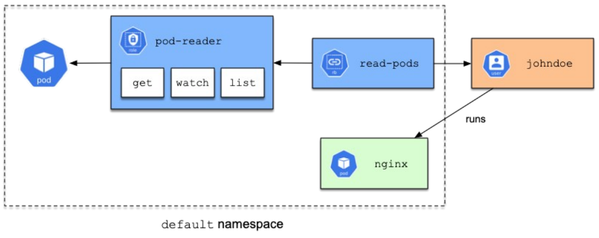
Lab04 – Deployment

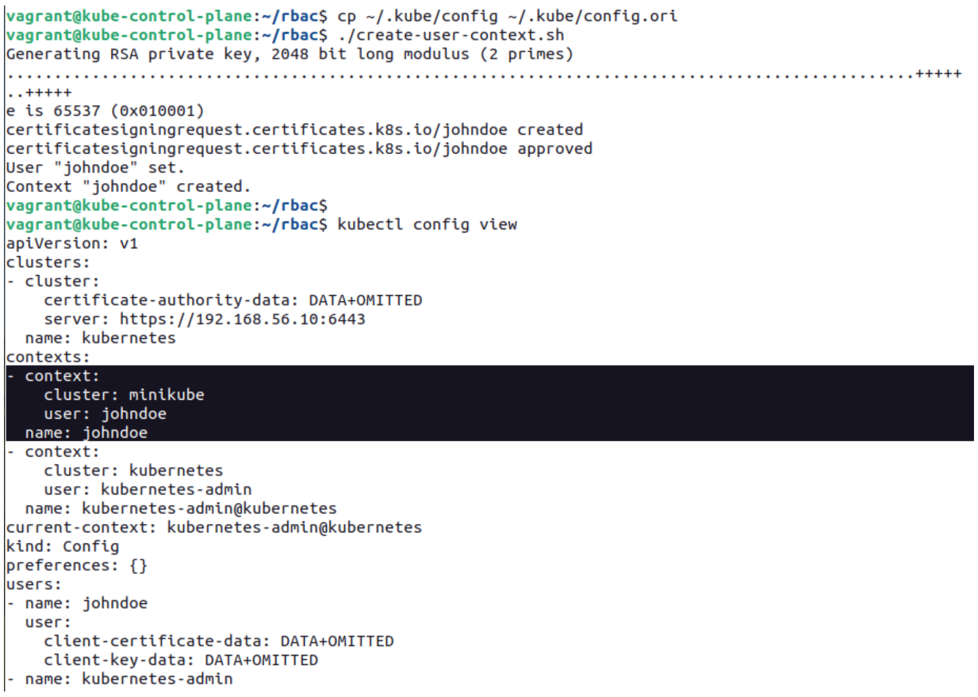
In this exercise, you will define Role Based Access Control (RBAC) to grant permissions to a specific user. The permissions should only apply to certain API resources and operations.

The following image shows the high-level architecture.



Creating the User

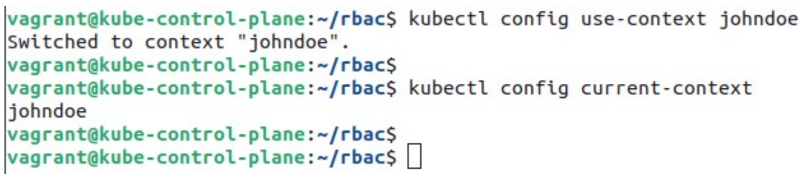
1. Save `~/.kube/config` file to `~/.kube/config.ori`, then run the script `create-user-context.sh`. It will achieve the following:
   - Create a private key.
   - Create and approve a CertificateSigningRequest.
   - Add a context entry named `johndoe` to the kubeconfig file to represent the user.



```
vagrant@kube-control-plane:~/rbac$ cp ~/.kube/config ~/.kube/config.ori
vagrant@kube-control-plane:~/rbac$ ./create-user-context.sh
Generating RSA private key, 2048 bit long modulus (2 primes)
...............................................................................................+++++
..+++++
e is 65537 (0x010001)
certificatesigningrequest.certificates.k8s.io/johndoe created
certificatesigningrequest.certificates.k8s.io/johndoe approved
User "johndoe" set.
Context "johndoe" created.
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://192.168.56.10:6443
  name: kubernetes
contexts:
- context:
    cluster: minikube
    user: johndoe
  name: johndoe
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: johndoe
  user:
    client-certificate-data: DATA+OMITTED
    client-key-data: DATA+OMITTED
- name: kubernetes-admin
```

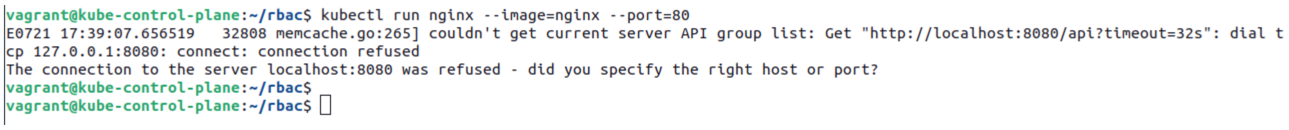
Checking Default User Permissions

2. Change to the context to `johndoe`. Use the `johndoe` context we added to the `kubeconfig` file earlier. You can check the current context using `config current-context`.



```
vagrant@kube-control-plane:~/rbac$ kubectl config use-context johndoe
Switched to context "johndoe".
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$ kubectl config current-context
johndoe
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$
```

3. Create a new Pod. What would you expect to happen?

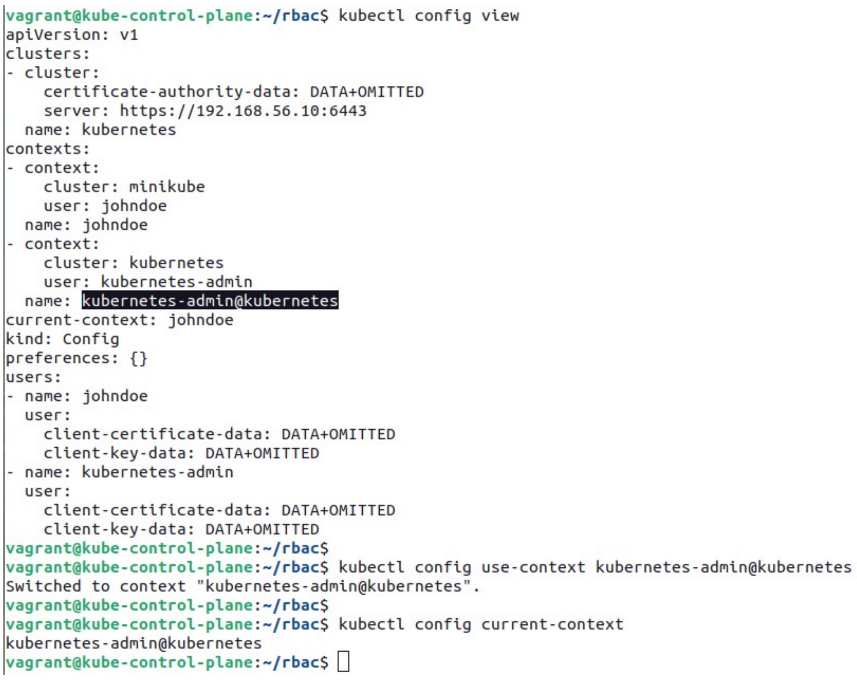


```
vagrant@kube-control-plane:~/rbac$ kubectl run nginx --image=nginx --port=80
E0721 17:39:07.656519   32808 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial t
cp 127.0.0.1:8080: connect: connection refused
The connection to the server localhost:8080 was refused - did you specify the right host or port?
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$
```

Creating a Pod in the current context won't work as the user `johndoe` does have the proper permissions.

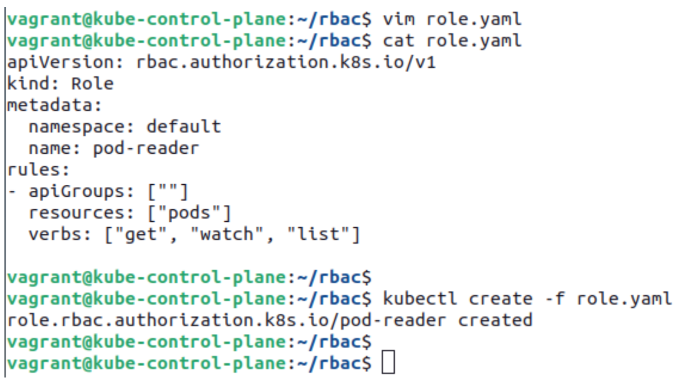
Granting Access to the User

4. Switch back to the original context with admin permissions.



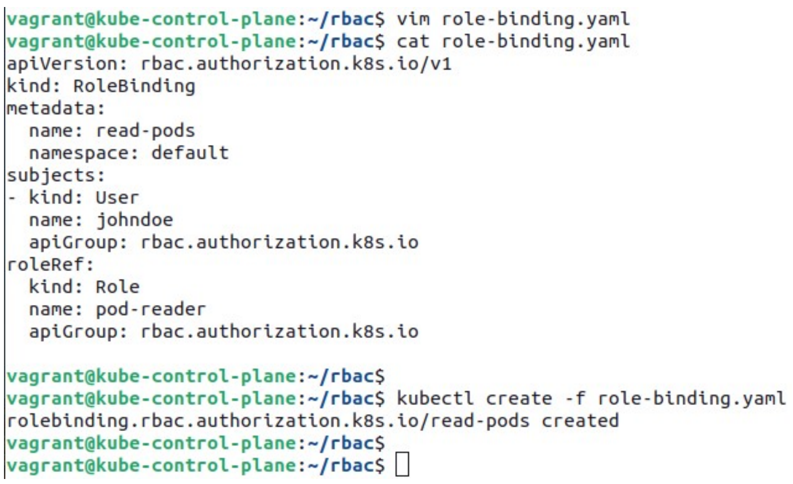
```
vagrant@kube-control-plane:~/rbac$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://192.168.56.10:6443
  name: kubernetes
contexts:
- context:
    cluster: minikube
    user: johndoe
  name: johndoe
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: johndoe
kind: Config
preferences: {}
users:
- name: johndoe
  user:
    client-certificate-data: DATA+OMITTED
    client-key-data: DATA+OMITTED
- name: kubernetes-admin
  user:
    client-certificate-data: DATA+OMITTED
    client-key-data: DATA+OMITTED
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$ kubectl config use-context kubernetes-admin@kubernetes
Switched to context "kubernetes-admin@kubernetes".
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$ kubectl config current-context
kubernetes-admin@kubernetes
vagrant@kube-control-plane:~/rbac$
```

5. Create a new Role named `pod-reader`. The Role should grant permissions to get, watch and list Pods.



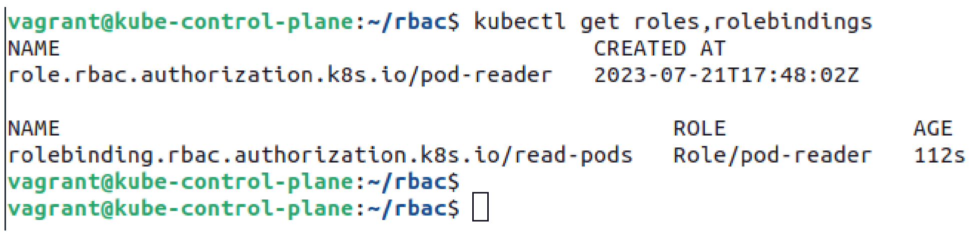
```
vagrant@kube-control-plane:~/rbac$ vim role.yaml
vagrant@kube-control-plane:~/rbac$ cat role.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: default
  name: pod-reader
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "watch", "list"]

vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$ kubectl create -f role.yaml
role.rbac.authorization.k8s.io/pod-reader created
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$
```

6. Create a new RoleBinding named `read-pods`. Map the user `johndoe` to the Role named `pod-reader`.



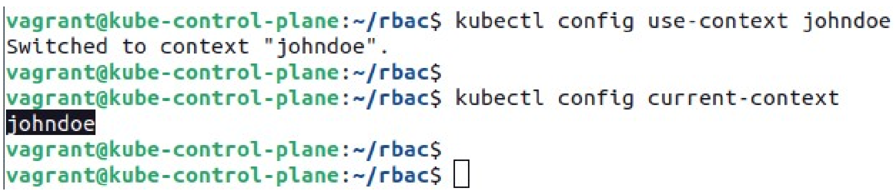
```
vagrant@kube-control-plane:~/rbac$ vim role-binding.yaml
vagrant@kube-control-plane:~/rbac$ cat role-binding.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: read-pods
  namespace: default
subjects:
- kind: User
  name: johndoe
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io

vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$ kubectl create -f role-binding.yaml
rolebinding.rbac.authorization.k8s.io/read-pods created
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$
```

7. Make sure that both objects have been created properly.



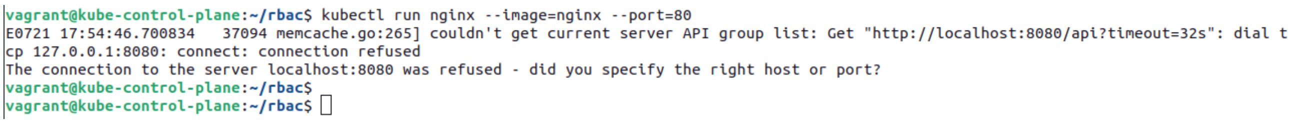
```
vagrant@kube-control-plane:~/rbac$ kubectl get roles,rolebindings
NAME                                           CREATED AT
role.rbac.authorization.k8s.io/pod-reader     2023-07-21T17:48:02Z

NAME                                                 ROLE                AGE
rolebinding.rbac.authorization.k8s.io/read-pods     Role/pod-reader     112s
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$
```

8. Switch to the context named `johndoe`.



```
vagrant@kube-control-plane:~/rbac$ kubectl config use-context johndoe
Switched to context "johndoe".
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$ kubectl config current-context
johndoe
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$
```

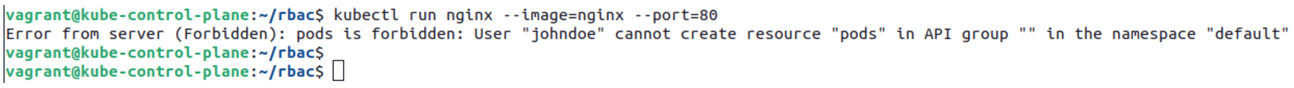
9. Create a new Pod named `nginx` with the image `nginx`. What would you expect to happen?



```
vagrant@kube-control-plane:~/rbac$ kubectl run nginx --image=nginx --port=80
E0721 17:54:46.700834   37094 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial t
cp 127.0.0.1:8080: connect: connection refused
The connection to the server localhost:8080 was refused - did you specify the right host or port?
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$
```

Creating a new Pod won't work as the user `johndoe` doesn't have the proper permissions.

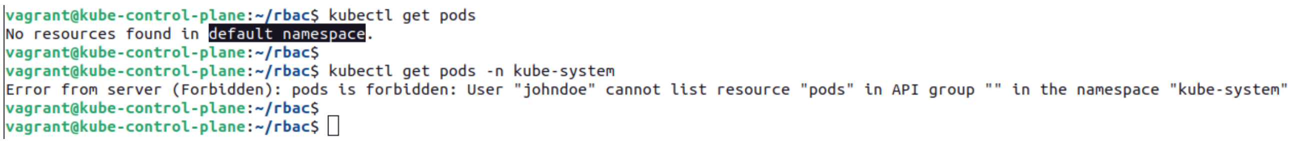
10. List the Pods in the namespace. What would you expect to happen?



```
vagrant@kube-control-plane:~/rbac$ kubectl run nginx --image=nginx --port=80
Error from server (Forbidden): pods is forbidden: User "johndoe" cannot create resource "pods" in API group "" in the namespace "default"
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$
```

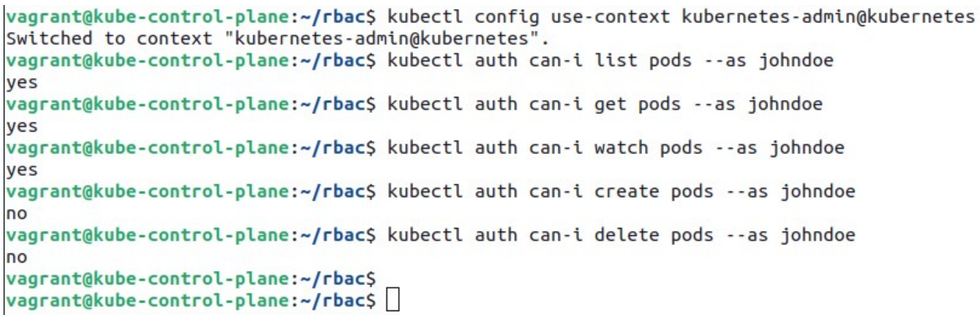
Creating a new Pod won't work as the user `johndoe` doesn't have the proper permissions.

11. List the Pods in the namespace. What would you expect to happen?



```
vagrant@kube-control-plane:~/rbac$ kubectl get pods
No resources found in default namespace.
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$ kubectl get pods -n kube-system
Error from server (Forbidden): pods is forbidden: User "johndoe" cannot list resource "pods" in API group "" in the namespace "kube-system"
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$
```

You can check the permissions of the `johndoe` user from the default context. As you can see, the operations `list`, `get`, and `watch` are permitted, whereas the `create` or `delete` operations are forbidden.



```
vagrant@kube-control-plane:~/rbac$ kubectl config use-context kubernetes-admin@kubernetes
Switched to context "kubernetes-admin@kubernetes".
vagrant@kube-control-plane:~/rbac$ kubectl auth can-i list pods --as johndoe
yes
vagrant@kube-control-plane:~/rbac$ kubectl auth can-i get pods --as johndoe
yes
vagrant@kube-control-plane:~/rbac$ kubectl auth can-i watch pods --as johndoe
yes
vagrant@kube-control-plane:~/rbac$ kubectl auth can-i create pods --as johndoe
no
vagrant@kube-control-plane:~/rbac$ kubectl auth can-i delete pods --as johndoe
no
vagrant@kube-control-plane:~/rbac$
vagrant@kube-control-plane:~/rbac$ 
```