KubeCon

North America 2017

Multi-Tenancy Support & Security Modeling with RBAC and Namespaces

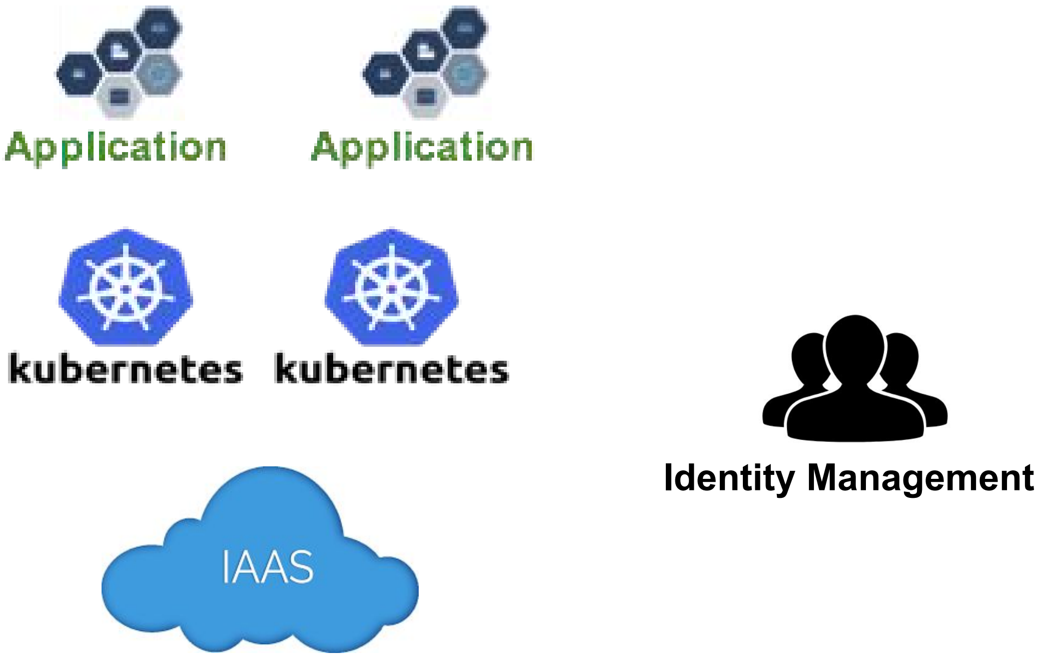
Frederick Vong, Staff Engineer, VMware
Michael Chen, Senior Manager, VMware

Dec 7, 2017

What are covered in this presentation

- A brief description of the project background
- A brief discussion of kubernetes namespace on how it can provide isolation
- What mechanisms are provided by RBAC to enforce policies/permissions
- How to build user security model using kubernetes features:
  - Namespace
  - RBAC
- A few user and security models will be discussed for both multi-tenancy and a single tenancy support on top of kubernetes cluster
  - Cluster level
  - Namespace level
- Demo

Stack Overview



Application

Application

kubernetes

kubernetes

IAAS

Identity Management

Personas

Cloud Administrator

- Cluster Management
- User & Group Management
- Overall Operations & Logs

Application Development Team

DevOps Administrator

- Scale Clusters
- Reporting, Dashboard and Operations Management for the Project / Apps

Developer

- Consumer for K8s API
- Definition of Application Resources
- Application and image deployment
- Application Operation (App Ops)

Kubernetes Namespaces and RBAC

What is Kubernetes

Open-source platform designed to automate deploying, scaling, and operating application containers.

For more readings, please go to this link below
https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

Some Kubernetes Concepts

- Node
  - a work machine in Kubernetes cluster
  - may be a VM or physical machine
- Namespace
  - virtual clusters that provide isolation of resources.
- Pod
  - unit of deployment: a single instance of an application in Kubernetes. One or more containers.
- Service a.k.a Svc
  - abstraction which defines a logical set of Pods and a policy by which to access them
- RBAC
  - Role-Based Access Control
  - Allowing admins to dynamically configure policies to drive authorization decisions

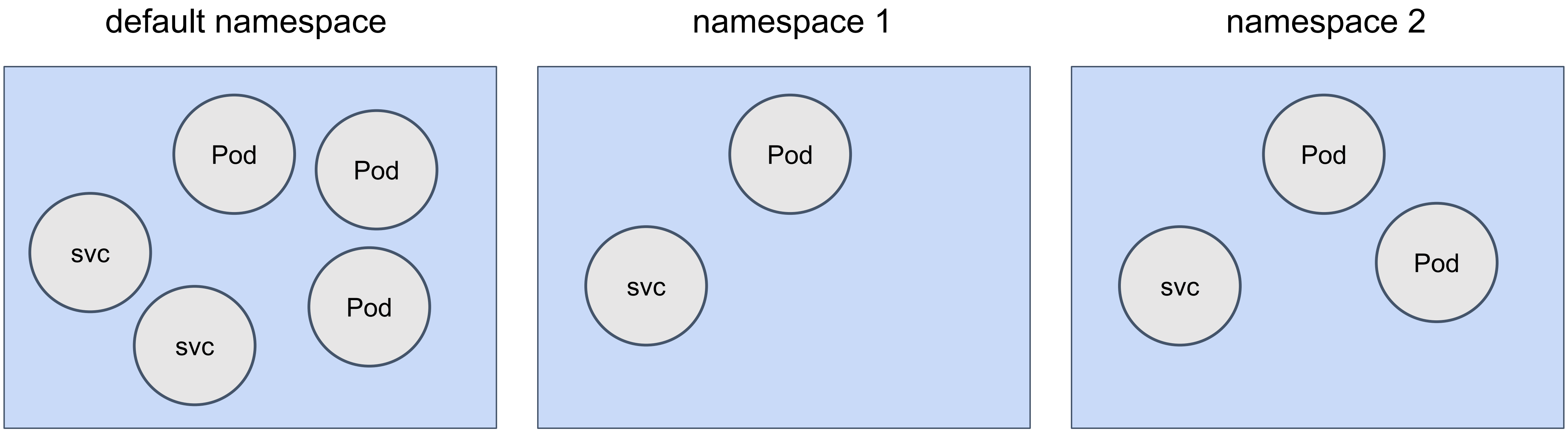
Kubernetes Security models

- Kubernetes does not dictate a particular secure model (cloud platform neutral)
- Two categories of users
  - service accounts managed by kubernetes
  - normal users managed by outside
- Can extend the authentication through plugins
- Can extend the authorization through plugins

Inside the Toolbox

Kubernetes Namespaces

Namespace provides isolation of resources



Kubernetes RBAC Concepts

- Rules - a set of permissions
  - Cluster Role
    - both cluster and namespace levels
  - Role
    - namespace level
- Granting Permission
  - ClusterRolebinding
    - cluster-wide and all namespaces
  - Rolebinding
    - a single namespace only
- Subjects (Part of the definition of ClusterRolebinding and Rolebinding)
  - users, groups and service accounts

Cluster Role

- can be used to grant read access to resources in any particular namespace, or across all namespaces
- Example - grant read access to nodes

```yaml
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  # "namespace" omitted since ClusterRoles are not namespaced
  name: node-reader
rules:
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["get", "watch", "list"]
```

Role

- can only be used to grant access to resources within a single namespace
- Example - grant read access to pods

```yaml
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: pod-reader
rules:
- apiGroups: [""] # "" indicates the core API group
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

Cluster Role Binding

- Grant the permissions defined in a role to a user or set of users. It holds a list of subjects (users, groups, or service accounts). It applies to cluster-wide.
- Example

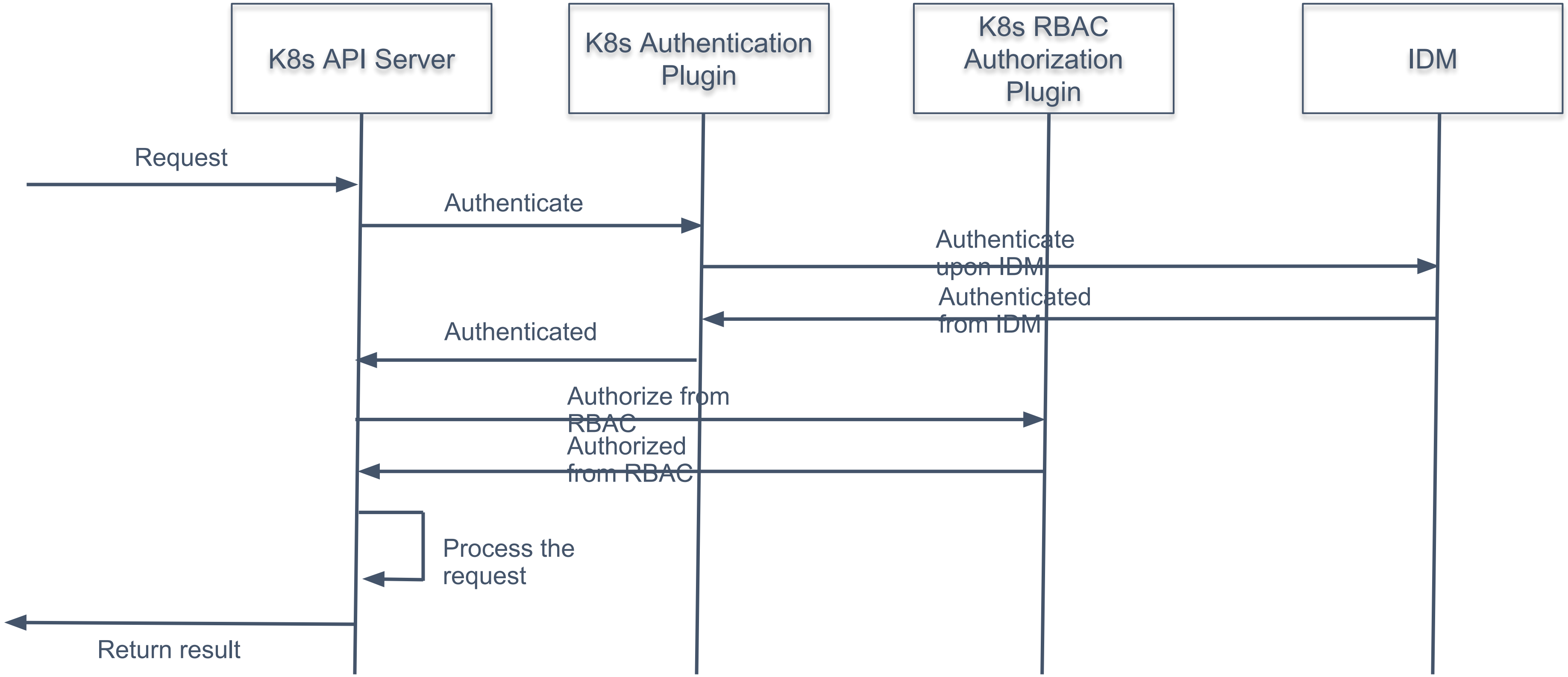
```
# This cluster role binding allows anyone in the "manager" group to read nodes in any namespace.
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: read-node-global
subjects:
- kind: Group
  name: manager
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: node-reader
  apiGroup: rbac.authorization.k8s.io
```

Role Binding

- Similar to Cluster Role Binding, however, the grant is limited within a namespace.
- Example

```yaml
# This role binding allows "jane" to read pods in the "default" namespace.
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: read-pods
  namespace: default
subjects:
- kind: User
  name: jane
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io
```

How all pieces work together



User and Security Models

Personas

Cloud Administrator

- Cluster Management
- User & Group Management
- Overall Operations & Logs

Application Development Team

DevOps Administrator

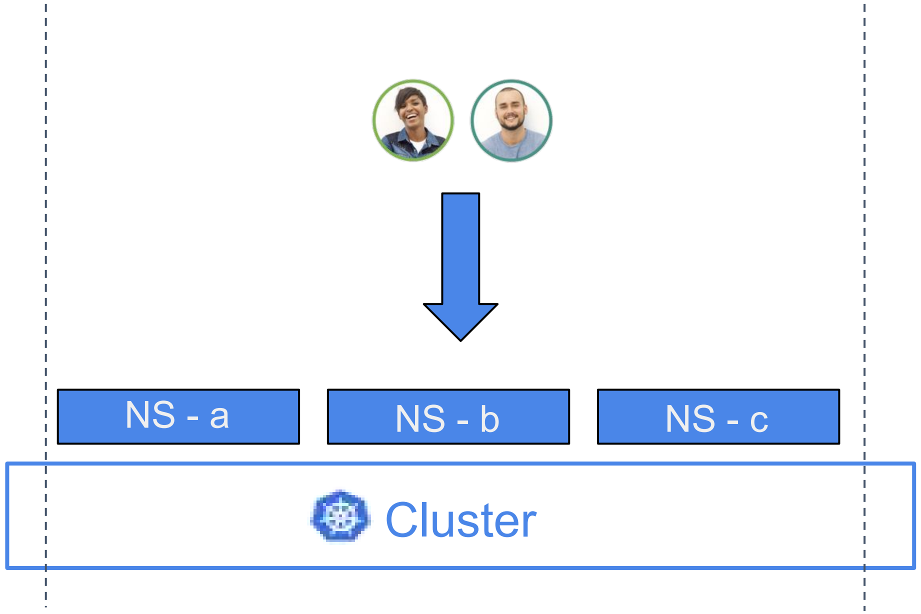
- Scale Clusters
- Reporting, Dashboard and Operations Management for the Project / Apps

Developer

- Consumer for K8s API
- Definition of Application Resources
- Application and image deployment
- Application Operation (App Ops)

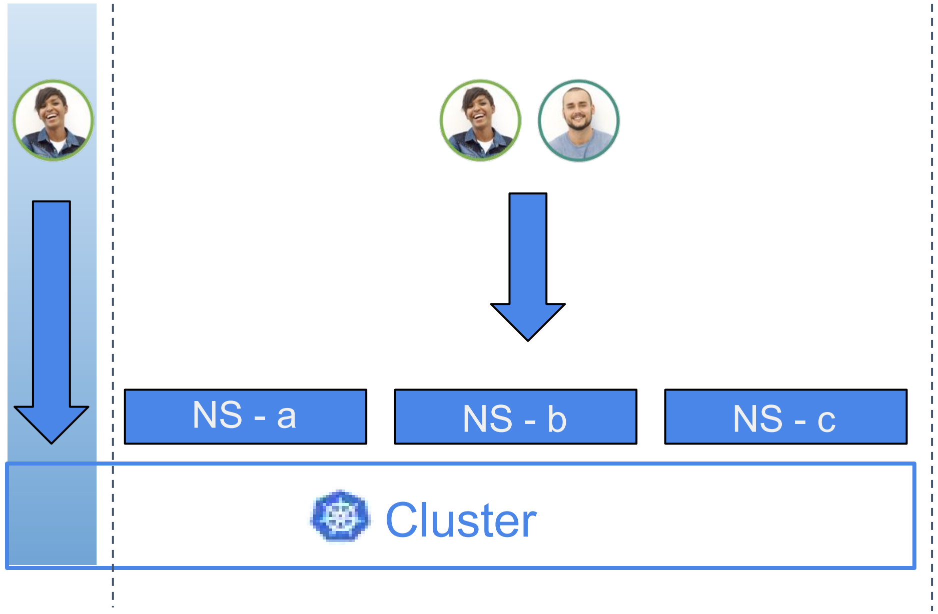
User and Security Model 1 - Exclusive Cluster

- Simplest Model
- Single tenancy
- Collapse the role of DevOps Admin and Developer.
- Cloud Admin have full control
  - User Access
  - Cluster Resources
- Any authorized user can create namespace.
- All namespaces and their resources are visible to all authorized users.
- Cluster resources are invisible to all users.



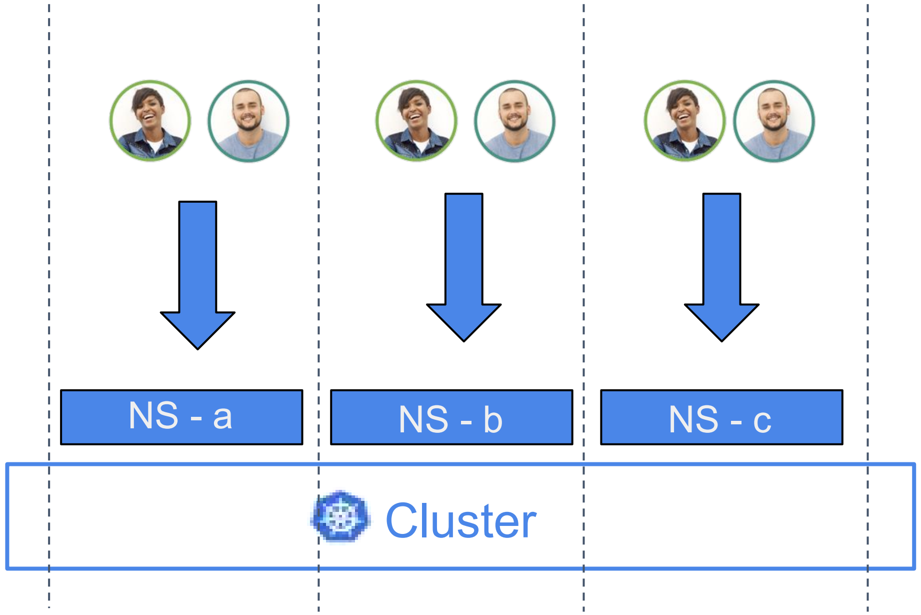
User and Security Model 1 - Variation of Exclusive Cluster

- Single tenancy
- Preserve the distinct role of DevOps Admin and Developer
- Cloud Admin still have full control
  - User Access
  - Cluster Resources
- Cloud admin delegates controls to DevOps admins on selected cluster level resources
- Any authorized users can create namespace.
- All namespaces and their resources are visible to all authorized users.
- Cluster resources are not visible to all authorized developers.



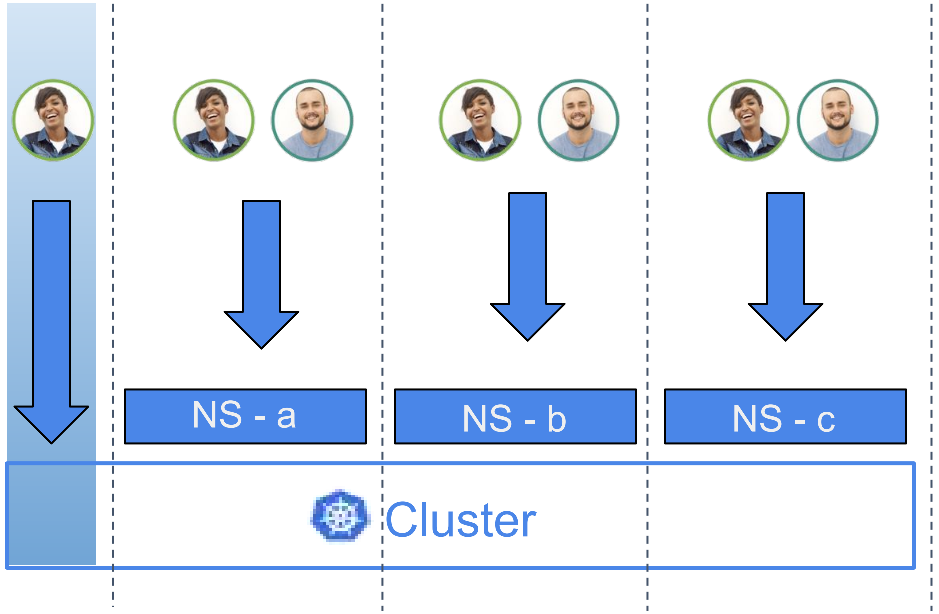
User and Security Model 2 - Shared Cluster

- Multi-tenancy support
- Collapse the role of DevOps Admin and Developer
- Cloud Admin has full control
  - User Access
  - Cluster Resources
- Only cloud admin can create namespace
- Resources under a namespace are visible to authorized users only
- Cluster resources are invisible to all users.



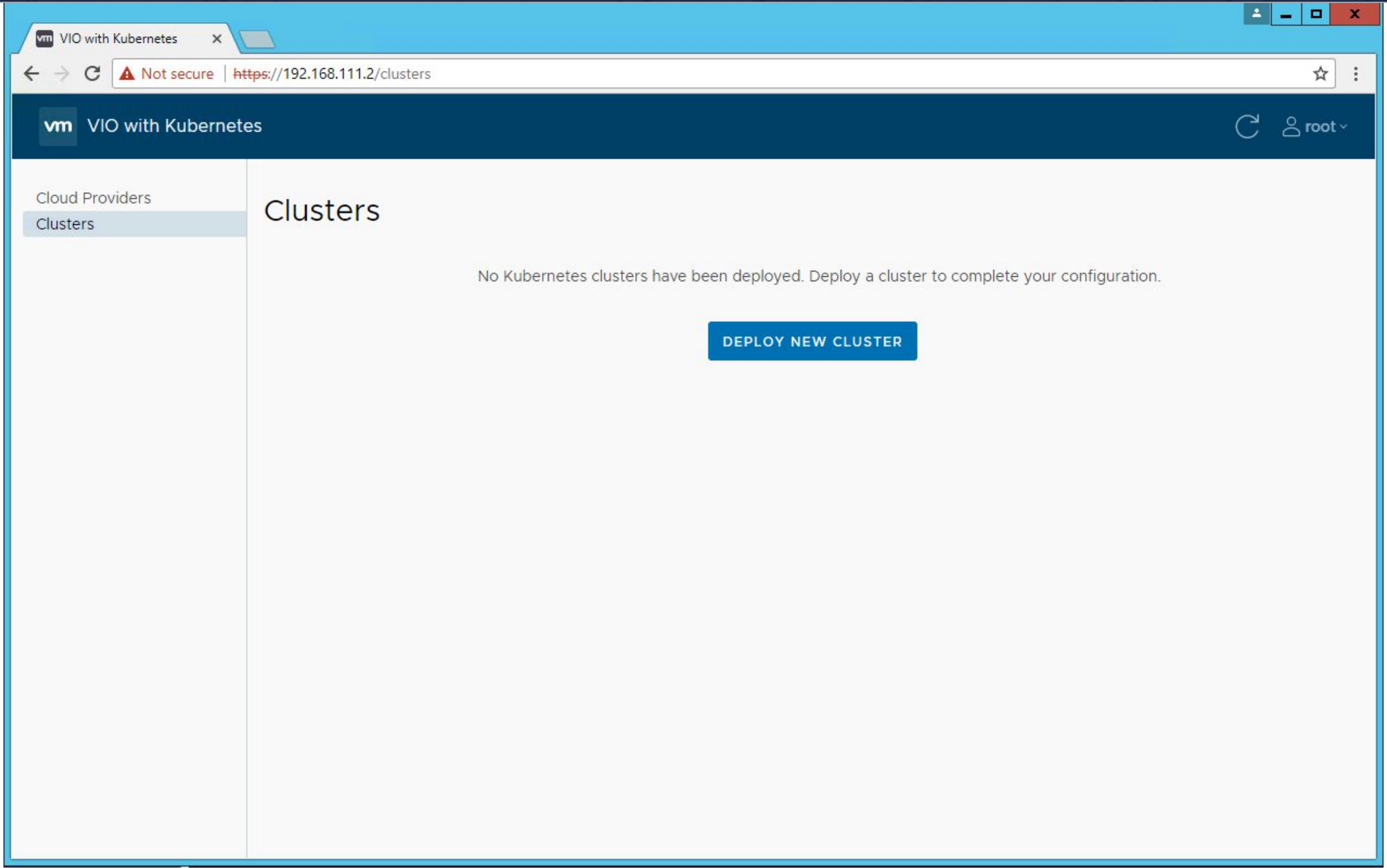
User and Security Model 2 - Variation of Shared Cluster

- Multi-tenancy support
- Preserve the distinct role of DevOps Admin and Developer
- Cloud Admin still has full control
  - User Access
  - Cluster Resources
- Cloud Admin delegates some controls to DevOps Admin on cluster level
- Cloud admin and DevOps Admin can create namespace
- Namespaces and their resources are visible to authorized developers only



Demo

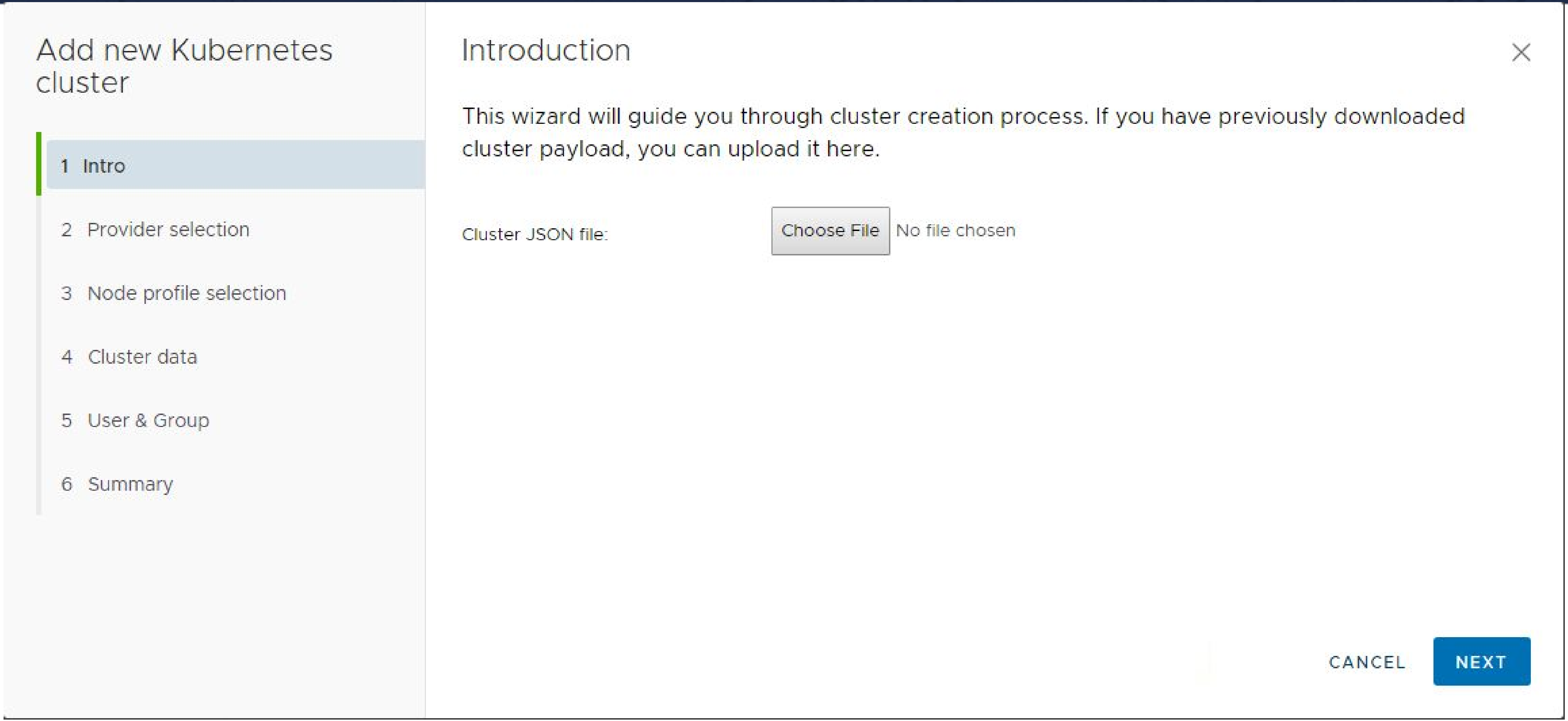
Creating Exclusive Cluster - 1/7



Clusters

No Kubernetes clusters have been deployed. Deploy a cluster to complete your configuration.

Creating Exclusive Cluster - 2/7

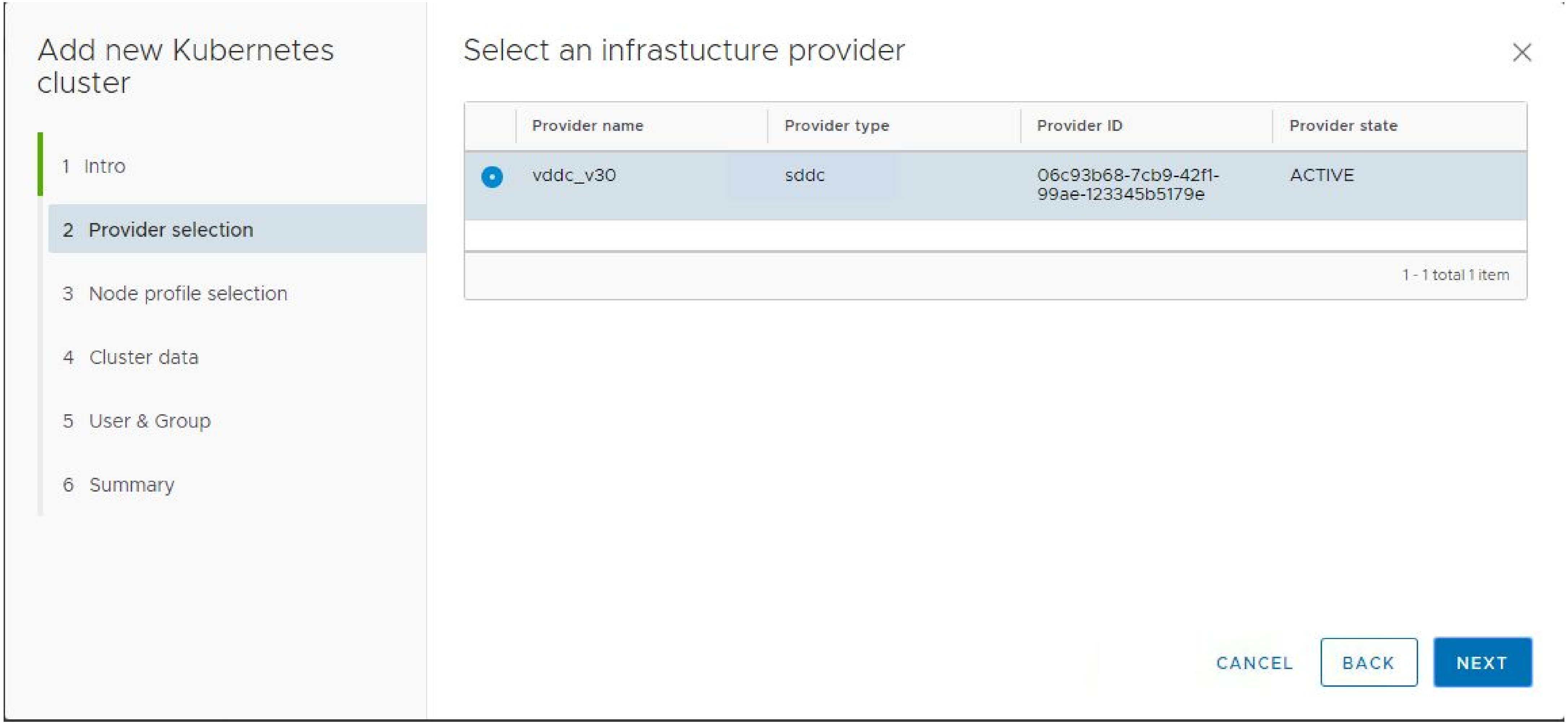


Introduction

This wizard will guide you through cluster creation process. If you have previously downloaded cluster payload, you can upload it here.

Cluster JSON file: Choose File No file chosen

Creating Exclusive Cluster - 3/7



Add new Kubernetes cluster

1 Intro

2 Provider selection

3 Node profile selection

4 Cluster data

5 User & Group

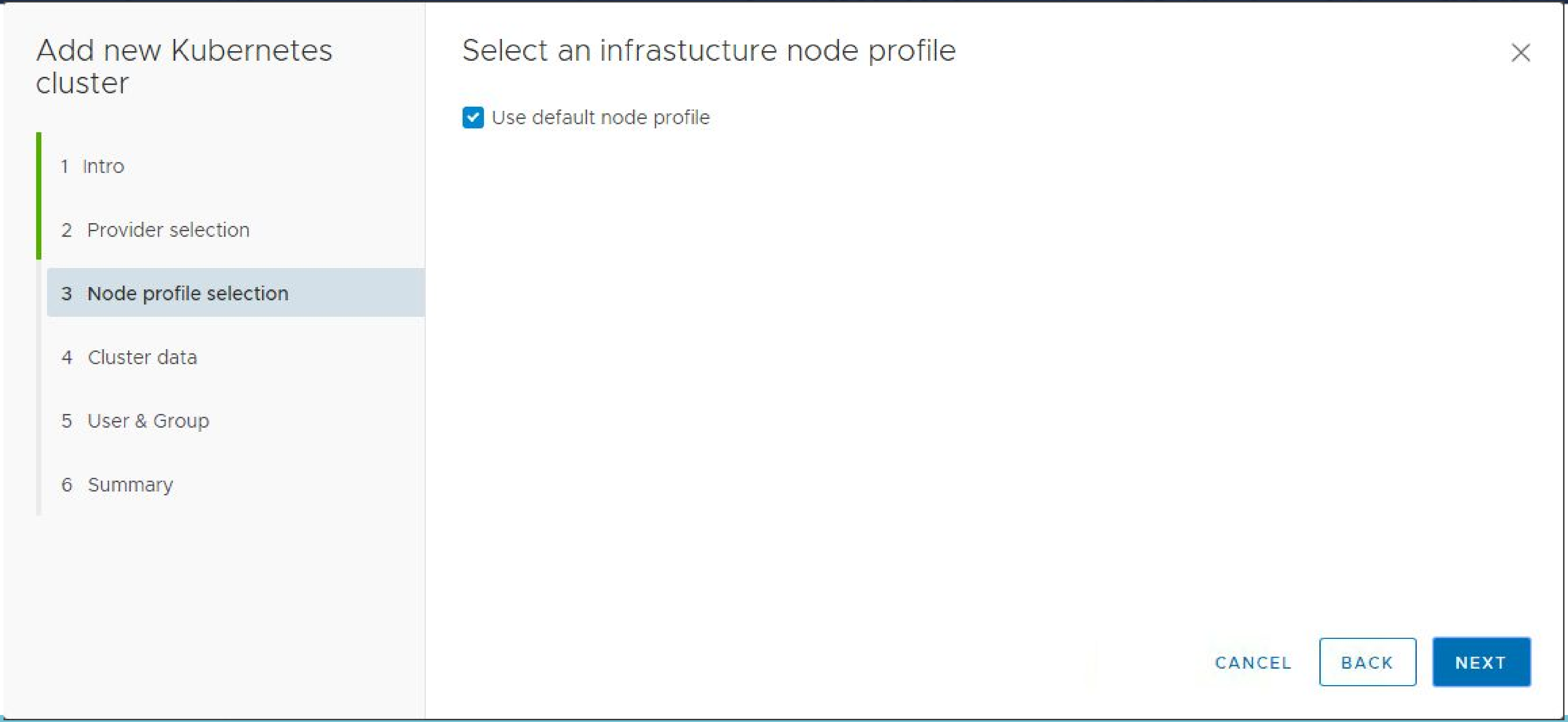
6 Summary

Select an infrastucture provider

	Provider name	Provider type	Provider ID	Provider state
●	vddc_v30	sddc	06c93b68-7cb9-42f1-99ae-123345b5179e	ACTIVE

1 - 1 total 1 item

Creating Exclusive Cluster - 4/7



Add new Kubernetes cluster

1 Intro

2 Provider selection

3 Node profile selection

4 Cluster data

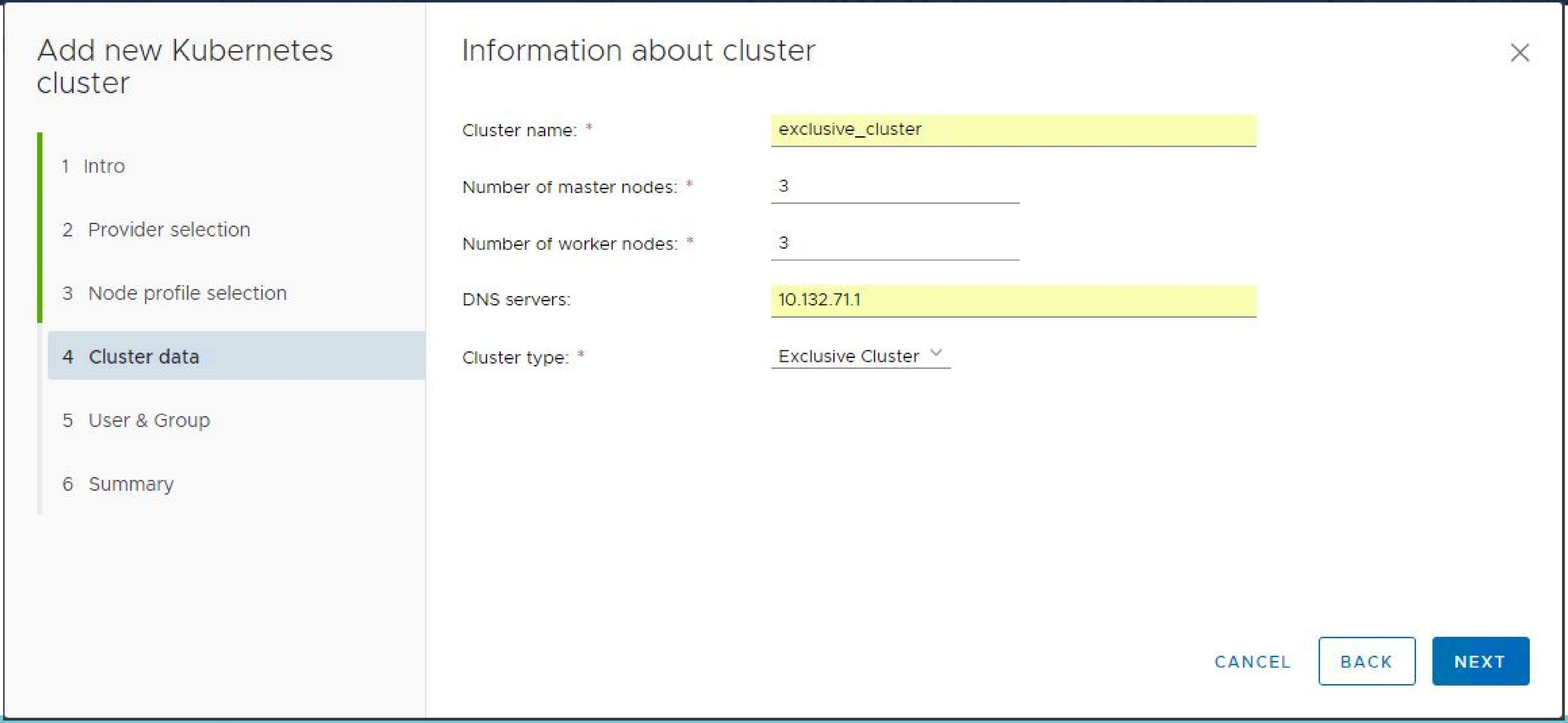
5 User & Group

6 Summary

Select an infrastucture node profile

☑ Use default node profile

Creating Exclusive Cluster - 5/7



Add new Kubernetes cluster

1 Intro

2 Provider selection

3 Node profile selection

4 Cluster data

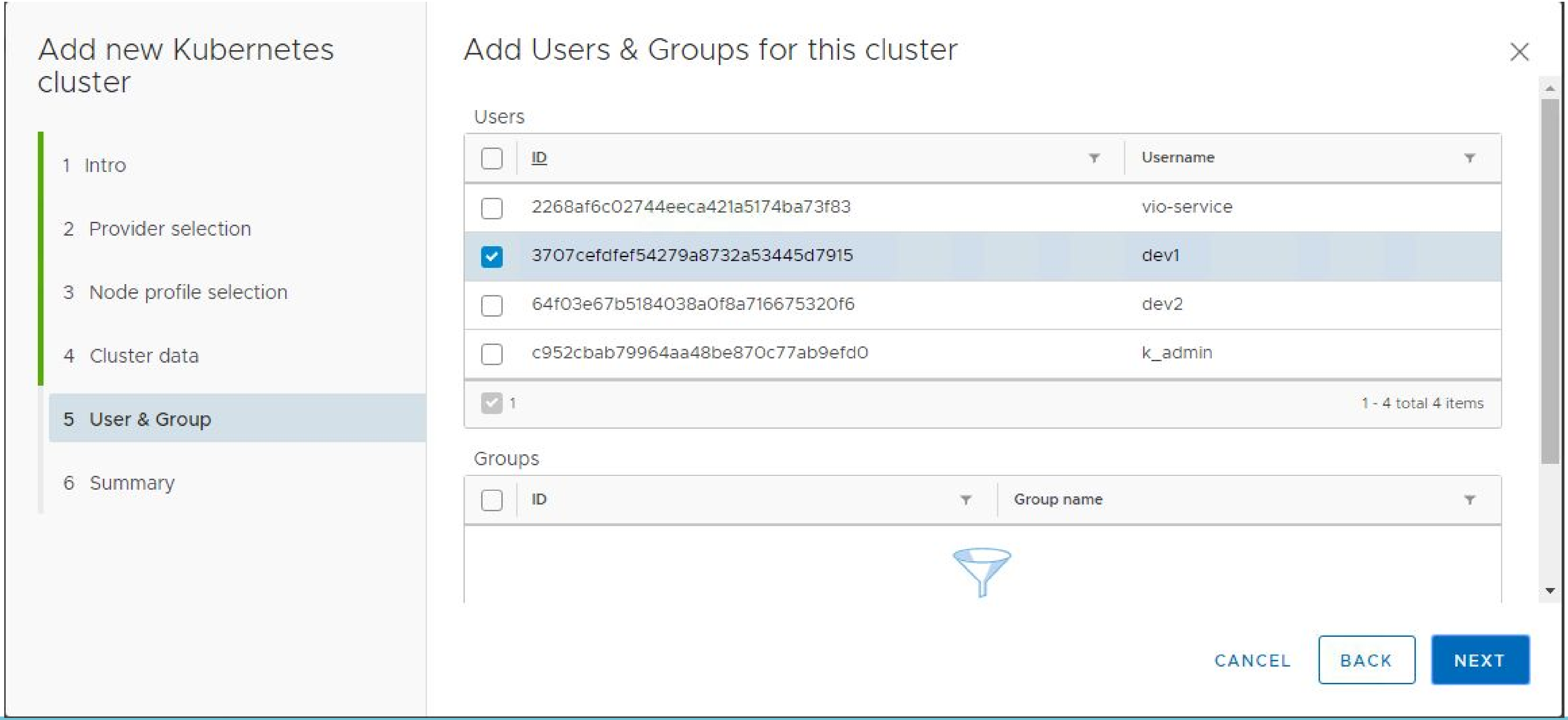
5 User & Group

6 Summary

Information about cluster

Cluster name: *	exclusive_cluster
Number of master nodes: *	3
Number of worker nodes: *	3
DNS servers:	10.132.71.1
Cluster type: *	Exclusive Cluster

Creating Exclusive Cluster - 6/7



Add new Kubernetes cluster

1 Intro

2 Provider selection

3 Node profile selection

4 Cluster data

5 User & Group

6 Summary

Add Users & Groups for this cluster

Users

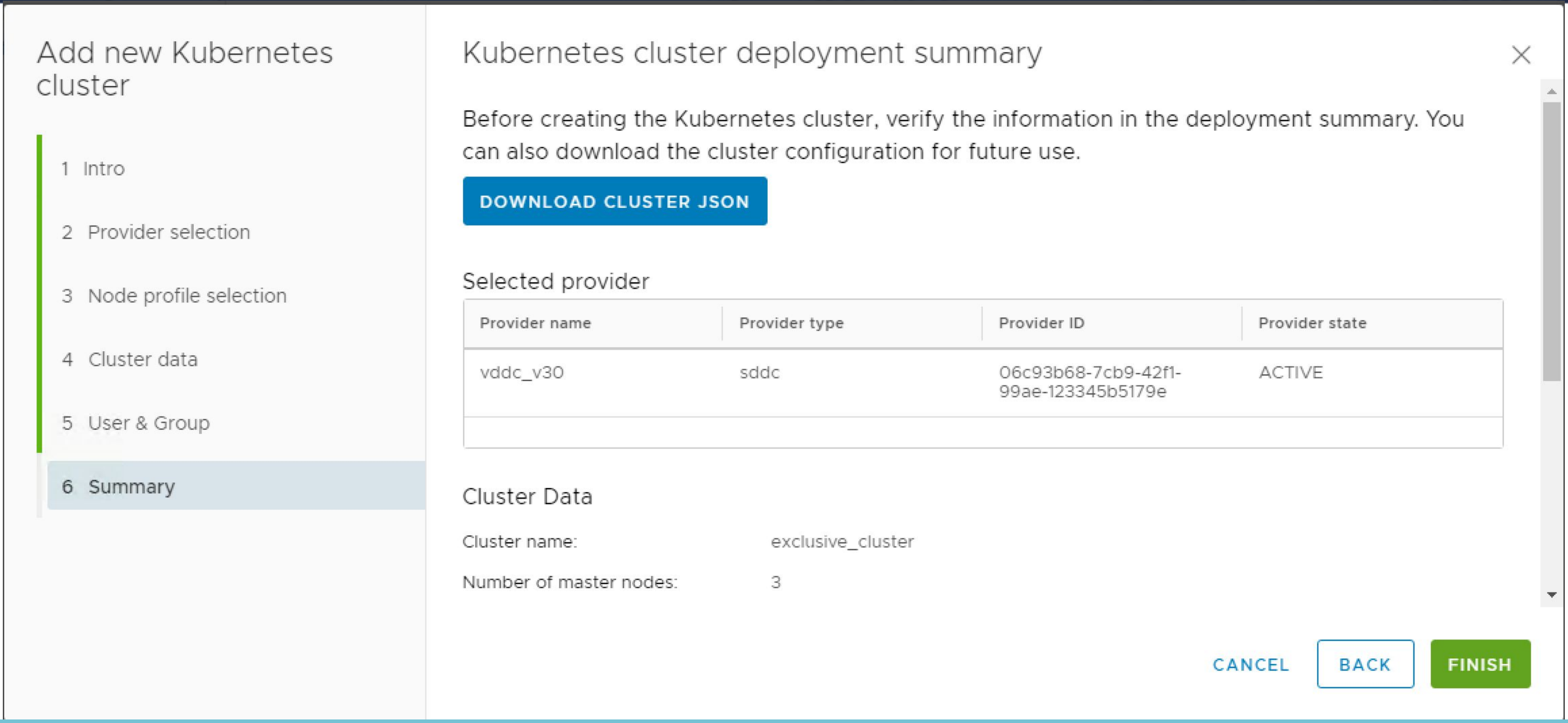
□	ID	Username
□	2268af6c02744eeca421a5174ba73f83	vio-service
☑	3707cefdfef54279a8732a53445d7915	dev1
□	64f03e67b5184038a0f8a716675320f6	dev2
□	c952cbab79964aa48be870c77ab9efd0	k_admin

☑ 1 1 - 4 total 4 items

Groups

□	ID	Group name

Creating Exclusive Cluster - 7/7



Add new Kubernetes cluster

Kubernetes cluster deployment summary

Before creating the Kubernetes cluster, verify the information in the deployment summary. You can also download the cluster configuration for future use.

Selected provider

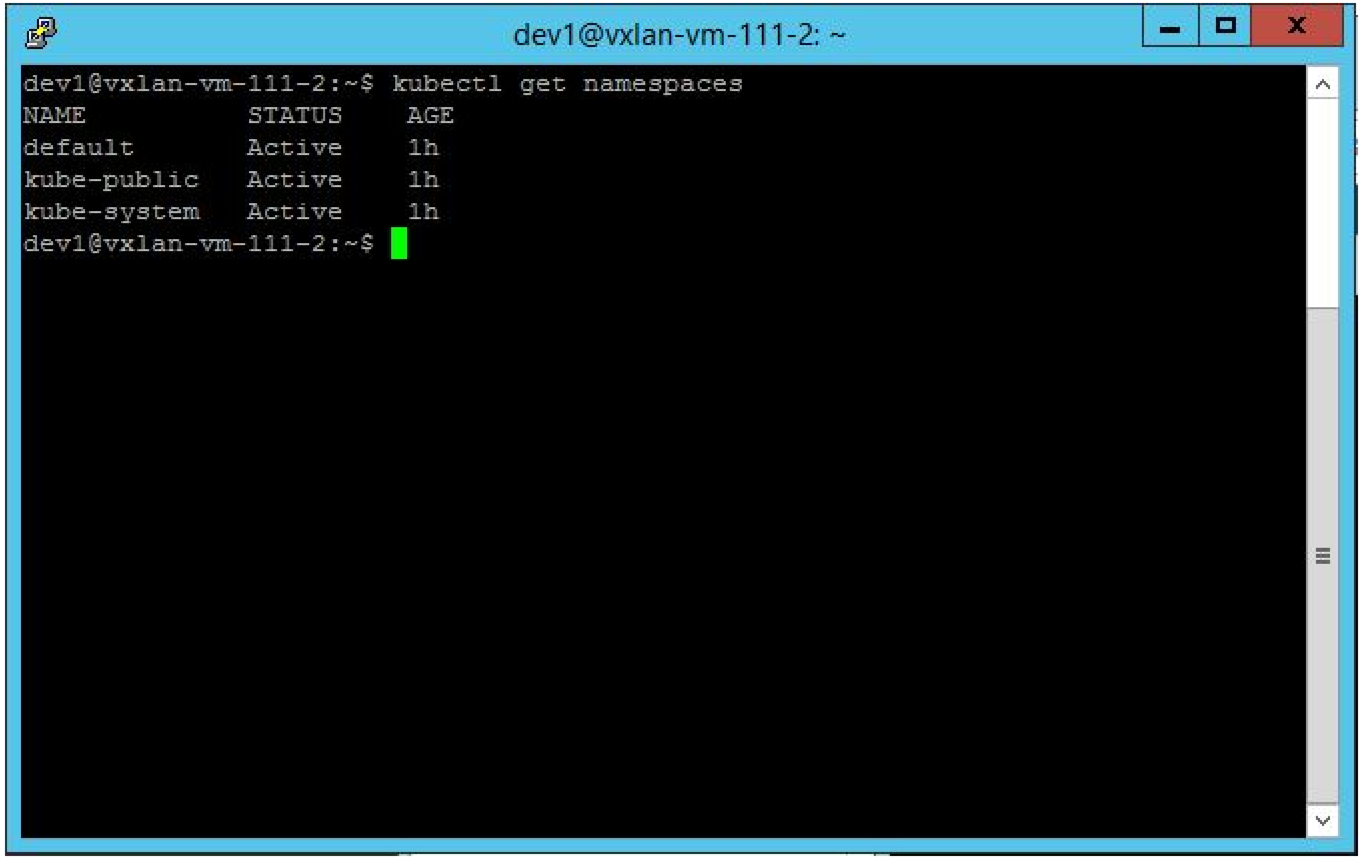
Provider name	Provider type	Provider ID	Provider state
vddc_v30	sddc	06c93b68-7cb9-42f1-99ae-123345b5179e	ACTIVE

Cluster Data

Cluster name: exclusive_cluster

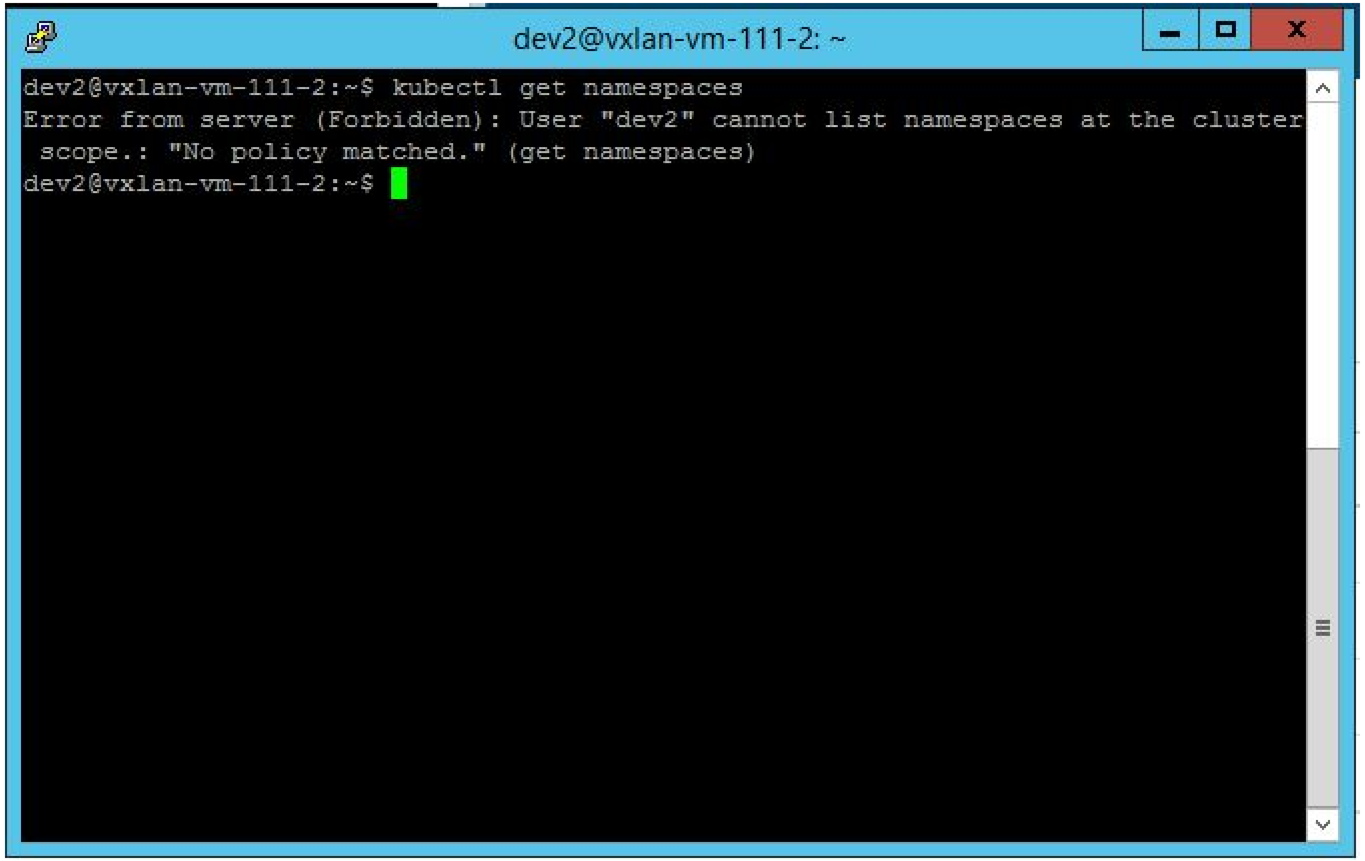
Number of master nodes: 3

User Dev1



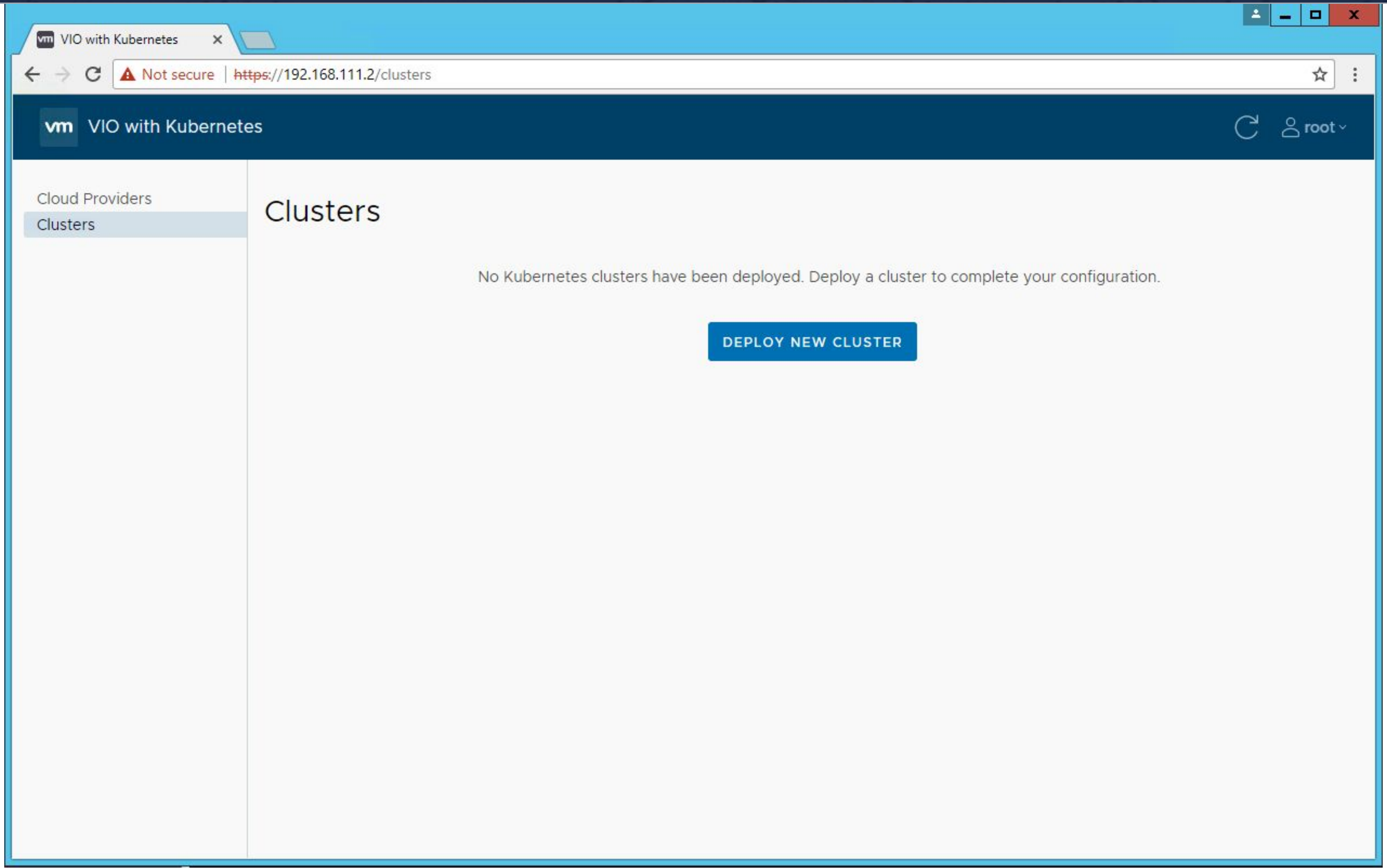
```
dev1@vxlan-vm-111-2:~$ kubectl get namespaces
NAME          STATUS    AGE
default       Active    1h
kube-public   Active    1h
kube-system   Active    1h
dev1@vxlan-vm-111-2:~$
```

User Dev2



```
dev2@vxlan-vm-111-2:~$ kubectl get namespaces
Error from server (Forbidden): User "dev2" cannot list namespaces at the cluster
 scope.: "No policy matched." (get namespaces)
dev2@vxlan-vm-111-2:~$
```

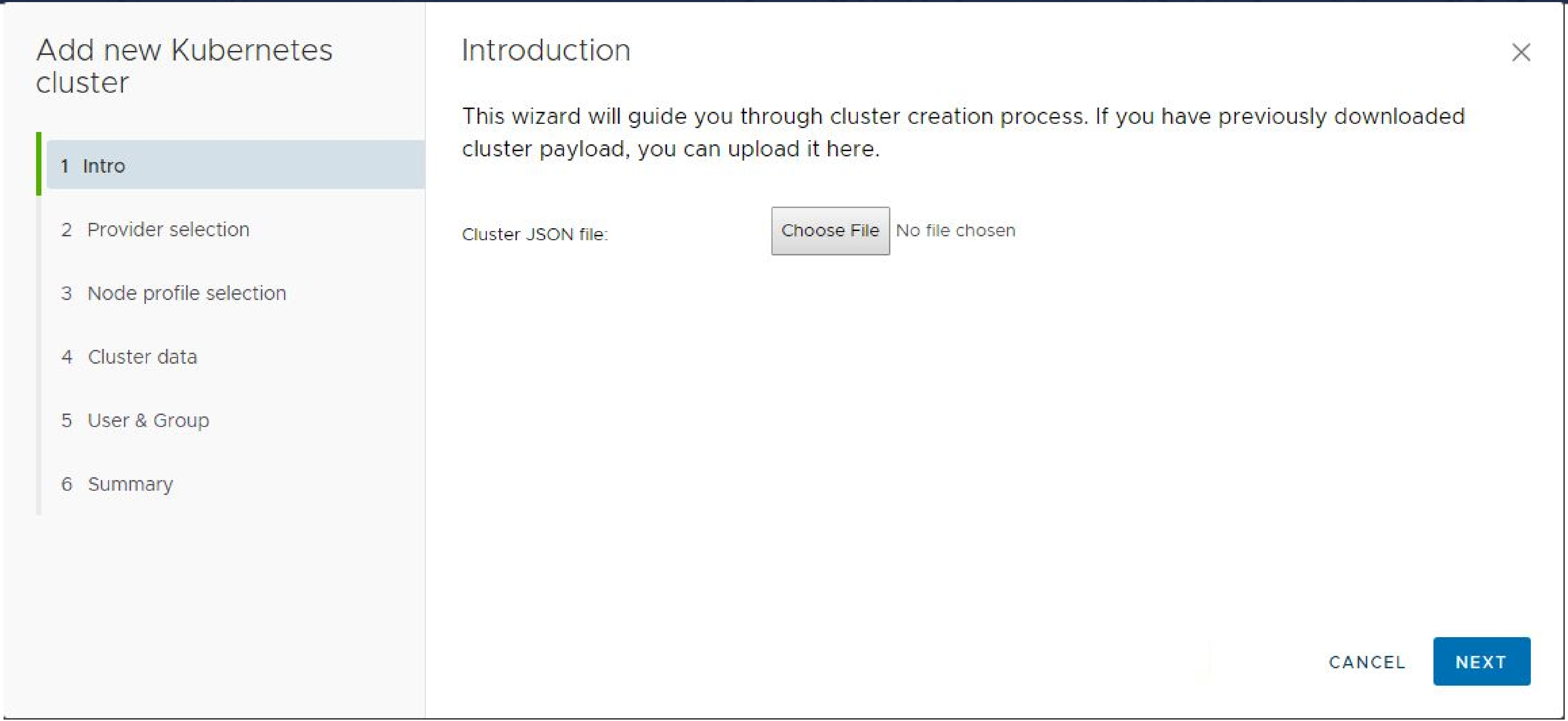
Creating Shared Cluster - 1/7



Clusters

No Kubernetes clusters have been deployed. Deploy a cluster to complete your configuration.

Creating Shared Cluster - 2/7



Add new Kubernetes cluster

1 Intro

2 Provider selection

3 Node profile selection

4 Cluster data

5 User & Group

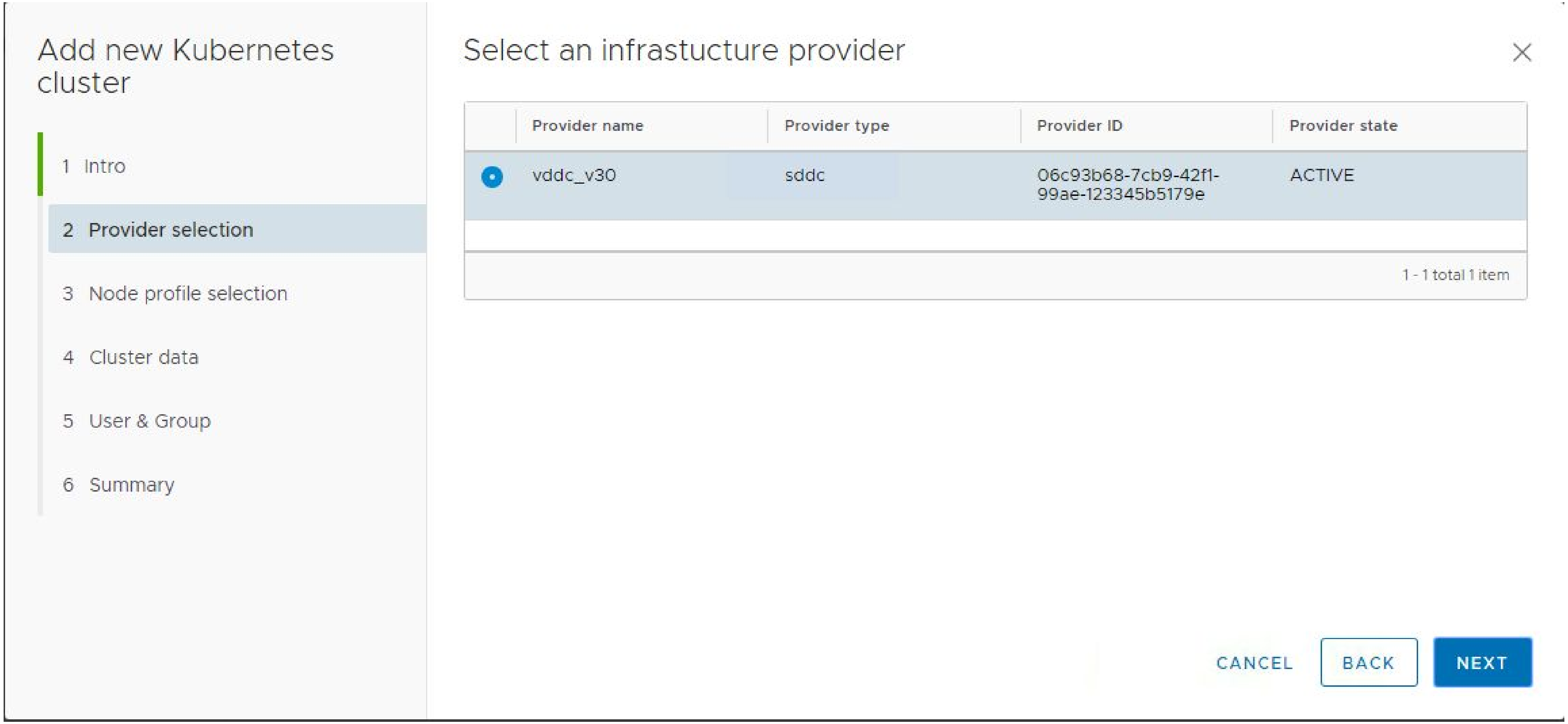
6 Summary

Introduction

This wizard will guide you through cluster creation process. If you have previously downloaded cluster payload, you can upload it here.

Cluster JSON file: Choose File No file chosen

Creating Shared Cluster - 3/7

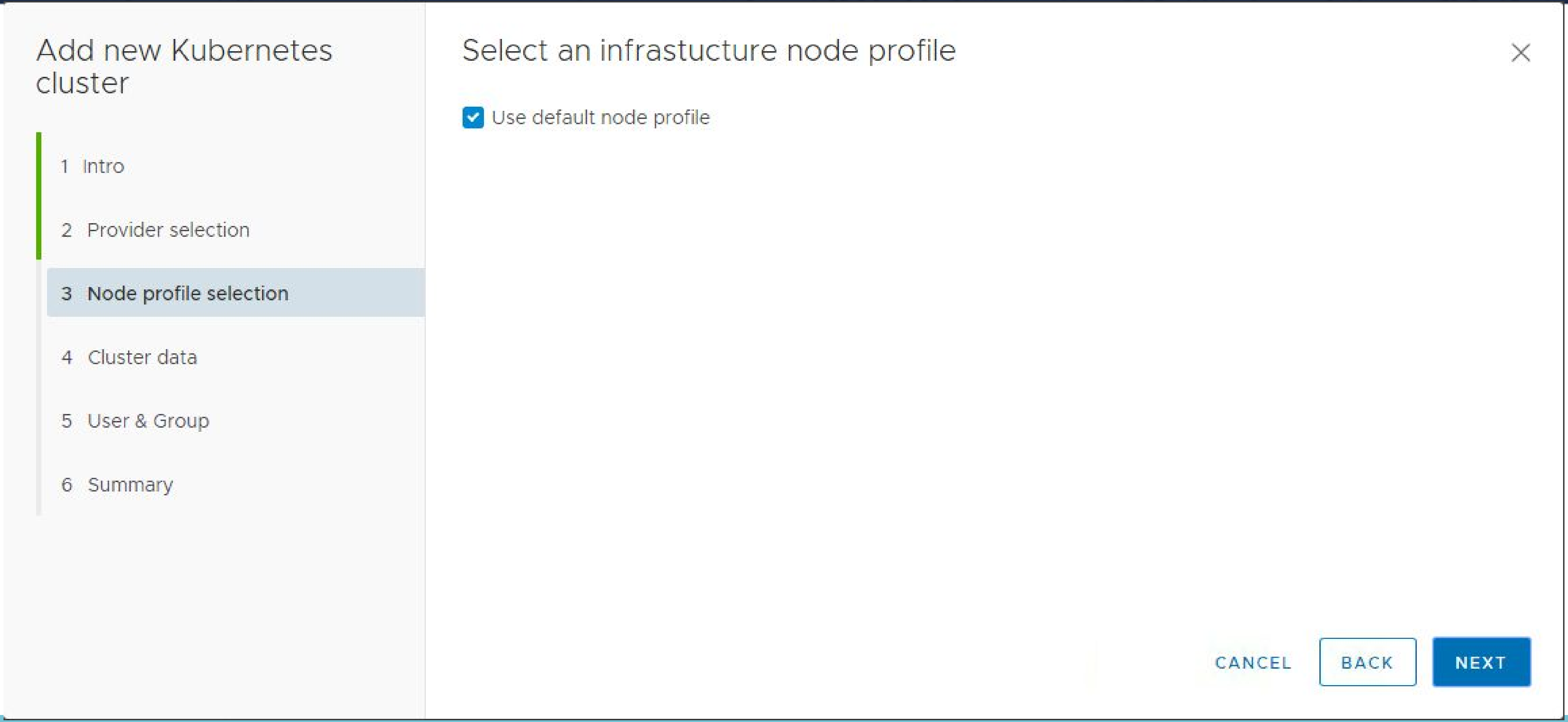


Select an infrastucture provider

	Provider name	Provider type	Provider ID	Provider state
●	vddc_v30	sddc	06c93b68-7cb9-42f1-99ae-123345b5179e	ACTIVE

1 - 1 total 1 item

Creating Shared Cluster - 4/7



Add new Kubernetes cluster

1 Intro

2 Provider selection

3 Node profile selection

4 Cluster data

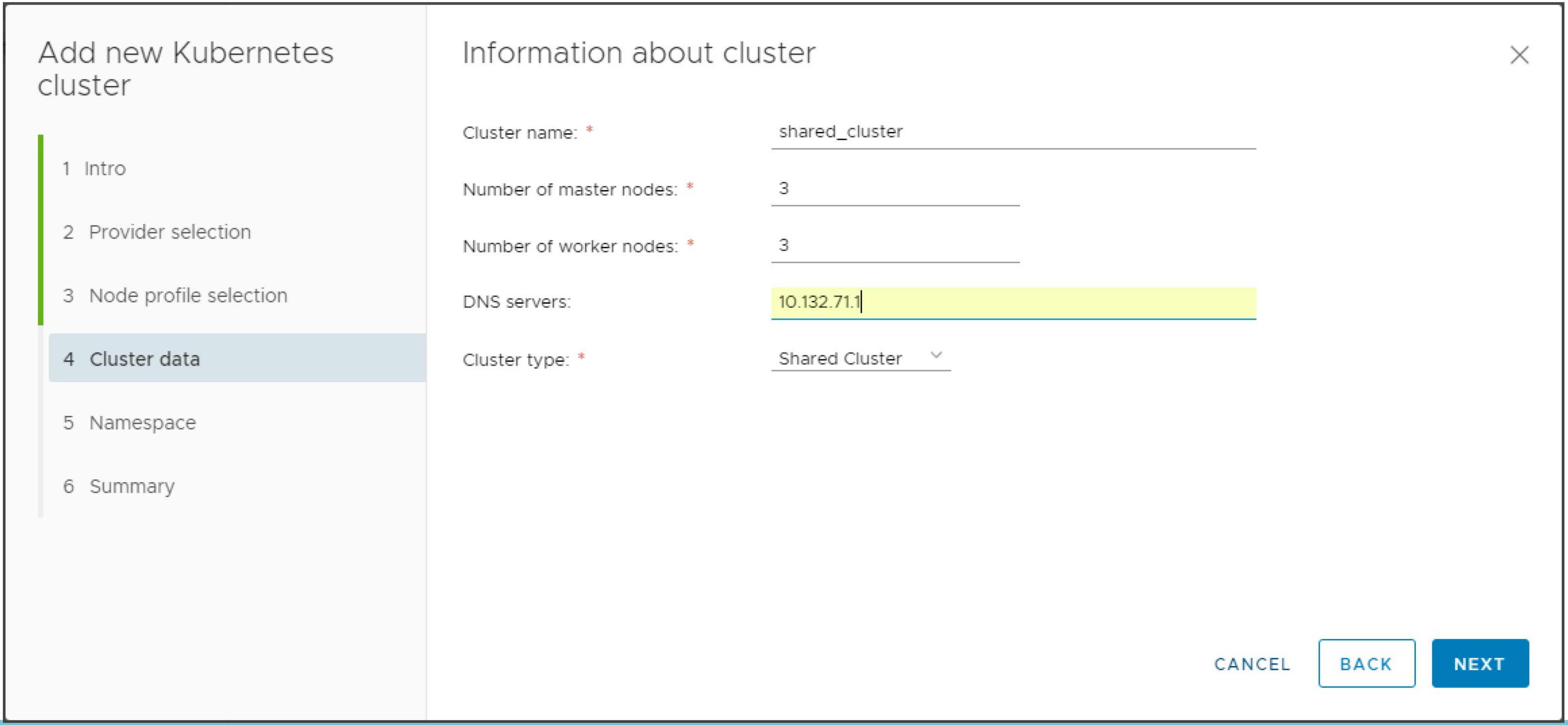
5 User & Group

6 Summary

Select an infrastucture node profile

☑ Use default node profile

Creating Shared Cluster - 5/7



Add new Kubernetes cluster

1 Intro

2 Provider selection

3 Node profile selection

4 Cluster data

5 Namespace

6 Summary

Information about cluster

Cluster name: * shared_cluster

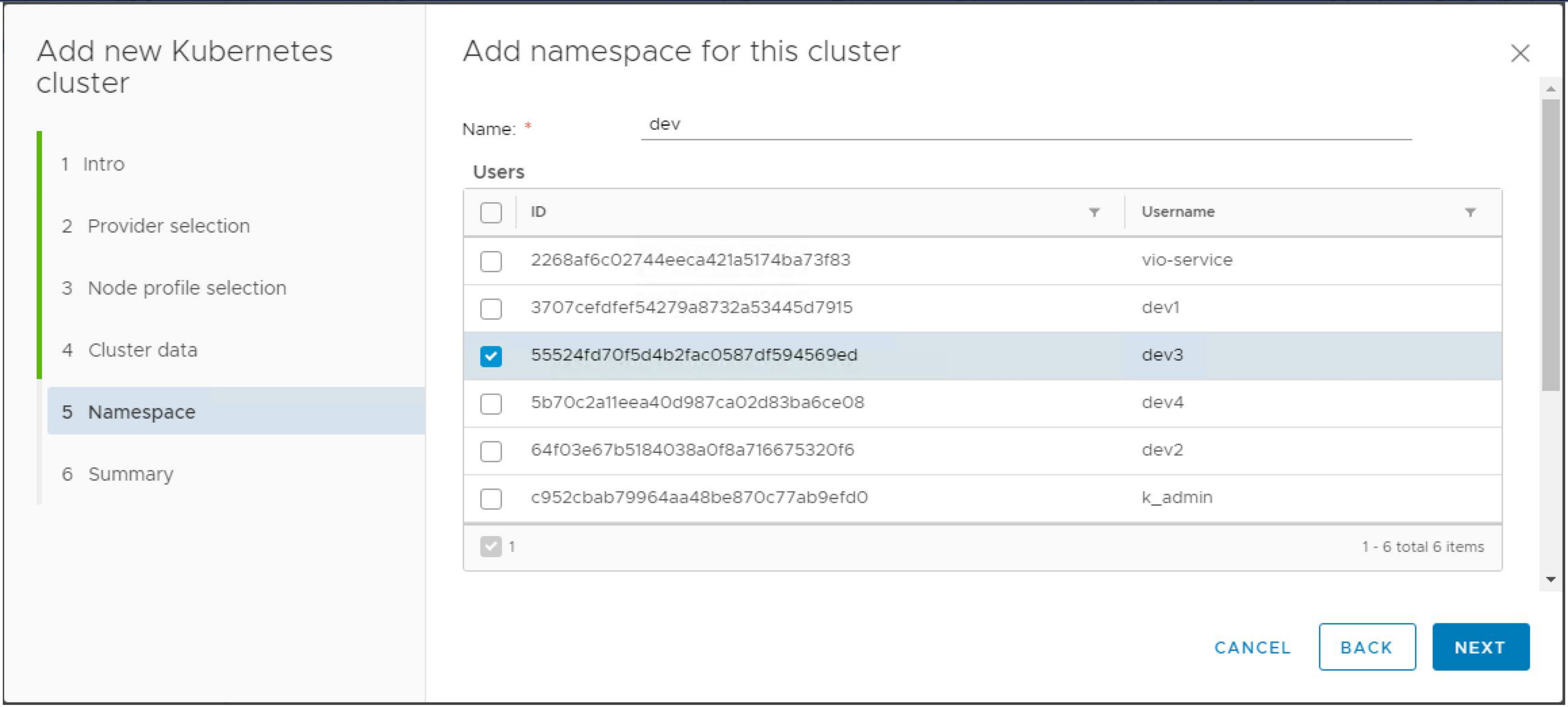
Number of master nodes: * 3

Number of worker nodes: * 3

DNS servers: 10.132.71.1

Cluster type: * Shared Cluster

Creating Shared Cluster - 6/7



Add new Kubernetes cluster

1 Intro

2 Provider selection

3 Node profile selection

4 Cluster data

5 Namespace

6 Summary

Add namespace for this cluster

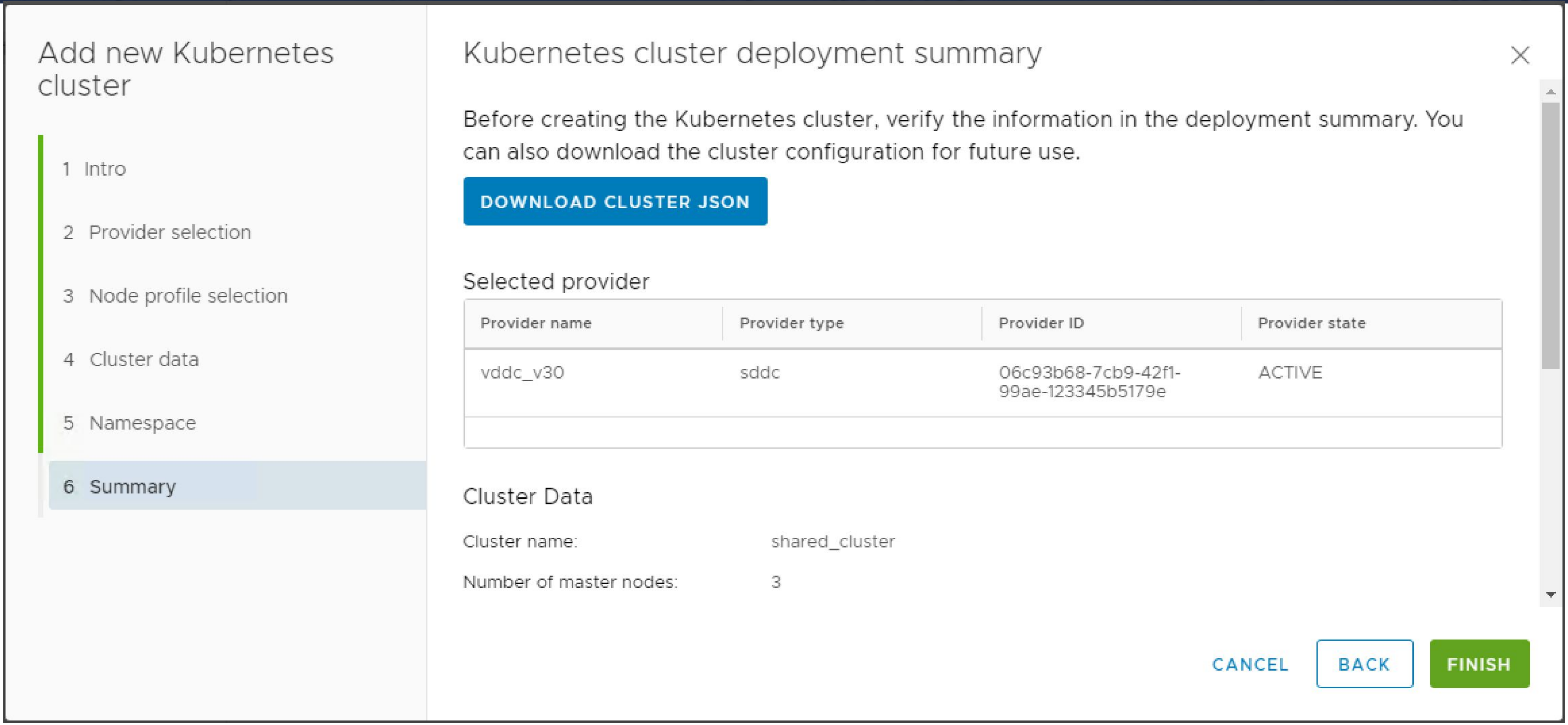
Name: * dev

Users

□	ID	Username
□	2268af6c02744eeca421a5174ba73f83	vio-service
□	3707cefdfef54279a8732a53445d7915	dev1
☑	55524fd70f5d4b2fac0587df594569ed	dev3
□	5b70c2a11eea40d987ca02d83ba6ce08	dev4
□	64f03e67b5184038a0f8a716675320f6	dev2
□	c952cbab79964aa48be870c77ab9efd0	k_admin

☑ 1 1 - 6 total 6 items

Creating Shared Cluster - 7/7



Add new Kubernetes cluster

1 Intro

2 Provider selection

3 Node profile selection

4 Cluster data

5 Namespace

6 Summary

Kubernetes cluster deployment summary

Before creating the Kubernetes cluster, verify the information in the deployment summary. You can also download the cluster configuration for future use.

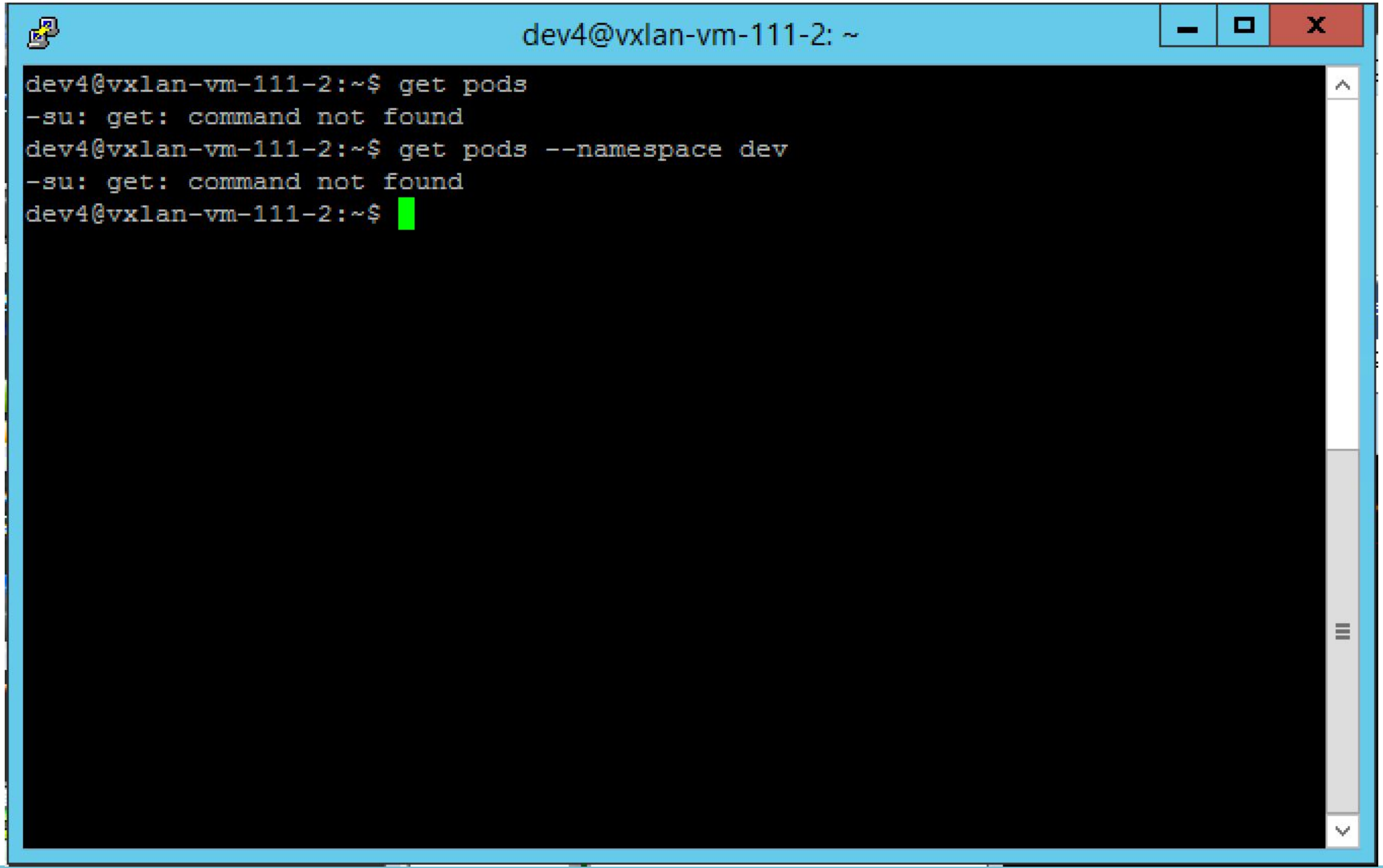
Selected provider

Provider name	Provider type	Provider ID	Provider state
vddc_v30	sddc	06c93b68-7cb9-42f1-99ae-123345b5179e	ACTIVE

Cluster Data

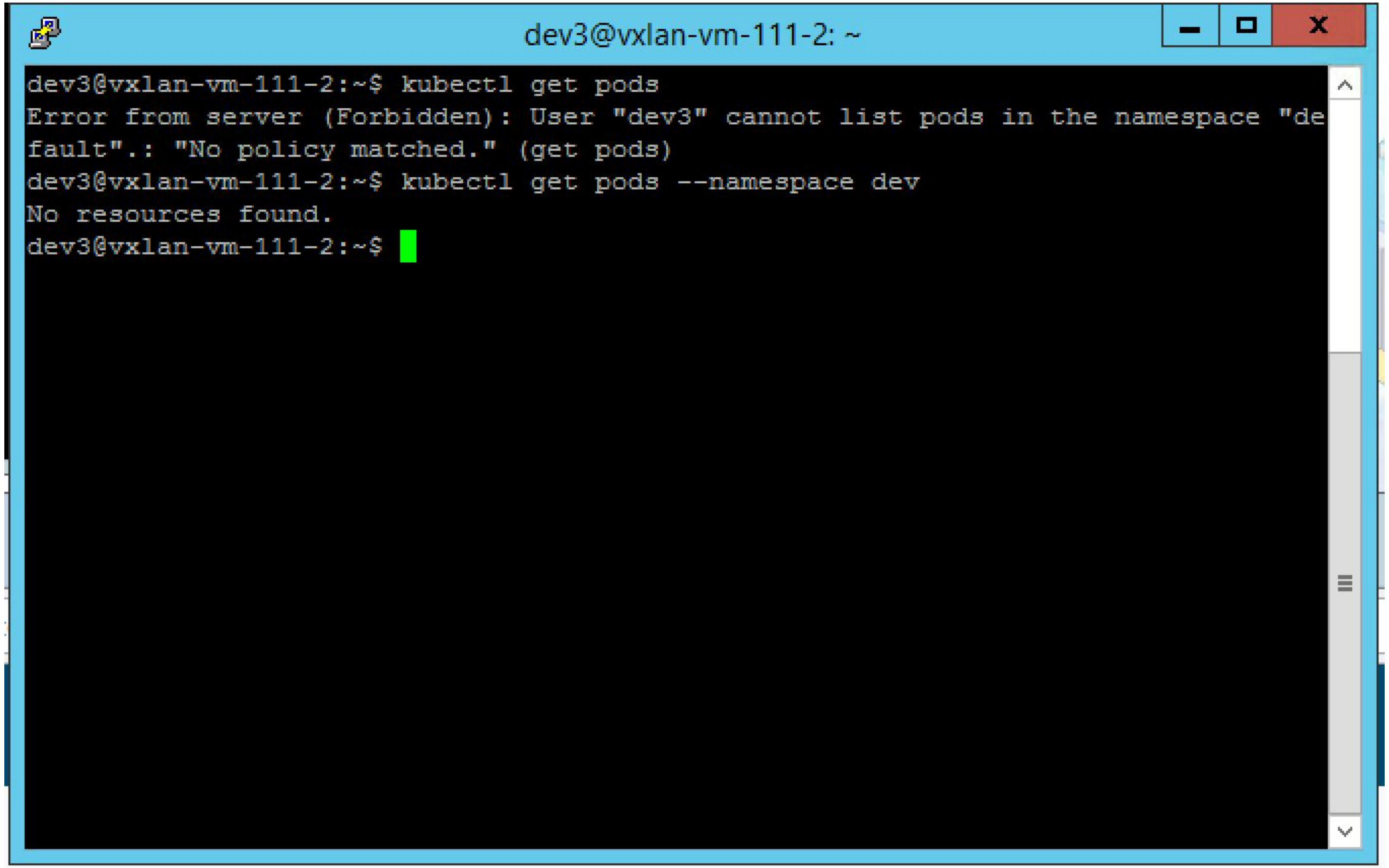
Cluster name:	shared_cluster
Number of master nodes:	3

User dev3



```
dev4@vxlan-vm-111-2:~$ get pods
-su: get: command not found
dev4@vxlan-vm-111-2:~$ get pods --namespace dev
-su: get: command not found
dev4@vxlan-vm-111-2:~$
```

User dev4



```
dev3@vxlan-vm-111-2:~$ kubectl get pods
Error from server (Forbidden): User "dev3" cannot list pods in the namespace "de
fault".: "No policy matched." (get pods)
dev3@vxlan-vm-111-2:~$ kubectl get pods --namespace dev
No resources found.
dev3@vxlan-vm-111-2:~$
```

References

References:

- https://kubernetes.io/docs/admin/authorization/rbac
- https://blogs.vmware.com/openstack/openstack-kubernetes-better-together/
- PKS: https://cloud.vmware.com/pivotal-container-service

We are hiring:

- http://bit.ly/vmwarekubecon

Questions?

Thank You

Frederick Vong, fvong@vmware.com

Michael Chen, michaelchen@vmware.com