Proteggi le app

Astra Control Center

NetApp
November 21, 2023

Sommario

Proteggi le app

Panoramica della protezione

Con Astra Control Center puoi creare backup, cloni, snapshot e policy di protezione per le tue applicazioni. Il backup delle tue applicazioni aiuta i tuoi servizi e i dati associati a essere il più possibile disponibili; durante uno scenario di disastro, il ripristino dal backup può garantire il ripristino completo di un'applicazione e dei dati associati con interruzioni minime. Backup, cloni e snapshot possono contribuire a proteggere da minacce comuni come ransomware, perdita accidentale di dati e disastri ambientali. "Scopri i tipi di protezione dei dati disponibili in Astra Control Center e quando utilizzarli".

Workflow di protezione delle app

Puoi utilizzare il seguente flusso di lavoro di esempio per iniziare a proteggere le tue applicazioni.

[Uno] Eseguire il backup di tutte le applicazioni

Per garantire la protezione immediata delle applicazioni, "creare un backup manuale di tutte le applicazioni".

[Due] Configurare una policy di protezione per ogni applicazione

Per automatizzare backup e snapshot futuri, "configurare una policy di protezione per ogni applicazione". Ad esempio, è possibile iniziare con backup settimanali e snapshot giornalieri, con un mese di conservazione per entrambi. Si consiglia vivamente di automatizzare backup e snapshot con una policy di protezione rispetto a backup e snapshot manuali.

[Tre] Facoltativo: Regolare i criteri di protezione

Man mano che le applicazioni e i loro modelli di utilizzo cambiano, regola le policy di protezione in base alle necessità per fornire la migliore protezione.

[Quattro] In caso di disastro, ripristinate le vostre applicazioni

In caso di perdita di dati, è possibile eseguire il ripristino "ripristino del backup più recente" primo per ogni applicazione. È quindi possibile ripristinare l'ultimo snapshot (se disponibile).

Proteggi le app con snapshot e backup

Proteggi le tue applicazioni eseguendo snapshot e backup utilizzando una policy di protezione automatica o ad-hoc. È possibile utilizzare l'interfaccia utente Astra o. "L'API Astra Control" per proteggere le applicazioni.

> Se utilizzi Helm per implementare le app, Astra Control Center richiede Helm versione 3. La gestione e la clonazione delle applicazioni implementate con Helm 3 (o aggiornate da Helm 2 a Helm 3) sono completamente supportate. Le app implementate con Helm 2 non sono supportate.

Quando crei un progetto per ospitare un'applicazione su un cluster OpenShift, al progetto (o namespace Kubernetes) viene assegnato un UID SecurityContext. Per consentire ad Astra Control Center di proteggere la tua applicazione e spostarla in un altro cluster o progetto in OpenShift, devi aggiungere policy che consentano all'applicazione di essere eseguita come qualsiasi UID. Ad esempio, i seguenti comandi CLI di OpenShift concedono le policy appropriate a un'applicazione WordPress.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

Configurare un criterio di protezione

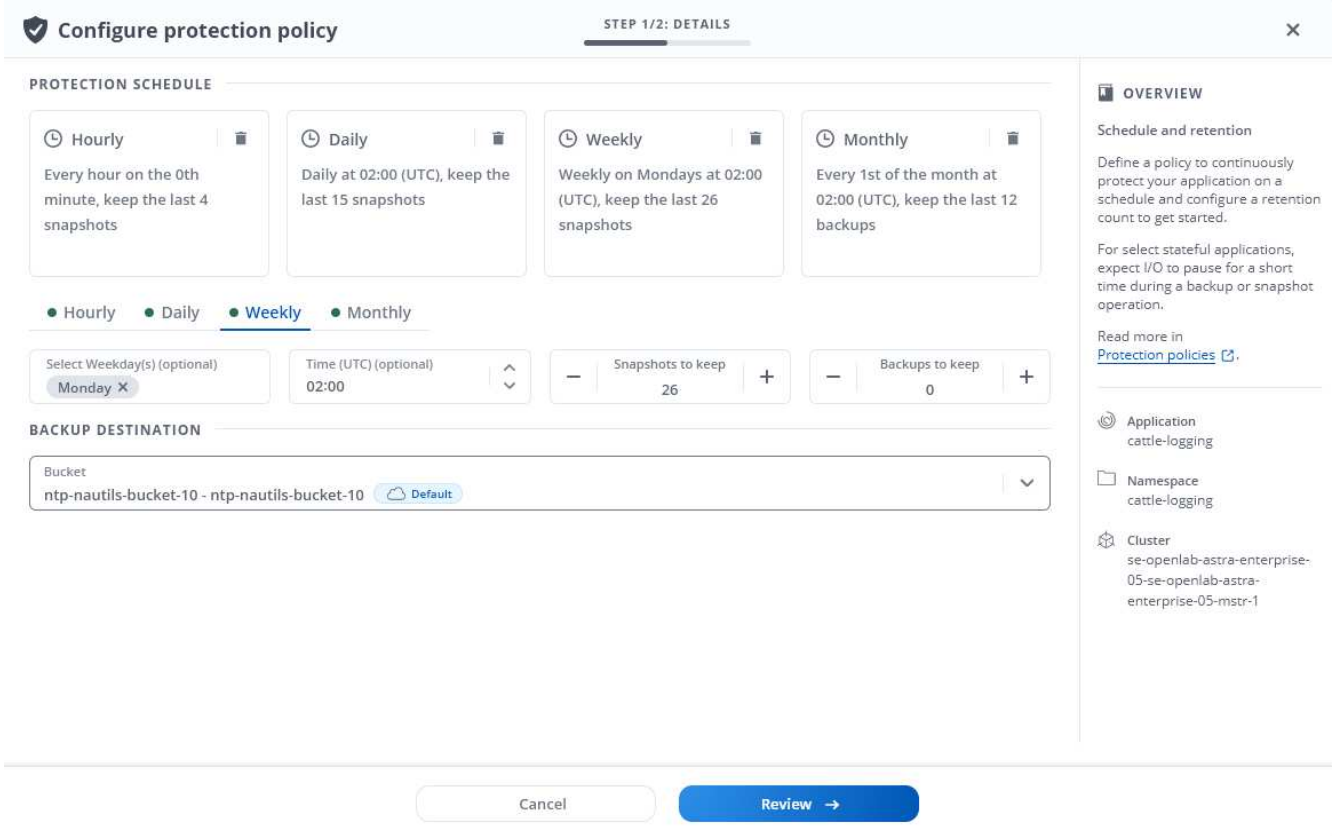
Una policy di protezione protegge un'applicazione creando snapshot, backup o entrambi in base a una pianificazione definita. È possibile scegliere di creare snapshot e backup ogni ora, ogni giorno, ogni settimana e ogni mese, nonché specificare il numero di copie da conservare. Ad esempio, una policy di protezione potrebbe creare backup settimanali e snapshot giornalieri e conservare backup e snapshot per un mese. La frequenza con cui vengono creati snapshot e backup e la durata della conservazione dipendono dalle esigenze dell'organizzazione.

Fasi

1. Selezionare **applicazioni**, quindi selezionare il nome di un'applicazione.
2. Selezionare **Data Protection** (protezione dati).
3. Selezionare **Configura policy di protezione**.
4. Definire un programma di protezione scegliendo il numero di snapshot e backup da conservare ogni ora, ogni giorno, ogni settimana e ogni mese.

   È possibile definire le pianificazioni orarie, giornaliere, settimanali e mensili contemporaneamente. Un programma non diventa attivo fino a quando non viene impostato un livello di conservazione.

   Nell'esempio seguente vengono impostati quattro programmi di protezione: ogni ora, ogni giorno, ogni settimana e ogni mese per snapshot e backup.



5. Selezionare **Revisione**.
6. Selezionare **Imposta policy di protezione.**

Risultato

Astra Control Center implementa la policy di protezione dei dati creando e conservando snapshot e backup utilizzando i criteri di pianificazione e conservazione definiti dall'utente.

Creare un'istantanea

Puoi creare uno snapshot on-demand in qualsiasi momento.

Fasi

1. Selezionare **applicazioni**.
2. Dal menu Options (Opzioni) nella colonna **Actions** (azioni) dell'applicazione desiderata, selezionare **Snapshot**.
3. Personalizzare il nome dell'istantanea, quindi selezionare **Review** (Rivedi).
4. Esaminare il riepilogo dell'istantanea e selezionare **Snapshot**.

Risultato

Viene avviato il processo di snapshot. Un'istantanea viene eseguita correttamente quando lo stato è **Available** nella colonna **Actions** nella pagina **Data Protection** > **Snapshots**.

Creare un backup

Puoi anche eseguire il backup di un'applicazione in qualsiasi momento.

I bucket S3 in Astra Control Center non riportano la capacità disponibile. Prima di eseguire il backup o la clonazione delle applicazioni gestite da Astra Control Center, controllare le informazioni del bucket nel sistema di gestione ONTAP o StorageGRID.

Fasi

1. Selezionare **applicazioni**.
2. Dal menu Options (Opzioni) nella colonna **Actions** (azioni) dell'applicazione desiderata, selezionare **Backup**.
3. Personalizzare il nome del backup.
4. Scegliere se eseguire il backup dell'applicazione da uno snapshot esistente. Se si seleziona questa opzione, è possibile scegliere da un elenco di snapshot esistenti.
5. Scegliere una destinazione per il backup selezionandola dall'elenco dei bucket di storage.
6. Selezionare **Revisione**.
7. Esaminare il riepilogo del backup e selezionare **Backup**.

Risultato

Astra Control Center crea un backup dell'applicazione.

Se la rete presenta un'interruzione o è eccessivamente lenta, potrebbe verificarsi un timeout dell'operazione di backup. In questo modo, il backup non viene eseguito correttamente.

Non esiste alcun modo per interrompere un backup in esecuzione. Se è necessario eliminare il backup, attendere che sia stato completato, quindi seguire le istruzioni riportate in Eliminare i backup. Per eliminare un backup non riuscito, "Utilizzare l'API di controllo Astra".

Dopo un'operazione di protezione dei dati (clone, backup, ripristino) e il successivo ridimensionamento persistente del volume, si verifica un ritardo di venti minuti prima che le nuove dimensioni del volume vengano visualizzate nell'interfaccia utente. L'operazione di protezione dei dati viene eseguita correttamente in pochi minuti ed è possibile utilizzare il software di gestione per il back-end dello storage per confermare la modifica delle dimensioni del volume.

Visualizzare snapshot e backup

È possibile visualizzare le istantanee e i backup di un'applicazione dalla scheda Data Protection (protezione dati).

Fasi

1. Selezionare **applicazioni**, quindi selezionare il nome di un'applicazione.
2. Selezionare **Data Protection** (protezione dati).

   Le istantanee vengono visualizzate per impostazione predefinita.

3. Selezionare **Backup** per visualizzare l'elenco dei backup.

Eliminare le istantanee

Eliminare le snapshot pianificate o on-demand non più necessarie.

Fasi

1. Selezionare **applicazioni**, quindi selezionare il nome di un'applicazione.
2. Selezionare **Data Protection** (protezione dati).
3. Dal menu Options (Opzioni) nella colonna **Actions** (azioni) per lo snapshot desiderato, selezionare **Delete snapshot** (Elimina snapshot).
4. Digitare la parola "DELETE" per confermare l'eliminazione, quindi selezionare **Yes, Delete snapshot**.

Risultato

Astra Control Center elimina lo snapshot.

Eliminare i backup

Eliminare i backup pianificati o on-demand non più necessari.

> Non esiste alcun modo per interrompere un backup in esecuzione. Se è necessario eliminare il backup, attendere che sia stato completato, quindi seguire queste istruzioni. Per eliminare un backup non riuscito, "Utilizzare l'API di controllo Astra".

1. Selezionare **applicazioni**, quindi selezionare il nome di un'applicazione.
2. Selezionare **Data Protection** (protezione dati).
3. Selezionare **Backup**.
4. Dal menu Options (Opzioni) nella colonna **Actions** (azioni) per il backup desiderato, selezionare **Delete backup** (Elimina backup).
5. Digitare la parola "DELETE" per confermare l'eliminazione, quindi selezionare **Yes, Delete backup**.

Risultato

Astra Control Center elimina il backup.

Ripristinare le applicazioni

Astra Control può ripristinare l'applicazione da uno snapshot o da un backup. Il ripristino da uno snapshot esistente sarà più rapido quando si ripristina l'applicazione nello stesso cluster. È possibile utilizzare l'interfaccia utente di Astra Control o. "L'API Astra Control" per ripristinare le applicazioni.

A proposito di questa attività

- Si consiglia vivamente di eseguire un'istantanea o un backup dell'applicazione prima di ripristinarla. In questo modo, è possibile clonare lo snapshot o il backup nel caso in cui il ripristino non abbia esito positivo.
- Se utilizzi Helm per implementare le app, Astra Control Center richiede Helm versione 3. La gestione e la clonazione delle applicazioni implementate con Helm 3 (o aggiornate da Helm 2 a Helm 3) sono completamente supportate. Le app implementate con Helm 2 non sono supportate.
- Se si esegue il ripristino in un cluster diverso, assicurarsi che il cluster utilizzi la stessa modalità di accesso al volume persistente (ad esempio ReadWriteMany). L'operazione di ripristino non riesce se la modalità di accesso al volume persistente di destinazione è diversa.
- Qualsiasi utente membro con vincoli di spazio dei nomi in base al nome/ID dello spazio dei nomi o alle etichette dello spazio dei nomi può clonare o ripristinare un'applicazione in un nuovo spazio dei nomi nello

stesso cluster o in qualsiasi altro cluster dell'account dell'organizzazione. Tuttavia, lo stesso utente non può accedere all'applicazione clonata o ripristinata nel nuovo namespace. Dopo la creazione di un nuovo spazio dei nomi mediante un'operazione di clone o ripristino, l'amministratore/proprietario dell'account può modificare l'account utente membro e aggiornare i vincoli di ruolo per consentire all'utente interessato di accedere al nuovo spazio dei nomi.

- Quando crei un progetto per ospitare un'applicazione su un cluster OpenShift, al progetto (o namespace Kubernetes) viene assegnato un UID SecurityContext. Per consentire ad Astra Control Center di proteggere la tua applicazione e spostarla in un altro cluster o progetto in OpenShift, devi aggiungere policy che consentano all'applicazione di essere eseguita come qualsiasi UID. Ad esempio, i seguenti comandi CLI di OpenShift concedono le policy appropriate a un'applicazione WordPress.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

Fasi

1. Selezionare **applicazioni**, quindi selezionare il nome di un'applicazione.
2. Selezionare **Data Protection**.
3. Se si desidera eseguire il ripristino da uno snapshot, tenere selezionata l'icona **Snapshot**. In caso contrario, selezionare l'icona **Backup** per eseguire il ripristino da un backup.
4. Dal menu Options (Opzioni) nella colonna **Actions** (azioni) per lo snapshot o il backup da cui si desidera eseguire il ripristino, selezionare **Restore application** (Ripristina applicazione).
5. **Restore details** (Dettagli ripristino): Specificare i dettagli dell'applicazione ripristinata. Per impostazione predefinita, vengono visualizzati il cluster e lo spazio dei nomi correnti. Lasciare intatti questi valori per ripristinare un'applicazione in-place, che ripristina l'applicazione a una versione precedente di se stessa. Modificare questi valori se si desidera ripristinare un cluster o uno spazio dei nomi diverso.
   - Immettere un nome e uno spazio dei nomi per l'applicazione.
   - Scegliere il cluster di destinazione per l'applicazione.
   - Selezionare **Revisione**.

   > Se si ripristina uno spazio dei nomi precedentemente cancellato, viene creato un nuovo spazio dei nomi con lo stesso nome come parte del processo di ripristino. Tutti gli utenti che disponevano dei diritti per gestire le applicazioni nello spazio dei nomi precedentemente cancellato devono ripristinare manualmente i diritti nello spazio dei nomi appena ricreato.

6. **Restore Summary** (Riepilogo ripristino): Esaminare i dettagli relativi all'azione di ripristino, digitare "restore" e selezionare **Restore**.

Risultato

Astra Control Center ripristina l'applicazione in base alle informazioni fornite. Se hai ripristinato l'applicazione in-place, il contenuto di eventuali volumi persistenti esistenti viene sostituito con il contenuto dei volumi persistenti dell'applicazione ripristinata.

Dopo un'operazione di protezione dei dati (cloning, backup, ripristino) e il successivo ridimensionamento persistente del volume, si verifica un ritardo fino a venti minuti prima che la nuova dimensione del volume venga visualizzata nell'interfaccia utente Web. L'operazione di protezione dei dati viene eseguita correttamente in pochi minuti ed è possibile utilizzare il software di gestione per il back-end dello storage per confermare la modifica delle dimensioni del volume.

Clonare e migrare le applicazioni

Clonare un'applicazione esistente per creare un'applicazione duplicata sullo stesso cluster Kubernetes o su un altro cluster. Quando Astra Control Center clona un'applicazione, crea un clone della configurazione dell'applicazione e dello storage persistente.

La clonazione può essere di aiuto nel caso in cui sia necessario spostare applicazioni e storage da un cluster Kubernetes a un altro. Ad esempio, è possibile spostare i carichi di lavoro attraverso una pipeline ci/CD e attraverso gli spazi dei nomi Kubernetes. È possibile utilizzare l'interfaccia utente Astra o. "L'API Astra Control" per clonare e migrare le applicazioni.

Di cosa hai bisogno

Per clonare le applicazioni in un cluster diverso, è necessario un bucket predefinito. Quando si aggiunge il primo bucket, questo diventa quello predefinito.

A proposito di questa attività

- Se si implementa un'applicazione con un StorageClass esplicitamente impostato e si deve clonare l'applicazione, il cluster di destinazione deve avere la StorageClass specificata in origine. Il cloning di un'applicazione con un StorageClass esplicitamente impostato su un cluster che non ha lo stesso StorageClass avrà esito negativo.
- Se si clonano istanze distribuite dall'operatore di Jenkins ci, è necessario ripristinare manualmente i dati persistenti. Si tratta di un limite del modello di implementazione dell'applicazione.
- I bucket S3 in Astra Control Center non riportano la capacità disponibile. Prima di eseguire il backup o la clonazione delle applicazioni gestite da Astra Control Center, controllare le informazioni del bucket nel sistema di gestione ONTAP o StorageGRID.
- Durante il backup di un'applicazione o il ripristino di un'applicazione, è possibile specificare un ID bucket. Un'operazione di cloni dell'applicazione, tuttavia, utilizza sempre il bucket predefinito definito. Non esiste alcuna opzione per modificare i bucket per un clone. Se si desidera controllare quale bucket viene utilizzato, è possibile farlo "modificare l'impostazione predefinita del bucket" oppure fare una "backup" seguito da un "ripristinare" separatamente.
- Qualsiasi utente membro con vincoli di spazio dei nomi in base al nome/ID dello spazio dei nomi o alle etichette dello spazio dei nomi può clonare o ripristinare un'applicazione in un nuovo spazio dei nomi nello stesso cluster o in qualsiasi altro cluster dell'account dell'organizzazione. Tuttavia, lo stesso utente non può accedere all'applicazione clonata o ripristinata nel nuovo namespace. Dopo la creazione di un nuovo spazio dei nomi mediante un'operazione di clone o ripristino, l'amministratore/proprietario dell'account può modificare l'account utente membro e aggiornare i vincoli di ruolo per consentire all'utente interessato di accedere al nuovo spazio dei nomi.

Considerazioni su OpenShift

- Se clonate un'applicazione tra cluster, i cluster di origine e di destinazione devono essere della stessa distribuzione di OpenShift. Ad esempio, se clonate un'applicazione da un cluster OpenShift 4.7, utilizzate un cluster di destinazione che è anche OpenShift 4.7.

- Quando crei un progetto per ospitare un'applicazione su un cluster OpenShift, al progetto (o namespace Kubernetes) viene assegnato un UID SecurityContext. Per consentire ad Astra Control Center di proteggere la tua applicazione e spostarla in un altro cluster o progetto in OpenShift, devi aggiungere policy che consentano all'applicazione di essere eseguita come qualsiasi UID. Ad esempio, i seguenti comandi CLI di OpenShift concedono le policy appropriate a un'applicazione WordPress.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

Fasi

1. Selezionare **applicazioni**.
2. Effettuare una delle seguenti operazioni:
   - Selezionare il menu Options (Opzioni) nella colonna **Actions** (azioni) per l'applicazione desiderata.
   - Selezionare il nome dell'applicazione desiderata e l'elenco a discesa Status (Stato) in alto a destra nella pagina.
3. Selezionare **Clone**.
4. **Clone Details**: Specificare i dettagli per il clone:
   - Immettere un nome.
   - Immettere uno spazio dei nomi per il clone.
   - Scegliere un cluster di destinazione per il clone.
   - Scegliere se si desidera creare il clone da uno snapshot o da un backup esistente. Se non si seleziona questa opzione, Astra Control Center crea il clone dallo stato corrente dell'applicazione.
5. **Origine**: Se si sceglie di clonare da uno snapshot o da un backup esistente, scegliere lo snapshot o il backup che si desidera utilizzare.
6. Selezionare **Revisione**.
7. **Clone Summary**: Leggi i dettagli sul clone e seleziona **Clone**.

Risultato

Astra Control Center clona l'applicazione in base alle informazioni fornite. L'operazione di clonazione viene eseguita correttamente quando il nuovo clone dell'applicazione si trova in `Available` nella pagina **applicazioni**.

> Dopo un'operazione di protezione dei dati (clone, backup, ripristino) e il successivo ridimensionamento persistente del volume, si verifica un ritardo di venti minuti prima che le nuove dimensioni del volume vengano visualizzate nell'interfaccia utente. L'operazione di protezione dei dati viene eseguita correttamente in pochi minuti ed è possibile utilizzare il software di gestione per il back-end dello storage per confermare la modifica delle dimensioni del volume.

Gestire gli hook di esecuzione delle applicazioni

Un gancio di esecuzione è uno script personalizzato che è possibile eseguire prima o dopo uno snapshot di un'applicazione gestita. Ad esempio, se si dispone di un'applicazione di database, è possibile utilizzare i ganci di esecuzione per sospendere tutte le transazioni del database prima di uno snapshot e riprendere le transazioni dopo il

completamento dello snapshot. Ciò garantisce snapshot coerenti con l'applicazione.

Hook di esecuzione predefiniti ed espressioni regolari

Per alcune applicazioni, Astra Control viene fornito con gli hook di esecuzione predefiniti, forniti da NetApp, che gestiscono le operazioni di blocco e scongelamento prima e dopo le snapshot. Astra Control utilizza espressioni regolari per associare l'immagine container di un'applicazione a queste applicazioni:

- MariaDB
  - Espressione regolare corrispondente
- MySQL
  - Espressione regolare corrispondente
- PostgreSQL
  - Espressione regolare corrispondente

In caso di corrispondenza, gli hook di esecuzione predefiniti forniti da NetApp per l'applicazione vengono visualizzati nell'elenco degli hook di esecuzione attivi dell'applicazione, che vengono eseguiti automaticamente quando vengono eseguite le istantanee dell'applicazione. Se una delle applicazioni personalizzate ha un nome immagine simile che corrisponde a una delle espressioni regolari (e non si desidera utilizzare gli hook di esecuzione predefiniti), è possibile modificare il nome dell'immagine, oppure disattiva il gancio di esecuzione predefinito per l'applicazione e utilizza un gancio personalizzato.

Non è possibile eliminare o modificare gli hook di esecuzione predefiniti.

Note importanti sugli hook di esecuzione personalizzati

Quando si pianificano gli hook di esecuzione per le applicazioni, considerare quanto segue.

- Astra Control richiede che gli hook di esecuzione siano scritti nel formato degli script di shell eseguibili.
- La dimensione dello script è limitata a 128 KB.
- Astra Control utilizza le impostazioni di esecuzione degli hook e qualsiasi criterio di corrispondenza per determinare quali hook sono applicabili a uno snapshot.
- Tutti i guasti degli uncini di esecuzione sono guasti di tipo soft; altri hook e snapshot vengono ancora tentati anche se un hook non funziona. Tuttavia, quando un gancio non funziona, viene registrato un evento di avviso nel registro eventi della pagina **attività**.
- Per creare, modificare o eliminare gli hook di esecuzione, è necessario essere un utente con autorizzazioni Owner, Admin o Member.
- Se l'esecuzione di un gancio di esecuzione richiede più di 25 minuti, l'hook non riesce, creando una voce del registro eventi con un codice di ritorno "N/A". Qualsiasi snapshot interessata verrà contrassegnata come non riuscita e una voce del registro eventi risultante annoterà il timeout.

> Poiché gli hook di esecuzione spesso riducono o disattivano completamente le funzionalità dell'applicazione con cui vengono eseguiti, è consigliabile ridurre al minimo il tempo necessario per l'esecuzione degli hook di esecuzione personalizzati.

Quando viene eseguita una snapshot, gli eventi di esecuzione hook hanno luogo nel seguente ordine:

1. Tutti gli hook di esecuzione pre-snapshot predefiniti forniti da NetApp applicabili vengono eseguiti sui container appropriati.

2. Tutti gli hook di esecuzione pre-snapshot personalizzati applicabili vengono eseguiti sui container appropriati. È possibile creare ed eseguire tutti gli hook pre-snapshot personalizzati necessari, ma l'ordine di esecuzione di questi hook prima dell'istantanea non è garantito né configurabile.
3. Viene eseguita l'istantanea.
4. Tutti gli hook di esecuzione post-snapshot personalizzati applicabili vengono eseguiti sui container appropriati. È possibile creare ed eseguire tutti gli hook post-snapshot personalizzati necessari, ma l'ordine di esecuzione di questi hook dopo l'istantanea non è garantito né configurabile.
5. Tutti gli hook di esecuzione post-snapshot predefiniti forniti da NetApp applicabili vengono eseguiti sui container appropriati.

> Prima di abilitarli in un ambiente di produzione, è necessario verificare sempre gli script hook di esecuzione. È possibile utilizzare il comando 'kubectl exec' per testare comodamente gli script. Dopo aver attivato gli hook di esecuzione in un ambiente di produzione, testare le snapshot risultanti per assicurarsi che siano coerenti. Per eseguire questa operazione, clonare l'applicazione in uno spazio dei nomi temporaneo, ripristinare lo snapshot e quindi testare l'applicazione.

Visualizzare gli hook di esecuzione esistenti

È possibile visualizzare gli hook di esecuzione predefiniti personalizzati o forniti da NetApp per un'applicazione.

Fasi

1. Accedere a **applicazioni** e selezionare il nome di un'applicazione gestita.
2. Selezionare la scheda **Execution Hooks**.

   È possibile visualizzare tutti gli hook di esecuzione attivati o disattivati nell'elenco risultante. È possibile visualizzare lo stato, l'origine e il momento dell'esecuzione di un gancio (pre o post-snapshot). Per visualizzare i registri degli eventi che circondano gli hook di esecuzione, accedere alla pagina **Activity** nell'area di navigazione a sinistra.

Creare un gancio di esecuzione personalizzato

È possibile creare un gancio di esecuzione personalizzato per un'applicazione. Vedere "Esempi di gancio di esecuzione" per esempi di gancio. Per creare gli hook di esecuzione, è necessario disporre delle autorizzazioni Owner (Proprietario), Admin (Amministratore) o Member (membro).

> Quando si crea uno script shell personalizzato da utilizzare come uncino di esecuzione, ricordarsi di specificare la shell appropriata all'inizio del file, a meno che non si stiano eseguendo comandi linux o fornendo il percorso completo di un eseguibile.

Fasi

1. Selezionare **applicazioni**, quindi selezionare il nome di un'applicazione gestita.
2. Selezionare la scheda **Execution Hooks**.
3. Selezionare **Aggiungi un nuovo gancio**.
4. Nell'area **Dettagli gancio**, a seconda dell'esecuzione del gancio, scegliere **Pre-Snapshot** o **Post-Snapshot**.
5. Immettere un nome univoco per l'hook.

6. (Facoltativo) inserire gli argomenti da passare al gancio durante l'esecuzione, premendo il tasto Invio dopo ogni argomento inserito per registrarne ciascuno.
7. Nell'area **Container Images** (immagini container), se il gancio deve essere eseguito su tutte le immagini container contenute nell'applicazione, attivare la casella di controllo **Apply to all container images** (Applica a tutte le immagini container). Se invece il gancio dovrebbe agire solo su una o più immagini container specificate, inserire i nomi delle immagini container nel campo **nomi delle immagini container da abbinare**.
8. Nell'area **script**, eseguire una delle seguenti operazioni:
   - Caricare uno script personalizzato.
     i. Selezionare l'opzione **carica file**.
     ii. Selezionare un file e caricarlo.
     iii. Assegnare allo script un nome univoco.
     iv. (Facoltativo) inserire eventuali note che altri amministratori dovrebbero conoscere sullo script.
   - Incollare uno script personalizzato dagli Appunti.
     i. Selezionare l'opzione **Incolla dagli Appunti**.
     ii. Selezionare il campo di testo e incollare il testo dello script nel campo.
     iii. Assegnare allo script un nome univoco.
     iv. (Facoltativo) inserire eventuali note che altri amministratori dovrebbero conoscere sullo script.
9. Selezionare **Aggiungi gancio**.

Disattiva un gancio di esecuzione

È possibile disattivare un gancio di esecuzione se si desidera impedirne temporaneamente l'esecuzione prima o dopo un'istantanea di un'applicazione. Per disattivare gli hook di esecuzione, è necessario disporre delle autorizzazioni Owner, Admin o Member.

Fasi

1. Selezionare **applicazioni**, quindi selezionare il nome di un'applicazione gestita.
2. Selezionare la scheda **Execution Hooks**.
3. Selezionare il menu Options (Opzioni) nella colonna **Actions** (azioni) per un gancio che si desidera disattivare.
4. Selezionare **Disable** (Disattiva).

Eliminare un gancio di esecuzione

È possibile rimuovere completamente un gancio di esecuzione se non è più necessario. Per eliminare gli hook di esecuzione, è necessario disporre delle autorizzazioni Owner, Admin o Member.

Fasi

1. Selezionare **applicazioni**, quindi selezionare il nome di un'applicazione gestita.
2. Selezionare la scheda **Execution Hooks**.
3. Selezionare il menu Options (Opzioni) nella colonna **Actions** (azioni) per il gancio che si desidera eliminare.
4. Selezionare **Delete** (Elimina).

Esempi di gancio di esecuzione

USA i seguenti esempi per avere un'idea di come strutturare i tuoi hook di esecuzione. È possibile utilizzare questi ganci come modelli o come script di test.

Semplice esempio di successo

Questo è un esempio di un semplice hook che riesce e scrive un messaggio in output standard e in errore standard.

```
#!/bin/sh

# success_sample.sh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}
```

```
#
# main
#

# log something to stdout
info "running success_sample.sh"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

Semplice esempio di successo (versione bash)

Questo è un esempio di un semplice hook che riesce e scrive un messaggio in output standard e standard error, scritto per bash.

```
#!/bin/bash

# success_sample.bash
#
# A simple noop success hook script for testing purposes.
#
# args: None


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}
```

```
#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running success_sample.bash"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

Semplice esempio di successo (versione zsh)

Questo è un esempio di un semplice hook che riesce e scrive un messaggio in output standard e errore standard, scritto per la shell Z.

```
#!/bin/zsh

# success_sample.zsh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}
```

```
#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.zsh"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

Esempio di successo con argomenti

Nell'esempio riportato di seguito viene illustrato come utilizzare gli ARG in un gancio.

```
#!/bin/sh

# success_sample_args.sh
#
# A simple success hook script with args for testing purposes.
#
# args: Up to two optional args that are echoed to stdout

#
# Writes the given message to standard output
#
# $* - The message to write
#
```

```
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running success_sample_args.sh"


# collect args
arg1=$1
arg2=$2

# output args and arg count to stdout
info "number of args: $#"
info "arg1 ${arg1}"
info "arg2 ${arg2}"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

Esempio di gancio pre-snapshot/post-snapshot

Nell'esempio seguente viene illustrato come utilizzare lo stesso script sia per un hook pre-snapshot che per un hook post-snapshot.

```
#!/bin/sh

# success_sample_pre_post.sh
#
# A simple success hook script example with an arg for testing purposes
# to demonstrate how the same script can be used for both a prehook and
posthook
#
# args: [pre|post]

# unique error codes for every error case
ebase=100
eusage=$((ebase+1))
ebadstage=$((ebase+2))
epre=$((ebase+3))
epost=$((ebase+4))


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
```

```
error() {
    msg "ERROR: $*" 1>&2
}


#
# Would run prehook steps here
#
prehook() {
    info "Running noop prehook"
    return 0
}

#
# Would run posthook steps here
#
posthook() {
    info "Running noop posthook"
    return 0
}


#
# main
#

# check arg
stage=$1
if [ -z "${stage}" ]; then
    echo "Usage: $0 <pre|post>"
    exit ${eusage}
fi

if [ "${stage}" != "pre" ] && [ "${stage}" != "post" ]; then
    echo "Invalid arg: ${stage}"
    exit ${ebadstage}
fi

# log something to stdout
info "running success_sample_pre_post.sh"

if [ "${stage}" = "pre" ]; then
    prehook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during prehook"
```

```
    fi
fi

if [ "${stage}" = "post" ]; then
    posthook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during posthook"
    fi
fi

exit ${rc}
```

Esempio di guasto

Nell'esempio riportato di seguito viene illustrato come gestire gli errori in un hook.

```
#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}
```

```
#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}
```

Esempio di errore dettagliato

Nell'esempio riportato di seguito viene illustrato come gestire gli errori in modo semplice, con una registrazione più dettagliata.

```
#!/bin/sh

# failure_sample_verbose.sh
#
# A simple failure hook script with args for testing purposes.
#
# args: [The number of lines to output to stdout]


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
```

```
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running failure_sample_verbose.sh"


# output arg value to stdout
linecount=$1
info "line count ${linecount}"

# write out a line to stdout based on line count arg
i=1
while [ "$i" -le ${linecount} ]; do
    info "This is line ${i} from failure_sample_verbose.sh"
    i=$(( i + 1 ))
done

error "exiting with error code 8"
exit 8
```

Errore con un esempio di codice di uscita

Nell'esempio riportato di seguito viene illustrato un errore di hook con un codice di uscita.

```
#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#
```

```
# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}
```

Esempio di successo dopo il guasto

Nell'esempio riportato di seguito viene illustrato un errore di hook alla prima esecuzione, ma dopo la seconda esecuzione.

```
#!/bin/sh

# failure_then_success_sample.sh
#
# A hook script that fails on initial run but succeeds on second run for
testing purposes.
#
# Helpful for testing retry logic for post hooks.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}
```

```
#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running failure_success sample.sh"


if [ -e /tmp/hook-test.junk ] ; then
    info "File does exist.  Removing /tmp/hook-test.junk"
    rm /tmp/hook-test.junk
    info "Second run so returning exit code 0"
    exit 0
else
    info "File does not exist.  Creating /tmp/hook-test.junk"
    echo "test" > /tmp/hook-test.junk
    error "Failed first run, returning exit code 5"
    exit 5
fi
```