mutual authentication

If you configure FPolicy external engines to use SSL mutual authentication when connecting storage virtual machine (SVM) data LIFs to external FPolicy servers, before creating the external engine, you must install the public certificate of the CA that signed the FPolicy server certificate along with the public certificate and key file for authentication of the SVM. You must not delete this certificate while any FPolicy policies are using the installed certificate.

If the certificate is deleted while FPolicy is using it for mutual authentication when connecting to an external FPolicy server, you cannot reenable a disabled FPolicy policy that uses that certificate. The FPolicy policy cannot be reenabled in this situation even if a new certificate with the same settings is created and installed on the SVM.

If the certificate has been deleted, you need to install a new certificate, create new FPolicy external engines that use the new certificate, and associate the new external engines with the FPolicy policy that you want to reenable by modifying the FPolicy policy.

Install certificates for SSL

The public certificate of the CA that is used to sign the FPolicy server certificate is installed by using the `security certificate install` command with the `-type` parameter set to `client-ca`. The private key and public certificate required for authentication of the SVM is installed by using the `security certificate install` command with the `-type` parameter set to `server`.

Certificates do not replicate in SVM disaster recovery relationships with a non-ID-preserve configuration

Security certificates used for SSL authentication when making connections to FPolicy servers do not replicate to SVM disaster recovery destinations with non-ID-preserve configurations. Although the FPolicy external-engine configuration on the SVM is replicated, security certificates are not replicated. You must manually install the security certificates on the destination.

When you set up the SVM disaster recovery relationship, the value you select for the `-identity-preserve` option of the `snapmirror create` command determines the configuration details that are replicated in the destination SVM.

If you set the `-identity-preserve` option to `true` (ID-preserve), all of the FPolicy configuration details are replicated, including the security certificate information. You must install the security certificates on the destination only if you set the option to `false` (non-ID-preserve).

Restrictions for cluster-scoped FPolicy external engines with MetroCluster and SVM disaster recovery configurations

You can create a cluster-scoped FPolicy external engine by assigning the cluster storage virtual machine (SVM) to the external engine. However, when creating a cluster-scoped external engine in a MetroCluster or SVM disaster recovery configuration, there are certain restrictions when choosing the authentication method that the SVM uses for external communication with the FPolicy server.

There are three authentication options that you can choose when creating external FPolicy servers: no authentication, SSL server authentication, and SSL mutual authentication. Although there are no restrictions when choosing the authentication option if the external FPolicy server is assigned to a data SVM, there are restrictions when creating a cluster-scoped FPolicy external engine:

Configuration	Permitted?
MetroCluster or SVM disaster recovery and a cluster-scoped FPolicy external engine with no authentication (SSL is not configured)	Yes
MetroCluster or SVM disaster recovery and a cluster-scoped FPolicy external engine with SSL server or SSL mutual authentication	No

- If a cluster-scoped FPolicy external engine with SSL authentication exists and you want to create a MetroCluster or SVM disaster recovery configuration, you must modify this external engine to use no authentication or remove the external engine before you can create the MetroCluster or SVM disaster recovery configuration.

- If the MetroCluster or SVM disaster recovery configuration already exists, ONTAP prevents you from creating a cluster-scoped FPolicy external engine with SSL authentication.

Complete the FPolicy external engine configuration worksheet

You can use this worksheet to record the values that you need during the FPolicy external engine configuration process. If a parameter value is required, you need to determine what value to use for those parameters before you configure the external engine.

Information for a basic external engine configuration

You should record whether you want to include each parameter setting in the external engine configuration and then record the value for the parameters that you want to include.

Type of information	Required	Include	Your values
Storage virtual machine (SVM) name	Yes	Yes	
Engine name	Yes	Yes	
Primary FPolicy servers	Yes	Yes	
Port number	Yes	Yes	
Secondary FPolicy servers	No		
External engine type	No		
SSL option for communication with external FPolicy server	Yes	Yes	
Certificate FQDN or custom common name	No		
Certificate serial number	No		
Certificate authority	No		

Information for advanced external engine parameters

To configure an external engine with advanced parameters, you must enter the configuration command while in advanced privilege mode.

Type of information	Required	Include	Your values
Timeout for canceling a request	No		
Timeout for aborting a request	No		

Interval for sending status requests	No		
Maximum outstanding requests on the FPolicy server	No		
Timeout for disconnecting a nonresponsive FPolicy server	No		
Interval for sending keep-alive messages to the FPolicy server	No		
Maximum reconnect attempts	No		
Receive buffer size	No		
Send buffer size	No		
Timeout for purging a session ID during reconnection	No		

Plan the FPolicy event configuration

Plan the FPolicy event configuration overview

Before you configure FPolicy events, you must understand what it means to create an FPolicy event. You must determine which protocols you want the event to monitor, which events to monitor, and which event filters to use. This information helps you plan the values that you want to set.

What it means to create an FPolicy event

Creating the FPolicy event means defining information that the FPolicy process needs to determine what file access operations to monitor and for which of the monitored events notifications should be sent to the external FPolicy server. The FPolicy event configuration defines the following configuration information:

- Storage virtual machine (SVM) name
- Event name
- Which protocols to monitor

  FPolicy can monitor SMB, NFSv3, and NFSv4 file access operations.

- Which file operations to monitor

  Not all file operations are valid for each protocol.

- Which file filters to configure

  Only certain combinations of file operations and filters are valid. Each protocol has its own set of supported combinations.

- Whether to monitor volume mount and unmount operations

> ℹ️ There is a dependency with three of the parameters (`-protocol`, `-file-operations`, `-filters`). The following combinations are valid for the three parameters:
>
> - You can specify the `-protocol` and `-file-operations` parameters.
> - You can specify all three of the parameters.
> - You can specify none of the parameters.

What the FPolicy event configuration contains

You can use the following list of available FPolicy event configuration parameters to help you plan your configuration:

Type of information	Option
SVM Specifies the SVM name that you want to associate with this FPolicy event. Each FPolicy configuration is defined within a single SVM. The external engine, policy event, policy scope, and policy that combine together to create an FPolicy policy configuration must all be associated with the same SVM.	`-vserver vserver_name`
Event name Specifies the name to assign to the FPolicy event. When you create the FPolicy policy you associate the FPolicy event with the policy using the event name. The name can be up to 256 characters long. > ℹ️ The name should be up to 200 characters long if configuring the event in a MetroCluster or SVM disaster recovery configuration. The name can contain any combination of the following ASCII-range characters: - `a` through `z` - `A` through `Z` - `0` through `9` - " _ ", "-", and "."	`-event-name event_name`

Protocol Specifies which protocol to configure for the FPolicy event. The list for `-protocol` can include one of the following values: • `cifs` • `nfsv3` • `nfsv4` (i) If you specify `-protocol`, then you must specify a valid value in the `-file-operations` parameter. As the protocol version changes, the valid values might change.	`-protocol protocol`
File operations Specifies the list of file operations for the FPolicy event. The event checks the operations specified in this list from all client requests using the protocol specified in the `-protocol` parameter. You can list one or more file operations by using a comma-delimited list. The list for `-file -operations` can include one or more of the following values: • `close` for file close operations • `create` for file create operations • `create-dir` for directory create operations • `delete` for file delete operations • `delete_dir` for directory delete operations • `getattr` for get attribute operations • `link` for link operations • `lookup` for lookup operations • `open` for file open operations • `read` for file read operations • `write` for file write operations • `rename` for file rename operations • `rename_dir` for directory rename operations • `setattr` for set attribute operations • `symlink` for symbolic link operations (i) If you specify `-file-operations`, then you must specify a valid protocol in the `-protocol` parameter.	`-file-operations` `file_operations,…`

Filters Specifies the list of filters for a given file operation for the specified protocol. The values in the `-filters` parameter are used to filter client requests. The list can include one or more of the following: (i) If you specify the `-filters` parameter, then you must also specify valid values for the `-file-operations` and `-protocol` parameters. • `monitor-ads` option to filter the client request for alternate data stream. • `close-with-modification` option to filter the client request for close with modification. • `close-without-modification` option to filter the client request for close without modification. • `first-read` option to filter the client request for first read. • `first-write` option to filter the client request for first write. • `offline-bit` option to filter the client request for offline bit set. Setting this filter results in the FPolicy server receiving notification only when offline files are accessed. • `open-with-delete-intent` option to filter the client request for open with delete intent. Setting this filter results in the FPolicy server receiving notification only when an attempt is made to open a file with the intent to delete it. This is used by file systems when the `FILE_DELETE_ON_CLOSE` flag is specified. • `open-with-write-intent` option to filter client request for open with write intent. Setting this filter results in the FPolicy server receiving notification only when an attempt is made to open a file with the intent to write something in it. • `write-with-size-change` option to filter the client request for write with size change.	`-filters filter, ...`

Filters continued
 	
  • `setattr-with-owner-change` option to filter the client setattr requests for changing owner of a file or a directory.
 	
  • `setattr-with-group-change` option to filter the client setattr requests for changing the group of a file or a directory.
 	
  • `setattr-with-sacl-change` option to filter the client setattr requests for changing the SACL on a file or a directory.
 	
    This filter is available only for the SMB and NFSv4 protocols.
 	
  • `setattr-with-dacl-change` option to filter the client setattr requests for changing the DACL on a file or a directory.
 	
    This filter is available only for the SMB and NFSv4 protocols.
 	
  • `setattr-with-modify-time-change` option to filter the client setattr requests for changing the modification time of a file or a directory.
 	
  • `setattr-with-access-time-change` option to filter the client setattr requests for changing the access time of a file or a directory.
 	
  • `setattr-with-creation-time-change` option to filter the client setattr requests for changing the creation time of a file or a directory.
 	
    This option is available only for the SMB protocol.
 	
  • `setattr-with-mode-change` option to filter the client setattr requests for changing the mode bits on a file or a directory.
 	
  • `setattr-with-size-change` option to filter the client setattr requests for changing the size of a file.
 	
  • `setattr-with-allocation-size-change` option to filter the client setattr requests for changing the allocation size of a file.
 	
    This option is available only for the SMB protocol.
 	
  • `exclude-directory` option to filter the client requests for directory operations.
 	
    When this filter is specified, the directory operations are not monitored.
	`-filters filter,...`

Is volume operation required
 	
Specifies whether monitoring is required for volume mount and unmount operations. The default is `false`.
	`-volume-operation {true |false}`
	
	`-filters filter,...`

| *FPolicy access denied notifications*

Beginning with ONTAP 9.13.1, users can receive notifications for failed file operations due to lack of permissions. These notifications are valuable for security, ransomware protection, and governance. Notifications will be generated for file operation failed due to lack of permission, which includes:

• Failures due to NTFS permissions.

• Failures due to Unix mode bits.

• Failures due to NFSv4 ACLs. | `-monitor-fileop-failure {true|false}` |

Supported file operation and filter combinations that FPolicy can monitor for SMB

When you configure your FPolicy event, you need to be aware that only certain combinations of file operations and filters are supported for monitoring SMB file access operations.

The list of supported file operation and filter combinations for FPolicy monitoring of SMB file access events is provided in the following table:

Supported file operations	Supported filters
close	monitor-ads, offline-bit, close-with-modification, close-without-modification, close-with-read, exclude-directory
create	monitor-ads, offline-bit
create_dir	Currently no filter is supported for this file operation.
delete	monitor-ads, offline-bit
delete_dir	Currently no filter is supported for this file operation.
getattr	offline-bit, exclude-dir
open	monitor-ads, offline-bit, open-with-delete-intent, open-with-write-intent, exclude-dir
read	monitor-ads, offline-bit, first-read
write	monitor-ads, offline-bit, first-write, write-with-size-change
rename	monitor-ads, offline-bit
rename_dir	Currently no filter is supported for this file operation.

setattr	monitor-ads, offline-bit, setattr_with_owner_change, setattr_with_group_change, setattr_with_mode_change, setattr_with_sacl_change, setattr_with_dacl_change, setattr_with_modify_time_change, setattr_with_access_time_change, setattr_with_creation_time_change, setattr_with_size_change, setattr_with_allocation_size_change, exclude_directory

Beginning with ONTAP 9.13.1, users can receive notifications for failed file operations due to lack of permissions. The list of supported access denied file operation and filter combinations for FPolicy monitoring of SMB file access events is provided in the following table:

Supported access denied file operation	Supported filters
open	NA

Supported file operation and filter combinations that FPolicy can monitor for NFSv3

When you configure your FPolicy event, you need to be aware that only certain combinations of file operations and filters are supported for monitoring NFSv3 file access operations.

The list of supported file operation and filter combinations for FPolicy monitoring of NFSv3 file access events is provided in the following table:

Supported file operations	Supported filters
create	offline-bit
create_dir	Currently no filter is supported for this file operation.
delete	offline-bit
delete_dir	Currently no filter is supported for this file operation.
link	offline-bit
lookup	offline-bit, exclude-dir
read	offline-bit, first-read
write	offline-bit, first-write, write-with-size-change
rename	offline-bit
rename_dir	Currently no filter is supported for this file operation.

setattr	offline-bit, setattr_with_owner_change, setattr_with_group_change, setattr_with_mode_change, setattr_with_modify_time_change, setattr_with_access_time_change, setattr_with_size_change, exclude_directory
symlink	offline-bit

Beginning with ONTAP 9.13.1, users can receive notifications for failed file operations due to lack of permissions. The list of supported access denied file operation and filter combinations for FPolicy monitoring of NFSv3 file access events is provided in the following table:

Supported access denied file operation	Supported filters
access	NA
create	NA
create_dir	NA
delete	NA
delete_dir	NA
link	NA
read	NA
rename	NA
rename_dir	NA
setattr	NA
write	NA

Supported file operation and filter combinations that FPolicy can monitor for NFSv4

When you configure your FPolicy event, you need to be aware that only certain combinations of file operations and filters are supported for monitoring NFSv4 file access operations.

The list of supported file operation and filter combinations for FPolicy monitoring of NFSv4 file access events is provided in the following table:

Supported file operations	Supported filters

close	offline-bit, exclude-directory
create	offline-bit
create_dir	Currently no filter is supported for this file operation.
delete	offline-bit
delete_dir	Currently no filter is supported for this file operation.
getattr	offline-bit, exclude-directory
link	offline-bit
lookup	offline-bit, exclude-directory
open	offline-bit, exclude-directory
read	offline-bit, first-read
write	offline-bit, first-write, write-with-size-change
rename	offline-bit
rename_dir	Currently no filter is supported for this file operation.
setattr	offline-bit, setattr_with_owner_change, setattr_with_group_change, setattr_with_mode_change, setattr_with_sacl_change, setattr_with_dacl_change, setattr_with_modify_time_change, setattr_with_access_time_change, setattr_with_size_change, exclude_directory
symlink	offline-bit

Beginning with ONTAP 9.13.1, users can receive notifications for failed file operations due to lack of permissions. The list of supported access denied file operation and filter combinations for FPolicy monitoring of NFSv4 file access events is provided in the following table:

Supported access denied file operation	Supported filters
access	NA
create	NA
create_dir	NA

delete	NA
delete_dir	NA
link	NA
open	NA
read	NA
rename	NA
rename_dir	NA
setattr	NA
write	NA

Complete the FPolicy event configuration worksheet

You can use this worksheet to record the values that you need during the FPolicy event configuration process. If a parameter value is required, you need to determine what value to use for those parameters before you configure the FPolicy event.

You should record whether you want to include each parameter setting in the FPolicy event configuration and then record the value for the parameters that you want to include.

Type of information	Required	Include	Your values
Storage virtual machine (SVM) name	Yes	Yes	
Event name	Yes	Yes	
Protocol	No		
File operations	No		
Filters	No		
Volume operation	No		
Access denied events (support beginning with ONTAP 9.13)	No		

Plan the FPolicy policy configuration

Plan the FPolicy policy configuration overview

Before you configure the FPolicy policy, you must understand which parameters are required when creating the policy as well as why you might want to configure certain optional parameters. This information helps you to determine which values to set for each parameter.

When creating an FPolicy policy you associate the policy with the following:

- The storage virtual machine (SVM)
- One or more FPolicy events
- An FPolicy external engine

You can also configure several optional policy settings.

What the FPolicy policy configuration contains

You can use the following list of available FPolicy policy required and optional parameters to help you plan your configuration:

Type of information	Option	Required	Default
SVM name Specifies the name of the SVM on which you want to create an FPolicy policy.	`-vserver vserver_name`	Yes	None

Policy name Specifies the name of the FPolicy policy. The name can be up to 256 characters long. (i) The name should be up to 200 characters long if configuring the policy in a MetroCluster or SVM disaster recovery configuration. The name can contain any combination of the following ASCII-range characters: • `a` through `z` • `A` through `Z` • `0` through `9` • "_", "-", and "."	`-policy-name policy_name`	Yes	None
Event names Specifies a comma-delimited list of events to associate with the FPolicy policy. • You can associate more than one event to a policy. • An event is specific to a protocol. • You can use a single policy to monitor file access events for more than one protocol by creating an event for each protocol that you want the policy to monitor, and then associating the events to the policy. • The events must already exist.	`-events event_name, ...`	Yes	None

External engine name Specifies the name of the external engine to associate with the FPolicy policy. • An external engine contains information required by the node to send notifications to an FPolicy server. • You can configure FPolicy to use the ONTAP native external engine for simple file blocking or to use an external engine that is configured to use external FPolicy servers (FPolicy servers) for more sophisticated file blocking and file management. • If you want to use the native external engine, you can either not specify a value for this parameter or you can specify `native` as the value. • If you want to use FPolicy servers, the configuration for the external engine must already exist.	`-engine engine_name`	Yes (unless the policy uses the internal ONTAP native engine)	`native`
Is mandatory screening required Specifies whether mandatory file access screening is required. • The mandatory screening setting determines what action is taken on a file access event in a case when all primary and secondary servers are down or no response is received from the FPolicy servers within a given timeout period. • When set to `true`, file access events are denied. • When set to `false`, file access events are allowed.	`-is-mandatory {true|false}`	No	`true`

Allow privileged access
 	
Specifies whether you want the FPolicy server to have privileged access to the monitored files and folders by using a privileged data connection.
 	
If configured, FPolicy servers can access files from the root of the SVM containing the monitored data using the privileged data connection.
 	
For privileged data access, SMB must be licensed on the cluster and all the data LIFs used to connect to the FPolicy servers must be configured to have `cifs` as one of the allowed protocols.
 	
If you want to configure the policy to allow privileged access, you must also specify the user name for the account that you want the FPolicy server to use for privileged access.
	`-allow`
	`-privileged`
	`-access {yes|no}`
		No (unless passthrough-read is enabled)	`no`

Privileged user name
 	
Specifies the user name of the account the FPolicy servers use for privileged data access.
 	
• The value for this parameter should use the "domain\user name" format.
 	
• If `-allow-privileged-access` is set to `no`, any value set for this parameter is ignored.
	`-privileged`
	`-user-name`
	`user_name`
		No (unless privileged access is enabled)	None

| Allow passthrough-read

Specifies whether the FPolicy servers can provide passthrough-read services for files that have been archived to secondary storage (offline files) by the FPolicy servers:

• Passthrough-read is a way to read data for offline files without restoring the data to the primary storage.

  Passthrough-read reduces response latencies because there is no need to recall files back to primary storage before responding to the read request. Additionally, passthrough-read optimizes storage efficiency by eliminating the need to consume primary storage space with files that are recalled solely to satisfy read requests.

• When enabled, the FPolicy servers provide the data for the file over a separate privileged data channel opened specifically for passthrough-reads.

• If you want to configure passthrough-read, the policy must also be configured to allow privileged access.	`-is-passthrough -read-enabled {true|false}`	No	`false`

Requirement for FPolicy scope configurations if the FPolicy policy uses the native engine

If you configure the FPolicy policy to use the native engine, there is a specific requirement for how you define the FPolicy scope configured for the policy.

The FPolicy scope defines the boundaries on which the FPolicy policy applies, for example whether the FPolicy applies to specified volumes or shares. There are a number of parameters that further restrict the scope to which the FPolicy policy applies. One of these parameters, `-is-file-extension-check-on -directories-enabled`, specifies whether to check file extensions on directories. The default value is `false`, which means that file extensions on directories are not checked.

When an FPolicy policy that uses the native engine is enabled on a share or volume and the `-is-file -extension-check-on-directories-enabled` parameter is set to `false` for the scope of the policy, directory access is denied. With this configuration, because the file extensions are not checked for directories, any directory operation is denied if it falls under the scope of the policy.

To ensure that directory access succeeds when using the native engine, you must set the `-is-file -extension-check-on-directories-enabled parameter` to `true` when creating the scope.

With this parameter set to `true`, extension checks happen for directory operations and the decision whether to

allow or deny access is taken based on the extensions included or excluded in the FPolicy scope configuration.

Complete the FPolicy policy worksheet

You can use this worksheet to record the values that you need during the FPolicy policy configuration process. You should record whether you want to include each parameter setting in the FPolicy policy configuration and then record the value for the parameters that you want to include.

Type of information	Include	Your values
Storage virtual machine (SVM) name	Yes	
Policy name	Yes	
Event names	Yes	
External engine name		
Is mandatory screening required?		
Allow privileged access		
Privileged user name		
Is passthrough-read enabled?		

Plan the FPolicy scope configuration

Plan the FPolicy scope configuration overview

Before you configure the FPolicy scope, you must understand what it means to create a scope. You must understand what the scope configuration contains. You also need to understand what the scope rules of precedence are. This information can help you plan the values that you want to set.

What it means to create an FPolicy scope

Creating the FPolicy scope means defining the boundaries on which the FPolicy policy applies. The storage virtual machine (SVM) is the basic boundary. When you create a scope for an FPolicy policy, you must define the FPolicy policy to which it will apply, and you must designate to which SVM you want to apply the scope.

There are a number of parameters that further restrict the scope within the specified SVM. You can restrict the scope by specifying what to include in the scope or by specifying what to exclude from the scope. After you apply a scope to an enabled policy, policy event checks get applied to the scope defined by this command.

Notifications are generated for file access events where matches are found in the "include" options. Notifications are not generated for file access events where matches are found in the "exclude" options.

The FPolicy scope configuration defines the following configuration information:

* SVM name

* Policy name

* The shares to include or exclude from what gets monitored

* The export policies to include or exclude from what gets monitored

* The volumes to include or exclude from what gets monitored

* The file extensions to include or exclude from what gets monitored

* Whether to do file extension checks on directory objects

> ⓘ There are special considerations for the scope for a cluster FPolicy policy. The cluster FPolicy policy is a policy that the cluster administrator creates for the admin SVM. If the cluster administrator also creates the scope for that cluster FPolicy policy, the SVM administrator cannot create a scope for that same policy. However, if the cluster administrator does not create a scope for the cluster FPolicy policy, then any SVM administrator can create the scope for that cluster policy. If the SVM administrator creates a scope for that cluster FPolicy policy, the cluster administrator cannot subsequently create a cluster scope for that same cluster policy. This is because the cluster administrator cannot override the scope for the same cluster policy.

What the scope rules of precedence are

The following rules of precedence apply to scope configurations:

* When a share is included in the `-shares-to-include` parameter and the parent volume of the share is included in the `-volumes-to-exclude` parameter, `-volumes-to-exclude` has precedence over `-shares-to-include`.

* When an export policy is included in the `-export-policies-to-include` parameter and the parent volume of the export policy is included in the `-volumes-to-exclude` parameter, `-volumes-to-exclude` has precedence over `-export-policies-to-include`.

* An administrator can specify both `-file-extensions-to-include` and `-file-extensions-to-exclude` lists.

  The `-file-extensions-to-exclude` parameter is checked before the `-file-extensions-to-include` parameter is checked.

What the FPolicy scope configuration contains

You can use the following list of available FPolicy scope configuration parameters to help you plan your configuration:

> ⓘ When configuring what shares, export policies, volumes, and file extensions to include or exclude from the scope, the include and exclude parameters can include metacharacters such as "?" and "*". The use of regular expressions is not supported.

Type of information	Option

SVM Specifies the SVM name on which you want to create an FPolicy scope. Each FPolicy configuration is defined within a single SVM. The external engine, policy event, policy scope, and policy that combine together to create an FPolicy policy configuration must all be associated with the same SVM.	`-vserver vserver_name`
Policy name Specifies the name of the FPolicy policy to which you want to attach the scope. The FPolicy policy must already exist.	`-policy-name policy_name`
Shares to include Specifies a comma-delimited list of shares to monitor for the FPolicy policy to which the scope is applied.	`-shares-to-include share_name, …`
Shares to exclude Specifies a comma-delimited list of shares to exclude from monitoring for the FPolicy policy to which the scope is applied.	`-shares-to-exclude share_name, …`
Volumes to include Specifies a comma-delimited list of volumes to monitor for the FPolicy policy to which the scope is applied.	`-volumes-to-include volume_name, …`
Volumes to exclude Specifies a comma-delimited list of volumes to exclude from monitoring for the FPolicy policy to which the scope is applied.	`-volumes-to-exclude volume_name, …`
Export policies to include Specifies a comma-delimited list of export policies to monitor for the FPolicy policy to which the scope is applied.	`-export-policies-to -include export_policy_name, …`
Export policies to exclude Specifies a comma-delimited list of export policies to exclude from monitoring for the FPolicy policy to which the scope is applied.	`-export-policies-to -exclude export_policy_name, …`
File extensions to include Specifies a comma-delimited list of file extensions to monitor for the FPolicy policy to which the scope is applied.	`-file-extensions-to -include file_extensions, …`

File extension to exclude
 	
Specifies a comma-delimited list of file extensions to exclude from monitoring for the FPolicy policy to which the scope is applied.
	`-file-extensions-to`
	`-exclude`
	`file_extensions, ...`

Is file extension check on directory enabled ?
 	
Specifies whether the file name extension checks apply to directory objects as well. If this parameter is set to `true`, the directory objects are subjected to the same extension checks as regular files. If this parameter is set to `false`, the directory names are not matched for extensions and notifications are sent for directories even if their name extensions do not match.
 	
If the FPolicy policy to which the scope is assigned is configured to use the native engine, this parameter must be set to `true`.
	`-is-file-extension`
	`-check-on-directories`
	`-enabled {true| false|}`

Complete the FPolicy scope worksheet

You can use this worksheet to record the values that you need during the FPolicy scope configuration process. If a parameter value is required, you need to determine what value to use for those parameters before you configure the FPolicy scope.

You should record whether you want to include each parameter setting in the FPolicy scope configuration and then record the value for the parameters that you want to include.

Type of information	Required	Include	Your values
Storage virtual machine (SVM) name	Yes	Yes	
Policy name	Yes	Yes	
Shares to include	No		
Shares to exclude	No		
Volumes to include	No		
Volumes to exclude	No		
Export policies to include	No		
Export policies to exclude	No		
File extensions to include	No		
File extension to exclude	No		

Is file extension check on directory enabled?	No		

Create the FPolicy configuration

Create the FPolicy external engine

You must create an external engine to start creating an FPolicy configuration. The external engine defines how FPolicy makes and manages connections to external FPolicy servers. If your configuration uses the internal ONTAP engine (the native external engine) for simple file blocking, you do not need to configure a separate FPolicy external engine and do not need to perform this step.

What you'll need

The external engine worksheet should be completed.

About this task

If the external engine is used in a MetroCluster configuration, you should specify the IP addresses of the FPolicy servers at the source site as primary servers. The IP addresses of the FPolicy servers at the destination site should be specified as secondary servers.

Steps

1. Create the FPolicy external engine by using the `vserver fpolicy policy external-engine create` command.

   The following command creates an external engine on storage virtual machine (SVM) vs1.example.com. No authentication is required for external communications with the FPolicy server.

   ```
   vserver fpolicy policy external-engine create -vserver-name vs1.example.com
   -engine-name engine1 -primary-servers 10.1.1.2,10.1.1.3 -port 6789 -ssl-option
   no-auth
   ```

2. Verify the FPolicy external engine configuration by using the `vserver fpolicy policy external-engine show` command.

   The following command display information about all external engines configured on SVM vs1.example.com:

   ```
   vserver fpolicy policy external-engine show -vserver vs1.example.com
   ```

```
                                Primary          Secondary
External
Vserver             Engine       Servers         Servers        Port Engine
Type
---------------     -----------  -------------- ----------- ------
-----------
vs1.example.com     engine1      10.1.1.2,       -              6789
synchronous

                                 10.1.1.3
```

The following command displays detailed information about the external engine named "engine1" on SVM
vs1.example.com:

```
vserver fpolicy policy external-engine show -vserver vs1.example.com -engine
-name engine1
```

```
                               Vserver: vs1.example.com
                                Engine: engine1
                Primary FPolicy Servers: 10.1.1.2, 10.1.1.3
         Port Number of FPolicy Service: 6789
              Secondary FPolicy Servers: -
                    External Engine Type: synchronous
 SSL Option for External Communication: no-auth
             FQDN or Custom Common Name: -
           Serial Number of Certificate: -
                   Certificate Authority: -
```

Create the FPolicy event

As part of creating an FPolicy policy configuration, you need to create an FPolicy event.
You associate the event with the FPolicy policy when it is created. An event defines which
protocol to monitor and which file access events to monitor and filter.

Before you begin

You should complete the FPolicy event worksheet.

Create the FPolicy event

1. Create the FPolicy event by using the `vserver fpolicy policy event create` command.

   ```
   vserver fpolicy policy event create -vserver vs1.example.com -event-name
   event1 -protocol cifs -file-operations open,close,read,write
   ```

2. Verify the FPolicy event configuration by using the `vserver fpolicy policy event show` command.

   ```
   vserver fpolicy policy event show -vserver vs1.example.com
   ```

```
                   Event              File                       Is Volume
Vserver            Name      Protocols Operations     Filters   Operation
---------------    --------- --------- ------------- --------- 
------------
vs1.example.com    event1    cifs      open, close,   -         false
                                       read, write
```

Create the FPolicy access denied events

Beginning with ONTAP 9.13.1, users can receive notifications for failed file operations due to lack of permissions. These notifications are valuable for security, ransomware protection, and governance.

1. Create the FPolicy event by using the `vserver fpolicy policy event create` command.

   ```
   vserver fpolicy policy event create -vserver vs1.example.com -event-name
   event1 -protocol cifs -monitor-fileop-failure true -file-operations open
   ```

Create persistent stores

Beginning with ONTAP 9.14.1, FPolicy allows you to set up a Persistent stores to capture file access events for asynchronous non-mandatory policies in the SVM. Persistent stores can help decouple client I/O processing from FPolicy notification processing to reduce client latency. Synchronous (mandatory or non-mandatory) and asynchronous mandatory configurations are not supported.

Best practices
- Before using the persistent store functionality, please ensure your partner applications support this configuration.
- The persistent store volume is setup on a per SVM basis. For each FPolicy enabled SVM you will need a persistent store volume.
- The persistent store volume name and the junction-path specified at the time of volume creation should match.
- Create the persistent store volume on the node with LIFs that expect maximum traffic to be monitored by Fpolicy.
- Have the snapshot policy set to `none` for that volume instead of `default`. This is to ensure that there is no accidental restore of the snapshot leading to loss of current events and to prevent possible duplicate event processing.
- Make the persistent store volume inaccessible for external user protocol access (CIFS/NFS) to avoid accidental corruption or deletion of the persisted event records. To achieve this, after enabling FPolicy, unmount the volume in ONTAP to remove the junction path, this makes it inaccessible for the user protocol access.

Steps
1. Create an empty volume on the SVM that can be provisioned for the persistent store:

   ```
   volume create -vserver <SVM Name> -volume <volume> -state <online> -junction
   ```

```
-path <path> -policy <default> -unix-permissions <777> -size <value>
-aggregate <aggregate name> -snapshot-policy <none>
```

  - Size of the persistent store volume is based on the time duration for which you want to persist the
    events that are not delivered to the external server (partner application).

    For example, if you want 30 minutes of events to persist in a cluster with a 30K notifications per second
    capacity:

    Required Volume Size = 30000 x 30 x 60 x 0.6KB (avg notification record size) = 32400000 KB = ~32
    GB

    To find the approximate notification rate, you can either reach out to your FPolicy partner application or
    utilize the FPolicy counter `requests_dispatched_rate`.

  - It is expected that an administrator user with sufficient RBAC privileges (to create a volume) will create
    a volume (using the volume cli command or REST API) of the desired size and provide the name of
    that volume as the `-volume` in the persistent store create CLI command or REST API.

2. Create the persistent store:

```
vserver fpolicy persistent store create -vserver <SVM> -persistent-store
<PS_name> -volume <volume>
```

  - persistent-store: The persistent store name

  - volume: The persistent store volume

3. After the persistent store is created, you can create the FPolicy policy and add the persistent store name to
   that policy. For more information, see Create the FPolicy policy.

Create the FPolicy policy

When you create the FPolicy policy, you associate an external engine and one or more
events to the policy. The policy also specifies whether mandatory screening is required,
whether the FPolicy servers have privileged access to data on the storage virtual
machine (SVM), and whether passthrough-read for offline files is enabled.

What you'll need

- The FPolicy policy worksheet should be completed.

- If you plan on configuring the policy to use FPolicy servers, the external engine must exist.

- At least one FPolicy event that you plan on associating with the FPolicy policy must exist.

- If you want to configure privileged data access, a SMB server must exist on the SVM.

- To configure a persistent store for a policy, the engine type must be **async** and the policy must be **non-mandatory**.

  For more information, see Create persistent stores.

Steps

1. Create the FPolicy policy:

```
vserver fpolicy policy create -vserver-name vserver_name -policy-name
```

```
policy_name -engine engine_name -events event_name, [-persistent-store
PS_name] [-is-mandatory {true|false}] [-allow-privileged-access {yes|no}] [-
privileged-user-name domain\user_name] [-is-passthrough-read-enabled
{true|false}]
```

- You can add one or more events to the FPolicy policy.

- By default, mandatory screening is enabled.

- If you want to allow privileged access by setting the `-allow-privileged-access` parameter to `yes`, you must also configure a privileged user name for privileged access.

- If you want to configure passthrough-read by setting the `-is-passthrough-read-enabled` parameter to `true`, you must also configure privileged data access.

  The following command creates a policy named "policy1" that has the event named "event1" and the external engine named "engine1" associated with it. This policy uses default values in the policy configuration: `vserver fpolicy policy create -vserver vs1.example.com -policy -name policy1 -events event1 -engine engine1`

  The following command creates a policy named "policy2" that has the event named "event2" and the external engine named "engine2" associated with it. This policy is configured to use privileged access using the specified user name. Passthrough-read is enabled:

  ```
  vserver fpolicy policy create -vserver vs1.example.com -policy-name policy2
  -events event2 -engine engine2 -allow-privileged-access yes -privileged-
  user-name example\archive_acct -is-passthrough-read-enabled true
  ```

  The following command creates a policy named "native1" that has the event named "event3" associated with it. This policy uses the native engine and uses default values in the policy configuration:

  ```
  vserver fpolicy policy create -vserver vs1.example.com -policy-name native1
  -events event3 -engine native
  ```

2. Verify the FPolicy policy configuration by using the `vserver fpolicy policy show` command.

   The following command displays information about the three configured FPolicy policies, including the following information:

   - The SVM associated with the policy

   - The external engine associated with the policy

   - The events associated with the policy

   - Whether mandatory screening is required

   - Whether privileged access is required `vserver fpolicy policy show`

```
   Vserver        Policy      Events    Engine    Is Mandatory  Privileged
                  Name                                          Access
-------------- --------- --------- --------- ------------
-----------
vs1.example.com  policy1    event1    engine1   true          no
vs1.example.com  policy2    event2    engine2   true          yes
vs1.example.com  native1    event3    native    true          no
```

Create the FPolicy scope

After creating the FPolicy policy, you need to create an FPolicy scope. When creating the scope, you associate the scope with an FPolicy policy. A scope defines the boundaries on which the FPolicy policy applies. Scopes can include or exclude files based on shares, export policies, volumes, and file extensions.

What you'll need

The FPolicy scope worksheet must be completed. The FPolicy policy must exist with an associated external engine (if the policy is configured to use external FPolicy servers) and must have at least one associated FPolicy event.

Steps

1. Create the FPolicy scope by using the `vserver fpolicy policy scope create` command.

   ```
   vserver fpolicy policy scope create -vserver-name vs1.example.com -policy-name
   policy1 -volumes-to-include datavol1,datavol2
   ```

2. Verify the FPolicy scope configuration by using the `vserver fpolicy policy scope show` command.

   ```
   vserver fpolicy policy scope show -vserver vs1.example.com -instance
   ```

   ```
                        Vserver: vs1.example.com
                         Policy: policy1
            Shares to Include: -
            Shares to Exclude: -
           Volumes to Include: datavol1, datavol2
           Volumes to Exclude: -
   Export Policies to Include: -
   Export Policies to Exclude: -
   File Extensions to Include: -
   File Extensions to Exclude: -
   ```

Enable the FPolicy policy

After you are through configuring an FPolicy policy configuration, you enable the FPolicy

policy. Enabling the policy sets its priority and starts file access monitoring for the policy.

What you'll need

The FPolicy policy must exist with an associated external engine (if the policy is configured to use external FPolicy servers) and must have at least one associated FPolicy event. The FPolicy policy scope must exist and must be assigned to the FPolicy policy.

About this task

The priority is used when multiple policies are enabled on the storage virtual machine (SVM) and more than one policy has subscribed to the same file access event. Policies that use the native engine configuration have a higher priority than policies for any other engine, regardless of the sequence number assigned to them when enabling the policy.

> ℹ️ A policy cannot be enabled on the admin SVM.

Steps

1. Enable the FPolicy policy by using the `vserver fpolicy enable` command.

   ```
   vserver fpolicy enable -vserver-name vs1.example.com -policy-name policy1
   -sequence-number 1
   ```

2. Verify that the FPolicy policy is enabled by using the `vserver fpolicy show` command.

   ```
   vserver fpolicy show -vserver vs1.example.com
   ```

   ```
                                   Sequence
    Vserver           Policy Name      Number  Status    Engine
    ---------------   -----------------  --------  --------  ---------
    vs1.example.com   policy1                 1  on        engine1
   ```

Manage FPolicy configurations

Modify FPolicy configurations

Commands for modifying FPolicy configurations

You can modify FPolicy configurations by modifying the elements that make up the configuration. You can modify external engines, FPolicy events, FPolicy scopes, and FPolicy policies. You can also enable or disable FPolicy policies. When you disable the FPolicy policy, file monitoring is discontinued for that policy.

It is recommended to disable the FPolicy policy before modifying the configuration.

If you want to modify…	Use this command…
External engines	`vserver fpolicy policy external-engine modify`

Events	`vserver fpolicy policy event modify`
Scopes	`vserver fpolicy policy scope modify`
Policies	`vserver fpolicy policy modify`

See the man pages for the commands for more information.

Enable or disable FPolicy policies

You can enable FPolicy policies after the configuration is complete. Enabling the policy sets its priority and starts file access monitoring for the policy. You can disable FPolicy policies if you want to stop file access monitoring for the policy.

What you'll need

Before enabling FPolicy policies, the FPolicy configuration must be completed.

About this task

- The priority is used when multiple policies are enabled on the storage virtual machine (SVM) and more than one policy has subscribed to the same file access event.
- Policies that use the native engine configuration have a higher priority than policies for any other engine, regardless of the sequence number assigned to them when enabling the policy.
- If you want to change the priority of an FPolicy policy, you must disable the policy and then reenable it using the new sequence number.

Step

1. Perform the appropriate action:

If you want to…	Enter the following command…
Enable an FPolicy policy	`vserver fpolicy enable -vserver-name vserver_name -policy-name policy_name -sequence-number integer`
Disable an FPolicy policy	`vserver fpolicy disable -vserver-name vserver_name -policy-name policy_name`

Display information about FPolicy configurations

How the show commands work

It is helpful when displaying information about the FPolicy configuration to understand how the `show` commands work.

A `show` command without additional parameters displays information in a summary form. Additionally, every `show` command has the same two mutually exclusive optional parameters, `-instance` and `-fields`.

When you use the `-instance` parameter with a `show` command, the command output displays detailed information in a list format. In some cases, the detailed output can be lengthy and include more information than you need. You can use the `-fields fieldname[,fieldname…]` parameter to customize the output so that it displays information only for the fields you specify. You can identity which fields that you can specify by entering `?` after the `-fields` parameter.

> (i) The output of a `show` command with the `-fields` parameter might display other relevant and necessary fields related to the requested fields.

Every `show` command has one or more optional parameters that filter that output and enable you to narrow the scope of information displayed in command output. You can identity which optional parameters are available for a command by entering `?` after the `show` command.

The `show` command supports UNIX-style patterns and wildcards to enable you to match multiple values in command-parameters arguments. For example, you can use the wildcard operator (*), the NOT operator (!), the OR operator (|), the range operator (integer…integer), the less-than operator (<), the greater-than operator (>), the less-than or equal to operator (<=), and the greater-than or equal to operator (>=) when specifying values.

For more information about using UNIX-style patterns and wildcards, see the Using the ONTAP command-line interface.

Commands for displaying information about FPolicy configurations

You use the `fpolicy show` commands to display information about the FPolicy configuration, including information about FPolicy external engines, events, scopes, and policies.

If you want to display information about FPolicy…	Use this command…
External engines	`vserver fpolicy policy external-engine show`
Events	`vserver fpolicy policy event show`
Scopes	`vserver fpolicy policy scope show`
Policies	`vserver fpolicy policy show`

See the man pages for the commands for more information.

Display information about FPolicy policy status

You can display information about the status for FPolicy policies to determine whether a policy is enabled, what external engine it is configured to use, what the sequence number is for the policy, and to which storage virtual machine (SVM) the FPolicy policy is associated.

About this task

If you do not specify any parameters, the command displays the following information:

- SVM name
- Policy name
- Policy sequence number
- Policy status

In addition to displaying information about policy status for FPolicy policies configured on the cluster or a specific SVM, you can use command parameters to filter the command's output by other criteria.

You can specify the `-instance` parameter to display detailed information about listed policies. Alternatively, you can use the `-fields` parameter to display only the indicated fields in the command output, or `-fields ?` to determine what fields you can use.

Step

1. Display filtered information about FPolicy policy status by using the appropriate command:

If you want to display status information about policies…	Enter the command…
On the cluster	`vserver fpolicy show`
That have the specified status	`vserver fpolicy show -status {on|off}`
On a specified SVM	`vserver fpolicy show -vserver vserver_name`
With the specified policy name	`vserver fpolicy show -policy-name policy_name`
That use the specified external engine	`vserver fpolicy show -engine engine_name`

Example

The following example displays the information about FPolicy policies on the cluster:

```
cluster1::> vserver fpolicy show

                                        Sequence
Vserver              Policy Name          Number  Status     Engine
-------------------  -------------------  --------  ---------  ---------
FPolicy              cserver_policy        -        off        eng1
vs1.example.com      v1p1                  -        off        eng2
vs1.example.com      v1p2                  -        off        native
vs1.example.com      v1p3                  -        off        native
vs1.example.com      cserver_policy        -        off        eng1
vs2.example.com      v1p1                  3        on         native
vs2.example.com      v1p2                  1        on         eng3
vs2.example.com      cserver_policy        2        on         eng1
```

Display information about enabled FPolicy policies

You can display information about enabled FPolicy policies to determine what FPolicy external engine it is configured to use, what the priority is for the policy, and to which storage virtual machine (SVM) the FPolicy policy is associated.

About this task

If you do not specify any parameters, the command displays the following information:

- SVM name
- Policy name
- Policy priority

You can use command parameters to filter the command's output by specified criteria.

Step

1. Display information about enabled FPolicy policies by using the appropriate command:

If you want to display information about enabled policies…	Enter the command…
On the cluster	`vserver fpolicy show-enabled`
On a specified SVM	`vserver fpolicy show-enabled -vserver vserver_name`
With the specified policy name	`vserver fpolicy show-enabled -policy-name policy_name`
With the specified sequence number	`vserver fpolicy show-enabled -priority integer`

Example

The following example displays the information about enabled FPolicy policies on the cluster:

```
cluster1::> vserver fpolicy show-enabled
Vserver                 Policy Name              Priority

----------------------- ------------------------ ----------
vs1.example.com         pol_native               native
vs1.example.com         pol_native2              native
vs1.example.com         pol1                     2
vs1.example.com         pol2                     4
```

Manage FPolicy server connections

Connect to external FPolicy servers

To enable file processing, you might need to manually connect to an external FPolicy server if the connection has previously been terminated. A connection is terminated after the server timeout is reached or due to some error. Alternatively, the administrator might manually terminate a connection.

About this task

If a fatal error occurs, the connection to the FPolicy server can be terminated. After resolving the issue that caused the fatal error, you must manually reconnect to the FPolicy server.

Steps

1. Connect to the external FPolicy server by using the `vserver fpolicy engine-connect` command.

   For more information about the command, see the man pages.

2. Verify that the external FPolicy server is connected by using the `vserver fpolicy show-engine` command.

   For more information about the command, see the man pages.

Disconnect from external FPolicy servers

You might need to manually disconnect from an external FPolicy server. This might be desirable if the FPolicy server has issues with notification request processing or if you need to perform maintenance on the FPolicy server.

Steps

1. Disconnect from the external FPolicy server by using the `vserver fpolicy engine-disconnect` command.

   For more information about the command, see the man pages.

2. Verify that the external FPolicy server is disconnected by using the `vserver fpolicy show-engine` command.

For more information about the command, see the man pages.

Display information about connections to external FPolicy servers

You can display status information about connections to external FPolicy servers (FPolicy servers) for the cluster or for a specified storage virtual machine (SVM). This information can help you determine which FPolicy servers are connected.

About this task

If you do not specify any parameters, the command displays the following information:

- SVM name
- Node name
- FPolicy policy name
- FPolicy server IP address
- FPolicy server status
- FPolicy server type

In addition to displaying information about FPolicy connections on the cluster or a specific SVM, you can use command parameters to filter the command's output by other criteria.

You can specify the `-instance` parameter to display detailed information about listed policies. Alternatively, you can use the `-fields` parameter to display only the indicated fields in the command output. You can enter `?` after the `-fields` parameter to find out which fields you can use.

Step

1. Display filtered information about connection status between the node and the FPolicy server by using the appropriate command:

If you want to display connection status information about FPolicy servers…	Enter…
That you specify	`vserver fpolicy show-engine -server IP_address`
For a specified SVM	`vserver fpolicy show-engine -vserver vserver_name`
That are attached with a specified policy	`vserver fpolicy show-engine -policy-name policy_name`

With the server status that you specify	`vserver fpolicy show-engine -server-status status` The server status can be one of the following: - `connected` - `disconnected` - `connecting` - `disconnecting`
With the specified type	`vserver fpolicy show-engine -server-type type` The FPolicy server type can be one of the following: - `primary` - `secondary`
That were disconnected with the specified reason	`vserver fpolicy show-engine -disconnect-reason text` Disconnect can be due to multiple reasons. The following are common reasons for disconnect: - `Disconnect command received from CLI.` - `Error encountered while parsing notification response from FPolicy server.` - `FPolicy Handshake failed.` - `SSL handshake failed.` - `TCP Connection to FPolicy server failed.` - `The screen response message received from the FPolicy server is not valid.`

Example

This example displays information about external engine connections to FPolicy servers on SVM vs1.example.com:

```
cluster1::> vserver fpolicy show-engine -vserver vs1.example.com
FPolicy                                         Server-      Server-
Vserver          Policy    Node        Server        status       type
--------------- --------- ------------ ------------- -------------
---------
vs1.example.com policy1    node1       10.1.1.2      connected    primary
vs1.example.com policy1    node1       10.1.1.3      disconnected primary
vs1.example.com policy1    node2       10.1.1.2      connected    primary
vs1.example.com policy1    node2       10.1.1.3      disconnected primary
```

This example displays information only about connected FPolicy servers:

```
cluster1::> vserver fpolicy show-engine -fields server -server-status
connected
node        vserver          policy-name server
---------- --------------- ----------- -------
node1       vs1.example.com policy1      10.1.1.2
node2       vs1.example.com policy1      10.1.1.2
```

Display information about the FPolicy passthrough-read connection status

You can display information about FPolicy passthrough-read connection status to external
FPolicy servers (FPolicy servers) for the cluster or for a specified storage virtual machine
(SVM). This information can help you determine which FPolicy servers have
passthrough-read data connections and for which FPolicy servers the passthrough-read
connection is disconnected.

About this task

If you do not specify any parameter, the command displays the following information:

- SVM name
- FPolicy policy name
- Node name
- FPolicy server IP address
- FPolicy passthrough-read connection status

In addition to displaying information about FPolicy connections on the cluster or a specific SVM, you can use
command parameters to filter the command's output by other criteria.

You can specify the `-instance` parameter to display detailed information about listed policies. Alternatively,
you can use the `-fields` parameter to display only the indicated fields in the command output. You can enter
`?` after the `-fields` parameter to find out which fields you can use.

Step

1. Display filtered information about connection status between the node and the FPolicy server by using the

appropriate command:

If you want to display connection status information about…	Enter the command…
FPolicy passthrough-read connection status for the cluster	`vserver fpolicy show-passthrough-read-connection`
FPolicy passthrough-read connection status for a specified SVM	`vserver fpolicy show-passthrough-read-connection -vserver vserver_name`
FPolicy passthrough-read connection status for a specified policy	`vserver fpolicy show-passthrough-read-connection -policy-name policy_name`
Detailed FPolicy passthrough-read connection status for a specified policy	`vserver fpolicy show-passthrough-read-connection -policy-name policy_name -instance`
FPolicy passthrough-read connection status for the status that you specify	`vserver fpolicy show-passthrough-read-connection -policy-name policy_name -server-status status` The server status can be one of the following: • `connected` • `disconnected`

Example

The following command displays information about passthrough-read connections from all FPolicy servers on the cluster:

```
cluster1::> vserver fpolicy show-passthrough-read-connection
                                       FPolicy          Server
Vserver           Policy Name   Node          Server           Status
---------------   ------------ ------------ -----------------
--------------
vs2.example.com  pol_cifs_2    FPolicy-01   2.2.2.2           disconnected
vs1.example.com  pol_cifs_1    FPolicy-01   1.1.1.1           connected
```

The following command displays detailed information about passthrough-read connections from FPolicy servers configured in the "pol_cifs_1" policy:

```
cluster1::> vserver fpolicy show-passthrough-read-connection -policy-name
pol_cifs_1 -instance

                                     Node: FPolicy-01
                                  Vserver: vs1.example.com
                                   Policy: pol_cifs_1
                                   Server: 1.1.1.1
            Session ID of the Control Channel: 8cef052e-2502-11e3-
88d4-123478563412
                            Server Status: connected
      Time Passthrough Read Channel was Connected: 9/24/2013 10:17:45
   Time Passthrough Read Channel was Disconnected: -
Reason for Passthrough Read Channel Disconnection: none
```