Configura la multi-tenancy su Red Hat OpenShift con NetApp ONTAP

NetApp Solutions

NetApp
April 26, 2024

Sommario

Configura la multi-tenancy su Red Hat OpenShift con NetApp ONTAP

Configurazione della multi-tenancy su Red Hat OpenShift con NetApp

Molte organizzazioni che eseguono più applicazioni o carichi di lavoro su container tendono a implementare un cluster Red Hat OpenShift per applicazione o carico di lavoro. Ciò consente loro di implementare un rigoroso isolamento per l'applicazione o il carico di lavoro, ottimizzare le performance e ridurre le vulnerabilità della sicurezza. Tuttavia, l'implementazione di un cluster Red Hat OpenShift separato per ciascuna applicazione pone un proprio insieme di problemi. Aumenta l'overhead operativo dovendo monitorare e gestire ciascun cluster da solo, aumenta i costi grazie alle risorse dedicate per le diverse applicazioni e ostacola l'efficienza della scalabilità.

Per risolvere questi problemi, si può prendere in considerazione l'esecuzione di tutte le applicazioni o i carichi di lavoro in un singolo cluster Red Hat OpenShift. Tuttavia, in un'architettura di questo tipo, le vulnerabilità legate all'isolamento delle risorse e alla sicurezza delle applicazioni sono una delle sfide principali. Qualsiasi vulnerabilità di sicurezza in un workload potrebbe naturalmente ricadersi in un altro workload, aumentando così la zona di impatto. Inoltre, qualsiasi utilizzo improvviso e non controllato delle risorse da parte di un'applicazione può influire sulle prestazioni di un'altra applicazione, poiché non esiste un criterio di allocazione delle risorse per impostazione predefinita.

Pertanto, le organizzazioni cercano soluzioni in grado di ottenere il meglio in entrambi i mondi, ad esempio, consentendo loro di eseguire tutti i propri carichi di lavoro in un singolo cluster e offrendo al contempo i vantaggi di un cluster dedicato per ogni carico di lavoro.

Una di queste soluzioni efficaci consiste nel configurare la multi-tenancy su Red Hat OpenShift. La multi-tenancy è un'architettura che consente a più tenant di coesistere sullo stesso cluster con un corretto isolamento delle risorse, della sicurezza e così via. In questo contesto, un tenant può essere visualizzato come un sottoinsieme delle risorse del cluster configurate per essere utilizzate da un particolare gruppo di utenti a scopo esclusivo. La configurazione della multi-tenancy su un cluster Red Hat OpenShift offre i seguenti vantaggi:

- Riduzione di CapEx e OpEx grazie alla condivisione delle risorse del cluster
- Riduzione dell'overhead operativo e di gestione
- Proteggere i carichi di lavoro dalla contaminazione incrociata delle violazioni della sicurezza
- Protezione dei carichi di lavoro da un peggioramento inatteso delle performance dovuto a conflitti di risorse

Per un cluster OpenShift multitenant completamente realizzato, è necessario configurare le quote e le restrizioni per le risorse cluster appartenenti a diversi bucket di risorse: Calcolo, storage, networking, sicurezza e così via. Anche se vengono trattati alcuni aspetti di tutti i bucket di risorse di questa soluzione, Ci concentriamo sulle Best practice per isolare e proteggere i dati serviti o consumati da più carichi di lavoro sullo stesso cluster Red Hat OpenShift configurando la multi-tenancy sulle risorse storage allocate dinamicamente da Astra Trident con il supporto di NetApp ONTAP.

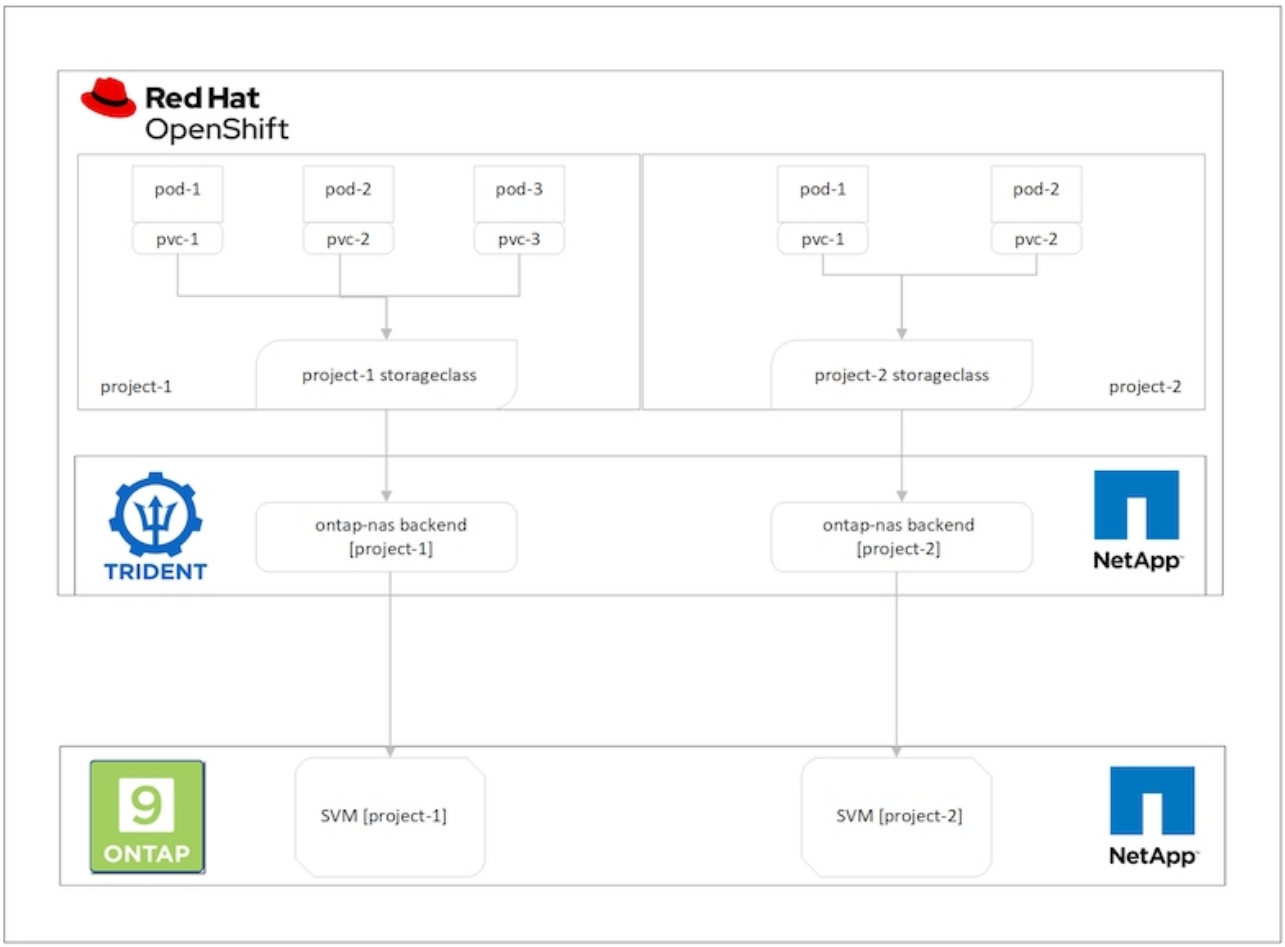
Architettura

Sebbene Red Hat OpenShift e Astra Trident supportati da NetApp ONTAP non forniscano l'isolamento tra i carichi di lavoro per impostazione predefinita, offrono un'ampia gamma di funzionalità che possono essere utilizzate per configurare la multi-tenancy. Per comprendere meglio la progettazione di una soluzione multi-

tenant su un cluster Red Hat OpenShift con Astra Trident supportato da NetApp ONTAP, prendiamo in considerazione un esempio con una serie di requisiti e descriviamo la configurazione che lo circonda.

Supponiamo che un'organizzazione esegue due dei propri workload su un cluster Red Hat OpenShift nell'ambito di due progetti su cui lavorano due team diversi. I dati per questi carichi di lavoro risiedono su PVC che vengono forniti dinamicamente da Astra Trident su un backend NAS NetApp ONTAP. L'organizzazione deve progettare una soluzione multi-tenant per questi due carichi di lavoro e isolare le risorse utilizzate per questi progetti per garantire il mantenimento della sicurezza e delle performance, concentrandosi principalmente sui dati che servono tali applicazioni.

La seguente figura illustra la soluzione multi-tenant su un cluster Red Hat OpenShift con Astra Trident supportato da NetApp ONTAP.



Requisiti tecnologici

1. Cluster di storage NetApp ONTAP
2. Cluster Red Hat OpenShift
3. Astra Trident

Red Hat OpenShift – risorse cluster

Dal punto di vista del cluster Red Hat OpenShift, la risorsa di primo livello da iniziare è il progetto. Un progetto OpenShift può essere visto come una risorsa di cluster che divide l'intero cluster OpenShift in più cluster

virtuali. Pertanto, l'isolamento a livello di progetto fornisce una base per la configurazione della multi-tenancy.

Successivamente, configurare RBAC nel cluster. La Best practice consiste nell'avere tutti gli sviluppatori che lavorano su un singolo progetto o workload configurati in un singolo gruppo di utenti nel provider di identità (IdP). Red Hat OpenShift consente l'integrazione di IdP e la sincronizzazione dei gruppi di utenti, consentendo così l'importazione di utenti e gruppi da IdP nel cluster. Ciò consente agli amministratori del cluster di separare l'accesso delle risorse del cluster dedicate a un progetto a un gruppo di utenti o a gruppi che lavorano su tale progetto, limitando in tal modo l'accesso non autorizzato a qualsiasi risorsa del cluster. Per ulteriori informazioni sull'integrazione di IdP con Red Hat OpenShift, consulta la documentazione "qui".

NetApp ONTAP

È importante isolare lo storage condiviso che funge da provider di storage persistente per un cluster Red Hat OpenShift per assicurarsi che i volumi creati sullo storage per ogni progetto appaiano agli host come se fossero creati su storage separato. A tale scopo, è possibile creare un numero di SVM (macchine virtuali di storage) su NetApp ONTAP pari al numero di progetti o carichi di lavoro e dedicare ogni SVM a un carico di lavoro.

Astra Trident

Dopo aver creato diverse SVM per diversi progetti su NetApp ONTAP, è necessario mappare ciascuna SVM su un backend Trident diverso. La configurazione di back-end su Trident determina l'allocazione dello storage persistente alle risorse del cluster OpenShift e richiede il mapping dei dettagli della SVM. Questo dovrebbe essere il driver del protocollo per il backend al minimo. Facoltativamente, consente di definire il provisioning dei volumi sullo storage e di impostare limiti per la dimensione dei volumi o l'utilizzo degli aggregati e così via. È possibile trovare i dettagli relativi alla definizione dei backend Trident "qui".

Red Hat OpenShift – risorse di storage

Dopo aver configurato i backend Trident, il passaggio successivo consiste nella configurazione di StorageClasses. Configura quante sono le classi di storage in cui sono presenti i backend, fornendo a ciascuna classe di storage l'accesso per eseguire lo spin up dei volumi su un solo backend. È possibile mappare StorageClass a un particolare backend Trident utilizzando il parametro storagePools durante la definizione della classe di storage. È possibile trovare i dettagli per definire una classe di storage "qui". Pertanto, esiste una mappatura uno a uno da StorageClass a Trident backend che punta a una SVM. In questo modo, tutte le attestazioni di storage tramite la StorageClass assegnata a quel progetto vengono gestite solo dalla SVM dedicata a quel progetto.

Poiché le classi di storage non sono risorse con spazio dei nomi, come possiamo garantire che le attestazioni di storage alla classe di storage di un progetto per pod in un altro namespace o progetto vengano rifiutate? La risposta è utilizzare ResourceQuotas. ResourceQuotas sono oggetti che controllano l'utilizzo totale delle risorse per progetto. Può limitare il numero e la quantità totale di risorse che possono essere utilizzate dagli oggetti nel progetto. Quasi tutte le risorse di un progetto possono essere limitate utilizzando ResourceQuotas e questo può aiutare le organizzazioni a ridurre i costi e le interruzioni dovute all'overprovisioning o all'eccessivo consumo di risorse. Consultare la documentazione "qui" per ulteriori informazioni.

In questo caso di utilizzo, dobbiamo limitare i pod di un progetto specifico al fine di richiedere storage da classi di storage non dedicate al loro progetto. A tale scopo, è necessario limitare le richieste di rimborso persistenti per volumi per altre classi di storage mediante l'impostazione `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` a 0. Inoltre, un amministratore del cluster deve garantire che gli sviluppatori di un progetto non abbiano accesso per modificare le ResourceQuotas.

Configurazione

Per qualsiasi soluzione multi-tenant, nessun utente può avere accesso a più risorse di cluster di quelle richieste. Pertanto, l'intero insieme di risorse da configurare come parte della configurazione multi-tenancy è diviso tra cluster-admin, storage-admin e sviluppatori che lavorano su ciascun progetto.

La seguente tabella descrive le diverse attività che devono essere eseguite da diversi utenti:

Ruolo	Attività
Cluster-admin	Crea progetti per applicazioni o carichi di lavoro diversi
	Creare ClusterRoles e RoleBinding per l'amministrazione dello storage
	Creazione di ruoli e associazioni per gli sviluppatori che assegnano l'accesso a progetti specifici
	[Facoltativo] configurare i progetti per pianificare i pod su nodi specifici
Storage-admin	Creare SVM su NetApp ONTAP
	Creare backend Trident
	Creare StorageClasses
	Creare ResourceQuotas di storage
Sviluppatori	Convalidare l'accesso per creare o applicare patch a PVC o pod nel progetto assegnato
	Convalida l'accesso per creare o applicare patch a PVC o pod in un altro progetto
	Convalida l'accesso per visualizzare o modificare progetti, ResourceQuotas e StorageClasses

Configurazione

Prerequisiti

- Cluster NetApp ONTAP
- Cluster Red Hat OpenShift
- Trident installato sul cluster
- Workstation di amministrazione con tool tridentctl e oc installati e aggiunti al percorso dei dollari
- Accesso amministratore a ONTAP
- Accesso cluster-admin al cluster OpenShift
- Il cluster è integrato con il provider di identità
- Il provider di identità è configurato in modo da distinguere in modo efficiente tra gli utenti di diversi team

Configurazione: Attività di amministrazione del cluster

Le seguenti attività vengono eseguite dall'amministratore del cluster Red Hat OpenShift:

1. Accedere al cluster Red Hat OpenShift come amministratore del cluster.
2. Creare due progetti corrispondenti a progetti diversi.

```
oc create namespace project-1
oc create namespace project-2
```

3. Creare il ruolo di sviluppatore per il progetto-1.

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-1
  name: developer-project-1
rules:
  - verbs:
      - '*'
    apiGroups:
      - apps
      - batch
      - autoscaling
      - extensions
      - networking.k8s.io
      - policy
      - apps.openshift.io
      - build.openshift.io
      - image.openshift.io
      - ingress.operator.openshift.io
      - route.openshift.io
      - snapshot.storage.k8s.io
      - template.openshift.io
    resources:
      - '*'
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - bindings
      - configmaps
      - endpoints
```

```
      - events
      - persistentvolumeclaims
      - pods
      - pods/log
      - pods/attach
      - podtemplates
      - replicationcontrollers
      - services
      - limitranges
      - namespaces
      - componentstatuses
      - nodes
  - verbs:
      - '*'
    apiGroups:
      - trident.netapp.io
    resources:
      - tridentsnapshots
EOF
```

La definizione del ruolo fornita in questa sezione è solo un esempio. I ruoli dello sviluppatore devono essere definiti in base ai requisiti dell'utente finale.

1. Allo stesso modo, creare ruoli di sviluppatore per il progetto 2.
2. Tutte le risorse storage di OpenShift e NetApp sono generalmente gestite da un amministratore dello storage. L'accesso per gli amministratori dello storage è controllato dal ruolo di operatore trident creato al momento dell'installazione di Trident. Inoltre, l'amministratore dello storage richiede l'accesso a ResourceQuotas per controllare il modo in cui lo storage viene utilizzato.
3. Creare un ruolo per la gestione di ResourceQuotas in tutti i progetti del cluster per associarlo all'amministratore dello storage.

```
cat << EOF | oc create -f -
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: resource-quotas-role
rules:
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - resourcequotas
  - verbs:
      - '*'
    apiGroups:
      - quota.openshift.io
    resources:
      - '*'
EOF
```

4. Assicurarsi che il cluster sia integrato con il provider di identità dell'organizzazione e che i gruppi di utenti siano sincronizzati con i gruppi di cluster. L'esempio seguente mostra che il provider di identità è stato integrato con il cluster e sincronizzato con i gruppi di utenti.

```
$ oc get groups
NAME                        USERS
ocp-netapp-storage-admins   ocp-netapp-storage-admin
ocp-project-1               ocp-project-1-user
ocp-project-2               ocp-project-2-user
```

1. Configurare ClusterRoleBinding per gli amministratori dello storage.

```
cat << EOF | oc create -f -
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-trident-operator
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-operator
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-resource-quotas-cr
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: resource-quotas-role
EOF
```

Per gli amministratori dello storage, devono essere associati due ruoli: trident-operator e Resource-quote.

1. Creare i RoleBinding per gli sviluppatori che associano il ruolo Developer-project-1 al gruppo corrispondente (ocp-project-1) nel progetto-1.

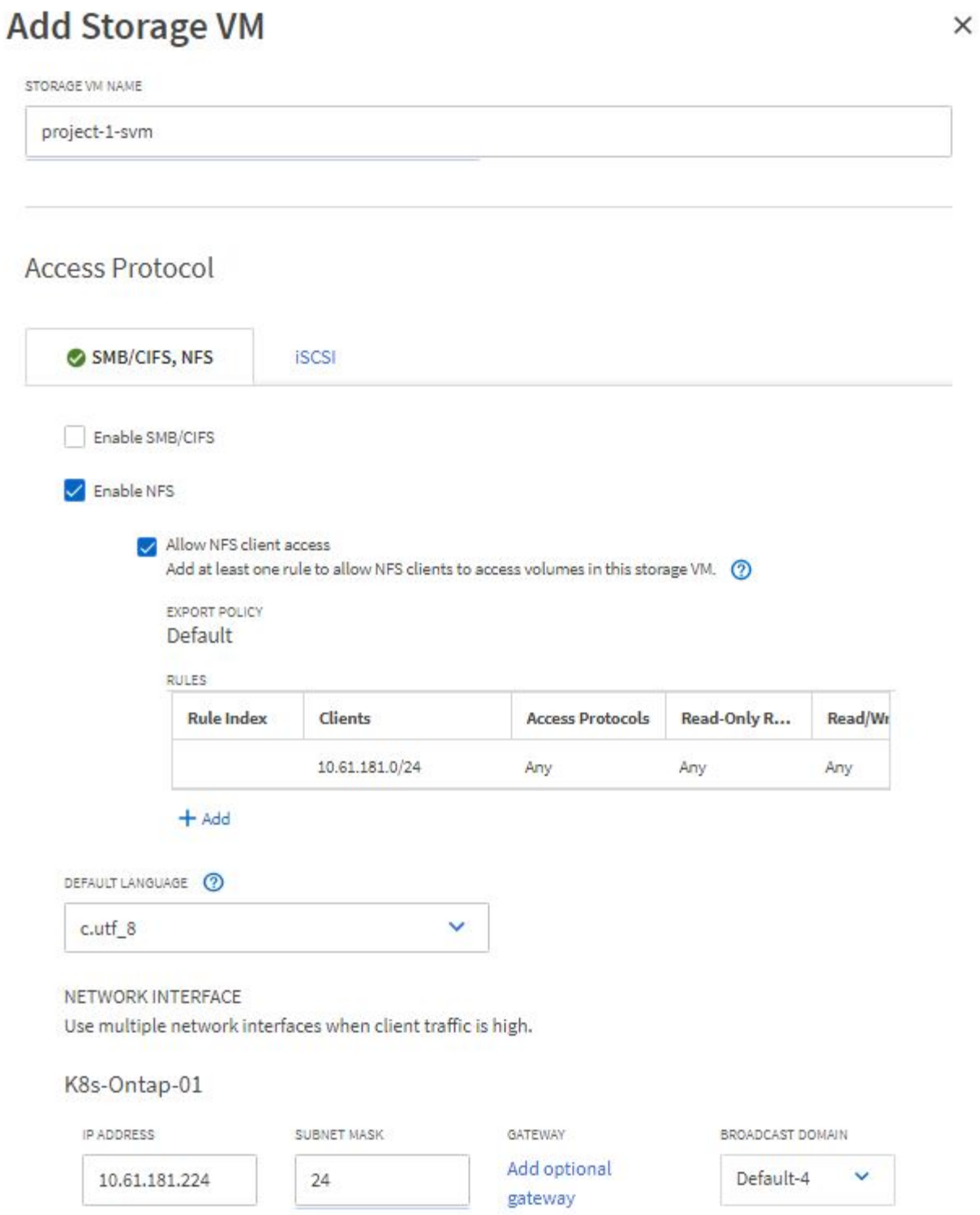
```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-1
EOF
```

2. Allo stesso modo, creare RoleBinding per gli sviluppatori che associano i ruoli di sviluppatore al gruppo di utenti corrispondente nel progetto-2.

Configurazione: Attività di amministrazione dello storage

Le seguenti risorse devono essere configurate da un amministratore dello storage:

1. Accedere al cluster NetApp ONTAP come amministratore.
2. Accedere a Storage > Storage VM (Storage > Storage VM) e fare clic su Add (Aggiungi). Creare due SVM, una per il progetto 1 e l'altra per il progetto 2, fornendo i dettagli richiesti. Inoltre, creare un account vsadmin per gestire SVM e le relative risorse.



1. Accedere al cluster Red Hat OpenShift come amministratore dello storage.
2. Creare il backend per il progetto 1 e mapparlo sulla SVM dedicata al progetto. NetApp consiglia di utilizzare l'account vsadmin di SVM per connettere il backend a SVM invece di utilizzare l'amministratore del cluster ONTAP.

```
cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_1",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.224",
    "svm": "project-1-svm",
    "username": "vsadmin",
    "password": "NetApp123"
}
EOF
```

Per questo esempio, viene utilizzato il driver ontap-nas. Utilizzare il driver appropriato per creare il backend in base al caso d'utilizzo.

Supponiamo che Trident sia installato nel progetto Trident.

1. Analogamente, creare il backend Trident per il progetto 2 e mapparlo sulla SVM dedicata al progetto 2.
2. Quindi, creare le classi di storage. Creare la classe di storage per il project-1 e configurarla per utilizzare i pool di storage dal backend dedicato al project-1 impostando il parametro storagePools.

```
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-1-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_1:.*"
EOF
```

3. Allo stesso modo, creare una classe di storage per il progetto 2 e configurarla per utilizzare i pool di storage dal back-end dedicato al progetto 2.
4. Creare un ResourceQuota per limitare le risorse nel progetto 1, richiedendo storage da storageclasses dedicati ad altri progetti.

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-1-sc-rq
  namespace: project-1
spec:
  hard:
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

5. Allo stesso modo, creare un ResourceQuota per limitare le risorse nel progetto 2, richiedendo lo storage da storageclasses dedicati ad altri progetti.

Convalida

Per convalidare l'architettura multi-tenant configurata nei passaggi precedenti, attenersi alla seguente procedura:

Convalidare l'accesso per creare PVC o pod nel progetto assegnato

1. Accedi come ocp-project-1-user, Developer in project-1.
2. Controllare l'accesso per creare un nuovo progetto.

```
oc create ns sub-project-1
```

3. Creare un PVC nel progetto 1 utilizzando lo storageclass assegnato al progetto 1.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

4. Controllare il PV associato al PVC.

```
oc get pv
```

5. Convalida che il PV e il suo volume siano creati in una SVM dedicata al progetto 1 su NetApp ONTAP.

```
volume show -vserver project-1-svm
```

6. Creare un pod nel progetto 1 e montare il PVC creato nel passaggio precedente.

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  volumes:
    - name: test-pvc-project-1
      persistentVolumeClaim:
       claimName: test-pvc-project-1
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: test-pvc-project-1
EOF
```

7. Verificare che il pod sia in funzione e che il volume sia stato montato.

```
oc describe pods test-pvc-pod -n project-1
```

Convalidare l'accesso per creare PVC o pod in un altro progetto o utilizzare risorse dedicate a un altro progetto

1. Accedi come ocp-project-1-user, Developer in project-1.
2. Creare un PVC nel progetto 1 utilizzando lo storageclass assegnato al progetto 2.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1-sc-2
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-2-sc
EOF
```

3. Creare un PVC nel progetto 2.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-2-sc-1
  namespace: project-2
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

4. Assicurarsi che i PVC `test-pvc-project-1-sc-2` e. `test-pvc-project-2-sc-1` non sono stati creati.

```
oc get pvc -n project-1
oc get pvc -n project-2
```

5. Creare un pod nel progetto 2.

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
EOF
```

Convalida l’accesso per visualizzare e modificare progetti, ResourceQuotas e StorageClasses

1. Accedi come ocp-project-1-user, Developer in project-1.
2. Controllare l’accesso per creare nuovi progetti.

```
oc create ns sub-project-1
```

3. Convalidare l’accesso per visualizzare i progetti.

```
oc get ns
```

4. Verificare se l’utente può visualizzare o modificare ResourceQuotas nel progetto-1.

```
oc get resourcequotas -n project-1
oc edit resourcequotas project-1-sc-rq -n project-1
```

5. Verificare che l’utente abbia accesso per visualizzare gli storageclasses.

```
oc get sc
```

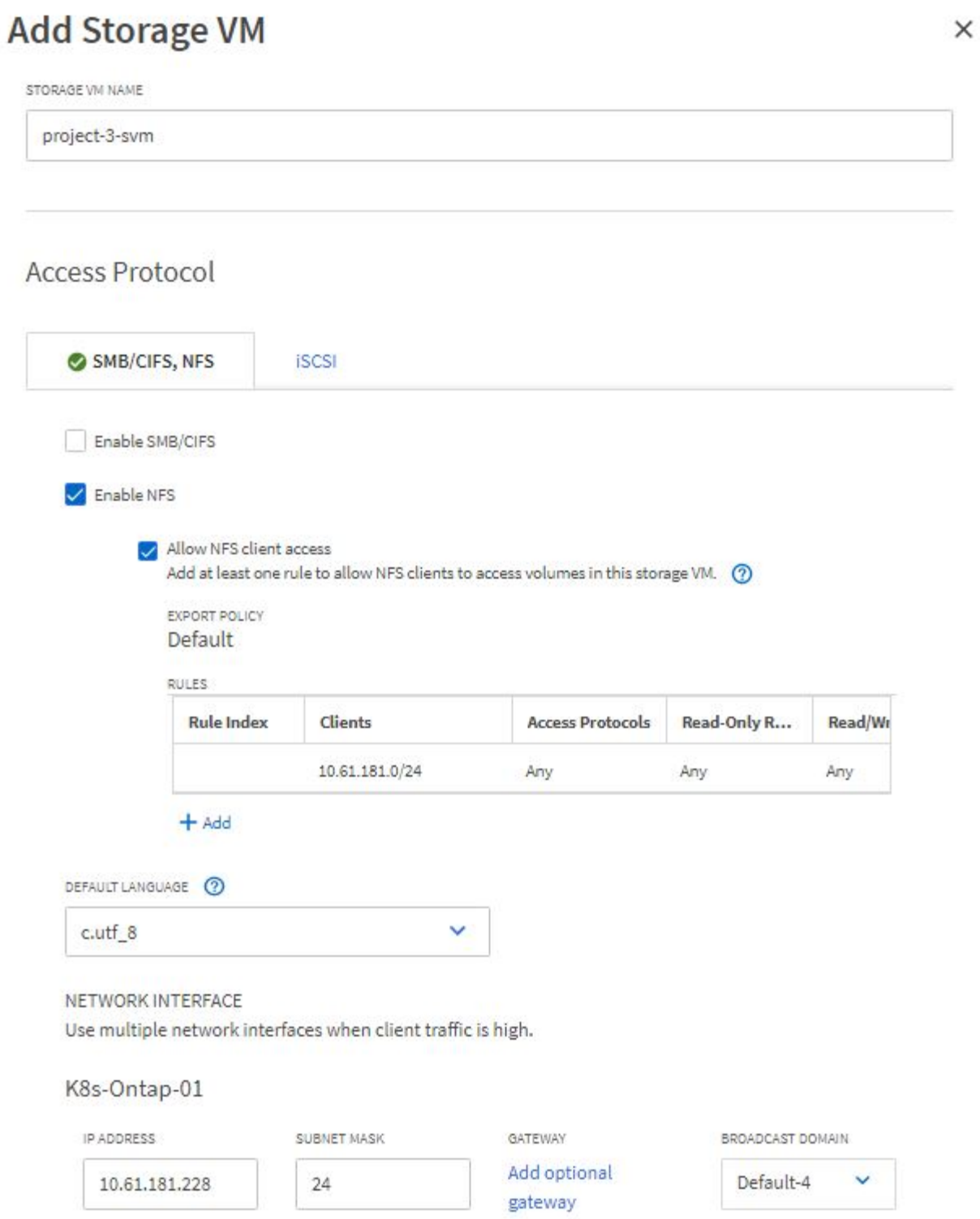
6. Controllare l’accesso per descrivere i magazzini.
7. Convalidare l’accesso dell’utente per modificare gli storageclasses.

```
oc edit sc project-1-sc
```

Scalabilità: Aggiunta di più progetti

In una configurazione multi-tenant, l'aggiunta di nuovi progetti con risorse di storage richiede una configurazione aggiuntiva per garantire che la multi-tenancy non venga violata. Per aggiungere altri progetti in un cluster multi-tenant, attenersi alla seguente procedura:

1. Accedere al cluster NetApp ONTAP come amministratore dello storage.
2. Selezionare `Storage → Storage VMs` e fare clic su `Add`. Creare una nuova SVM dedicata al progetto
3. Inoltre, creare un account vsadmin per gestire SVM e le relative risorse.



1. Accedere al cluster Red Hat OpenShift come amministratore del cluster.
2. Creare un nuovo progetto.

```
oc create ns project-3
```

3. Assicurarsi che il gruppo di utenti per il project-3 sia creato su IdP e sincronizzato con il cluster OpenShift.

```
oc get groups
```

4. Creare il ruolo di sviluppatore per il progetto 3.

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-3
  name: developer-project-3
rules:
  - verbs:
      - '*'
    apiGroups:
      - apps
      - batch
      - autoscaling
      - extensions
      - networking.k8s.io
      - policy
      - apps.openshift.io
      - build.openshift.io
      - image.openshift.io
      - ingress.operator.openshift.io
      - route.openshift.io
      - snapshot.storage.k8s.io
      - template.openshift.io
    resources:
      - '*'
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - bindings
      - configmaps
      - endpoints
      - events
      - persistentvolumeclaims
      - pods
      - pods/log
      - pods/attach
      - podtemplates
      - replicationcontrollers
      - services
```

```
      - limitranges
      - namespaces
      - componentstatuses
      - nodes
  - verbs:
      - '*'
    apiGroups:
      - trident.netapp.io
    resources:
      - tridentsnapshots
EOF
```

La definizione del ruolo fornita in questa sezione è solo un esempio. Il ruolo dello sviluppatore deve essere definito in base ai requisiti dell'utente finale.

1. Creare il RoleBinding per gli sviluppatori nel progetto-3 che legano il ruolo di sviluppatore-progetto-3 al gruppo corrispondente (ocp-progetto-3) nel progetto-3.

```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-3-developer
  namespace: project-3
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-3
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-3
EOF
```

2. Accedere al cluster Red Hat OpenShift come amministratore dello storage
3. Creare un backend Trident e mapparlo sulla SVM dedicata al progetto 3. NetApp consiglia di utilizzare l'account vsadmin della SVM per connettere il backend alla SVM invece di utilizzare l'amministratore del cluster ONTAP.

```
cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_3",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.228",
    "svm": "project-3-svm",
    "username": "vsadmin",
    "password": "NetApp!23"
}
EOF
```

Per questo esempio, viene utilizzato il driver ontap-nas. Utilizzare il driver appropriato per creare il backend in base al caso d'utilizzo.

Supponiamo che Trident sia installato nel progetto Trident.

1. Creare la classe di storage per il progetto 3 e configurarla per utilizzare i pool di storage dal back-end dedicato al progetto 3.

```
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-3-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_3:.*"
EOF
```

2. Creare un ResourceQuota per limitare le risorse nel progetto 3, richiedendo storage da storageclasses dedicati ad altri progetti.

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-3-sc-rq
  namespace: project-3
spec:
  hard:
    project-1-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

3. Applicare patch alle ResourceQuotas in altri progetti per limitare l'accesso alle risorse in tali progetti dallo storage dallo storageclass dedicato al progetto-3.

```
oc patch resourcequotas project-1-sc-rq -n project-1 --patch
'{"spec":{"hard":{ "project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
oc patch resourcequotas project-2-sc-rq -n project-2 --patch
'{"spec":{"hard":{ "project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
```