# **Lab x: Container Security Context**

## **Defining security contexts with default user**

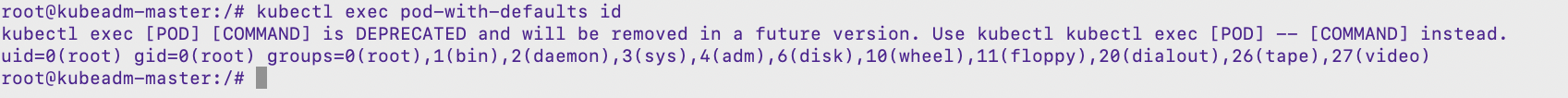
1. It allows you to lock down your containers, so that only certain processes can do certain things. This ensures the stability of your containers and allows you to give control or take it away. In this lesson, we’ll go through how to set the security context at the container level and the pod level.
2. Run an alpine container with default security

$ kubectl run pod-with-defaults --image alpine --restart Never -- /bin/sleep 999999



1. Check the ID on the container:

$ kubectl exec pod-with-defaults id



## **Defining security contexts with specific user**

1. The YAML for a container that runs as a user. View the file security-cxt.yaml

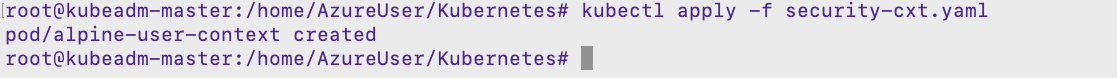
$ vim security-cxt.yaml

A picture containing bird

Description automatically generated

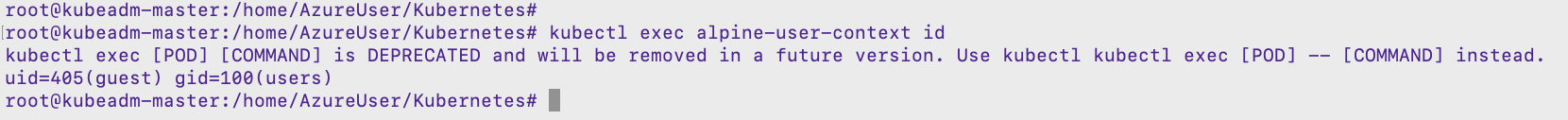
1. Create the resource from above yaml file

$ kubectl apply -f security-cxt.yaml



1. Check the user context

$ kubectl exec alpine-user-context id



## **Defining security contexts with non-root user**

1. The YAML for a pod that runs the container as non-root:

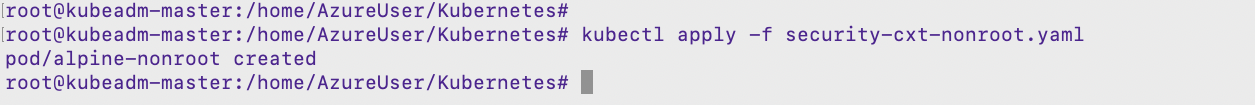
$ vim security-cxt-nonroot.yaml

A screenshot of a social media post

Description automatically generated

1. Create a pod that runs the container as non-root:

$ kubectl apply -f security-cxt-nonroot.yaml



1. View more information about the pod error:

$ kubectl describe pod alpine-nonroot

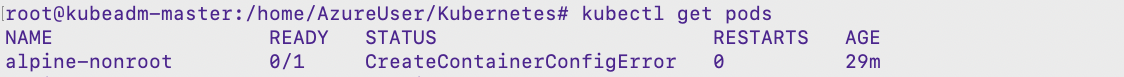
$ kubectl get pods

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated



## **Defining security contexts with privileged container pod**

1. The YAML for a privileged container pod:

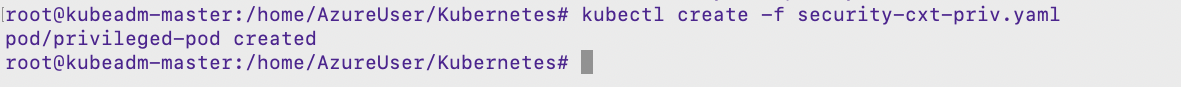
$ vim security-cxt-priv.yaml

A picture containing bird

Description automatically generated

1. Create the privileged container pod:

kubectl apply -f security-cxt-priv.yaml



1. View the devices on the default container:

$ kubectl exec -it pod-with-defaults ls /dev

A screenshot of a cell phone

Description automatically generated

1. View the devices on the privileged pod container:

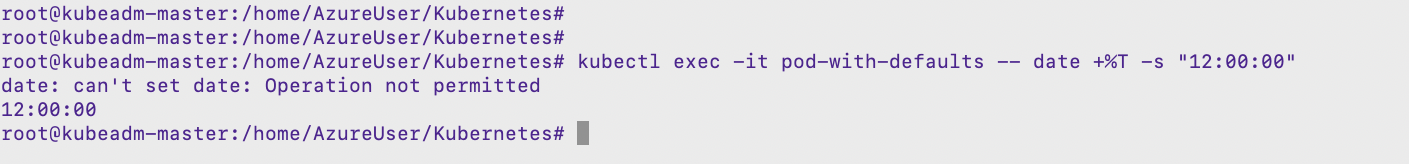
$ kubectl exec -it privileged-pod ls /dev

A screenshot of a social media post

Description automatically generated

1. Try to change the time on a default container pod:

$ kubectl exec -it pod-with-defaults -- date +%T -s "12:00:00"



## **Defining security contexts with privileged container pod – add capability**

1. The YAML for a container that will allow you to change the time:

$ vim security-cxt-time.yaml

A screenshot of a cell phone

Description automatically generated

1. Create the pod that will allow you to change the container’s time:

$ kubectl create -f security-cxt-time.yaml

1. Change the time on a container:

$ kubectl exec -it kernelchange-pod -- date +%T -s "12:00:00"



## **Defining security contexts with privileged container pod – remove capability**

1. The YAML for a container that removes capabilities:

$ vim security-cxt-rmcap.yaml

A screenshot of a social media post

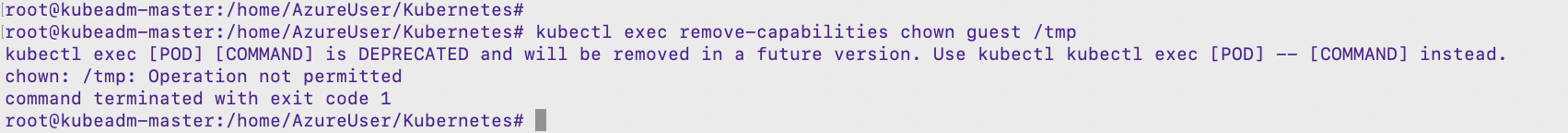
Description automatically generated

1. Create a pod that’s container has capabilities removed:

$ kubectl apply -f security-cxt-rmcap.yaml

1. Try to change the ownership of a container with removed capability:

$ kubectl exec remove-capabilities chown guest /tmp



## **Defining security contexts with privileged container pod – ReadOnly**

1. The YAML for a pod container that can’t write to the local filesystem:

$ vim security-cxt-readonly.yaml

1. Create a pod that will not allow you to write to the local container filesystem:

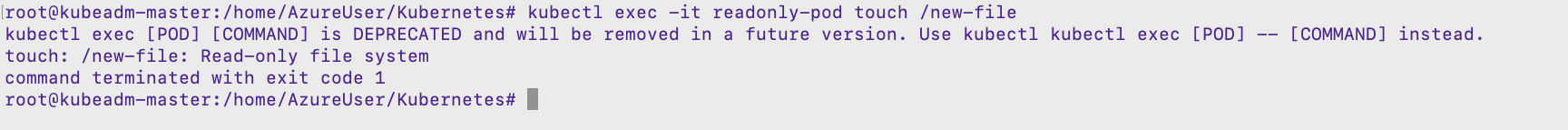
$ kubectl apply -f security-cxt-readonly.yaml

A screenshot of a cell phone

Description automatically generated

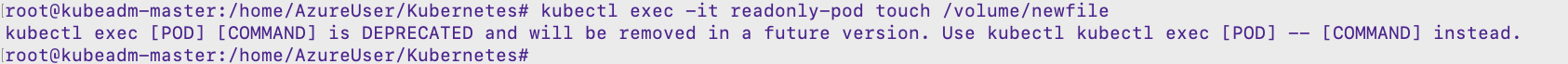
1. Try to write to the container filesystem:

$ kubectl exec -it readonly-pod touch /new-file



1. Create a file on the volume mounted to the container:

$ kubectl exec -it readonly-pod touch /volume/newfile



1. View the file on the volume that’s mounted:

$ kubectl exec -it readonly-pod -- ls -la /volume/newfile

