



設定

NetApp Solutions

NetApp
April 10, 2024

This PDF was generated from https://docs.netapp.com/ja-jp/netapp-solutions/containers/rh-os-n_use_case_multitenancy_configuration_prerequisites.html on April 10, 2024. Always check docs.netapp.com for the latest.

目次

設定	1
設定	1
Configuration : クラスタ管理者のタスク	2
設定 : ストレージ管理者のタスク	6
検証	9
拡張 : プロジェクトの追加	14

設定

マルチテナント解決策では、必要以上に多くのクラスタリソースにアクセスすることはできません。つまり、マルチテナンシー構成の一部として構成するリソースセット全体が、クラスタ管理者、ストレージ管理者、および各プロジェクトに取り組む開発者に分けられます。

次の表に、各ユーザが実行する各タスクを示します。

ロール	タスク
* Cluster-admin*	さまざまなアプリケーションやワークロード用のプロジェクトを作成できます
	Storage Admin 用の ClusterRoles および RoleBindings を作成します
	ロールとロールの作成特定のアクセス権を割り当てる開発者のためのバインド プロジェクト
	[オプション] 特定のノードでポッドをスケジュールするようにプロジェクトを設定します
* ストレージ管理者 *	NetApp ONTAP に SVM を作成する
	Trident バックエンドを作成
	ストレージクラスを作成します
	ストレージリソースクォータを作成します
* 開発者 *	割り当てられたプロジェクトで PVC またはポッドを作成またはパッチするためのアクセスを検証します
	アクセスを検証して、別のプロジェクトで PVC またはポッドを作成またはパッチします
	アクセス権を検証して、プロジェクト、リソースクォータ、ストレージクラスを表示または編集します

設定

前提条件

- NetApp ONTAP クラスタ：
- Red Hat OpenShift クラスタ
- Trident がクラスタにインストールされている。
- tridentctl および OC ツールがインストールされ、\$PATH に追加された管理ワークステーション。
- ONTAP への管理アクセス。
- OpenShift クラスタへのクラスタ管理者アクセス。
- クラスタがアイデンティティプロバイダに統合されました。
- アイデンティティプロバイダは、異なるチームのユーザを効率的に区別するように設定されています。

Configuration : クラスタ管理者のタスク

Red Hat OpenShift cluster-admin によって次のタスクが実行されます。

1. Red Hat OpenShift クラスタに cluster-admin としてログインします。
2. 異なるプロジェクトに対応する 2 つのプロジェクトを作成します。

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

3. project-1 の開発者ロールを作成します。

```
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:
  - trident snapshots
EOF
```



ここで説明するロール定義は単なる例です。エンドユーザの要件に基づいて開発者の役割を定義する必要があります。

1. 同様に、project-2 の開発者ロールを作成します。
2. すべての OpenShift およびネットアップストレージリソースは、通常はストレージ管理者が管理します。ストレージ管理者向けのアクセスは、Trident のインストール時に作成された Trident オペレータロールによって制御されます。これに加えて、ストレージ管理者は ResourceQuotas にアクセスして、ストレージの消費方法を制御する必要があります。
3. クラスタ内のすべてのプロジェクトの ResourceQuotas を管理する役割を作成して、ストレージ管理者に割り当てます。

```
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. クラスタが組織のアイデンティティプロバイダと統合され、ユーザグループがクラスタグループと同期されていることを確認します。次の例は、アイデンティティプロバイダがクラスタに統合され、ユーザグループと同期されていることを示しています。

```
$ 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. ストレージ管理者用の ClusterRoleBindings を設定します。

```

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

```



ストレージ管理者の場合は、Trident オペレータとリソースクォータの 2 つのロールにバインドする必要があります。

1. ロールの作成 - developer-project-1 のロールを project-1 の対応するグループ (OCP-project-1) にバインドする開発者のバインディング。

```
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. 同様に、開発者の役割を project-2 の対応するユーザーグループにバインドする開発者の RoleBindings を作成します。

設定：ストレージ管理者のタスク

ストレージ管理者が次のリソースを設定する必要があります。

1. NetApp ONTAP クラスタに admin としてログインします。
2. Storage > Storage VMs と進み、Add をクリックします。必要な詳細を指定して、プロジェクト 1 用とプロジェクト 2 用に 1 つずつ、2 つの SVM を作成します。また、SVM とそのリソースを管理するには vsadmin アカウントを作成します。

Add Storage VM



STORAGE VM NAME

project-1-svm

Access Protocol

☒ SMB/CIFS, NFS

iSCSI

☐ Enable SMB/CIFS

☒ Enable NFS

☒ Allow NFS client access

Add at least one rule to allow NFS clients to access volumes in this storage VM. [?](#)

EXPORT POLICY

Default

RULES

Rule Index	Clients	Access Protocols	Read-Only R...	Read/Wr
	10.61.181.0/24	Any	Any	Any

[+ Add](#)

DEFAULT LANGUAGE [?](#)

c.utf_8

NETWORK INTERFACE

Use multiple network interfaces when client traffic is high.

K8s-Ontap-01

IP ADDRESS

10.61.181.224

SUBNET MASK

24

GATEWAY

[Add optional gateway](#)

BROADCAST DOMAIN

Default-4

1. ストレージ管理者として Red Hat OpenShift クラスタにログインします。
2. project-1 のバックエンドを作成し、プロジェクト専用の SVM にマッピングします。ONTAP クラスタ管理者を使用する代わりに、SVM の vsadmin アカウントを使用してバックエンドを SVM に接続することを推奨します。

```
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
```



この例では ONTAP と NAS のドライバを使用しています。ユースケースに基づいてバックエンドを作成する場合は、適切なドライバを使用します。



Trident が Trident プロジェクトにインストールされているとします。

1. 同様に、project-2 の Trident バックエンドを作成し、project-2 に専用の SVM にマッピングします。
2. 次に、ストレージクラスを作成します。StoragePools パラメータを設定して、project-1 専用のバックエンドのストレージプールを使用するように project-1 のストレージクラスを作成し、これを設定します。

```
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. 同様に、project-2 に対してストレージクラスを作成し、project-2 に専用のバックエンドのストレージプールを使用するように設定します。
4. ResourceQuota を作成して 'プロジェクト 1 内のリソースを制限し' 他のプロジェクト専用のストレージを要求します

```
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. 同様に 'ResourceQuota' を作成して 'project-2' 内のリソースを制限し '他のプロジェクト専用のストレージ' を要求します

検証

前の手順で設定したマルチテナントアーキテクチャを検証するには、次の手順を実行します。

割り当てられたプロジェクトで **PVC** またはポッドを作成するためのアクセスを検証します

1. OCP-project-1-user として、project-1 の開発者としてログインします。
2. アクセス権をチェックして新しいプロジェクトを作成してください

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

3. project-1 に割り当てられたストレージクラスを使用して 'project-1' に PVC を作成します

```
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. PVCに関連付けられているPVを確認します

```
oc get pv
```

5. PVとそのボリュームが、NetApp ONTAP 上のプロジェクト 1 専用の SVM に作成されていることを確認します。

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

6. project-1 にポッドを作成し、前の手順で作成した PVC をマウントします。

```
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. ポッドが実行中かどうか、およびボリュームがマウントされているかどうかを確認します。

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

アクセスを検証して別のプロジェクトに **PVC** またはポッドを作成するか、別のプロジェクト専用のリソースを使用します

1. OCP-project-1-user として、project-1 の開発者としてログインします。
2. project-2 に割り当てられたストレージクラスを使用して 'project-1 に PVC を作成します

```
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. PROJECT-2 で PVC を作成します。

```
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. PVC 「test-pvc-project-1-sc-2」 および 「test-pvc-project-2-ssc-1」 が作成されていないことを確認します。

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

5. プロジェクト 2 でポッドを作成します。

```
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
```

アクセス権を検証して、プロジェクト、リソースクォータ、ストレージクラスを表示および編集します

1. OCP-project-1-user として、project-1 の開発者としてログインします。
2. アクセス権をチェックして新しいプロジェクトを作成してください。

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

3. アクセスを検証してプロジェクトを表示します

```
oc get ns
```

4. ユーザーが ResourceQuotas を表示または編集できるかどうかを確認します プロジェクト 1

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

5. ユーザーがストレージクラスを表示するためのアクセス権を持っていることを確認します

```
oc get sc
```

6. ストレージクラスについては 'アクセスを確認してください
7. ストレージクラスを編集するためにユーザーのアクセス権を検証します

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

拡張：プロジェクトの追加

マルチテナント構成でストレージリソースを使用する新しいプロジェクトを追加する場合、マルチテナンシーを違反しないように追加の設定が必要になります。マルチテナントクラスタでプロジェクトを追加するには、次の手順を実行します。

1. NetApp ONTAP クラスタにストレージ管理者としてログインします。
2. 「ストレージ → ストレージ VM」に移動し、「追加」をクリックします。project-3 専用の新しい SVM を作成します。また、SVM とそのリソースを管理するには vsadmin アカウントを作成します。

Add Storage VM



STORAGE VM NAME

project-3-svm

Access Protocol

☒ SMB/CIFS, NFS

iSCSI

☐ Enable SMB/CIFS

☒ Enable NFS

☒ Allow NFS client access

Add at least one rule to allow NFS clients to access volumes in this storage VM. [?](#)

EXPORT POLICY

Default

RULES

Rule Index	Clients	Access Protocols	Read-Only R...	Read/Wr
	10.61.181.0/24	Any	Any	Any

[+ Add](#)

DEFAULT LANGUAGE [?](#)

c.utf_8

NETWORK INTERFACE

Use multiple network interfaces when client traffic is high.

K8s-Ontap-01

IP ADDRESS

10.61.181.228

SUBNET MASK

24

GATEWAY

[Add optional gateway](#)

BROADCAST DOMAIN

Default-4

1. Red Hat OpenShift クラスタにクラスタ管理者としてログインします
2. 新しいプロジェクトを作成します。

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

3. IdP に project-3 のユーザグループが作成され、 OpenShift クラスタと同期されていることを確認してください。

```
oc get groups
```

4. project-3 の開発者ロールを作成します。

```
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:
- trident snapshots
EOF

```



ここで説明するロール定義は単なる例です。開発者ロールは、エンドユーザの要件に基づいて定義する必要があります。

1. プロジェクト 3 の開発者用に RoleBinding を作成します。これは、developer-project-3 の役割を、project-3 の対応するグループ (OCP-project-3) にバインドします。

```

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. Red Hat OpenShift クラスタにストレージ管理者としてログインします
3. Trident バックエンドを作成し、project-3 専用の SVM にマッピングします。ONTAP クラスタ管理者を使用する代わりに、SVM の vsadmin アカウントを使用してバックエンドを SVM に接続することを推奨します。

```
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
```



この例では ONTAP と NAS のドライバを使用しています。ユースケースに基づいてバックエンドを作成するための適切なドライバを使用します。



Trident が Trident プロジェクトにインストールされているとします。

1. project-3 用のストレージクラスを作成し、project-3 専用のバックエンドのストレージプールを使用するように設定します。

```
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. ResourceQuota を作成して 'プロジェクト 3 のリソースを制限しますストレージを要求するストレージは '他のプロジェクト専用のストレージになります

```
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. 他のプロジェクトの ResourceQuotas にパッチを適用して'プロジェクト内のリソースがプロジェクト 3 専用のストレージからストレージにアクセスするのを制限します

```
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}}}'
```

著作権に関する情報

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。