https://github.com/aquasecurity/trivy
https://kubernetes.io/docs/concepts/workloads/pods/

Namespace: r61

Identifying the container image of a Pod
Scanning a container image with Trivy

IMPORTANT: Trivy is installed as a command-line tool on the cluster node controlplane.

Identifying the Container Image of a Pod
The namespace r61 contains three different Pods. Ensure that all Pods transition into the Running status.
Identify the container images used by the Pod.

```
root@controlplane:~$ kubectl get ns
NAME              STATUS   AGE
default           Active   7m13s
kube-flannel      Active   6m57s
kube-node-lease   Active   7m14s
kube-public       Active   7m14s
kube-system       Active   7m14s
r61               Active   6m57s
root@controlplane:~$ kubectl get pods -n r61
NAME       READY   STATUS    RESTARTS   AGE
backend    1/1     Running   0          7m33s
logstash   1/1     Running   0          7m33s
loop       1/1     Running   0          7m33s
```

Check the images of each Pod in the namespace r61 using the kubectl describe command. The used images are bmuschko/nodejs-hello-world:1.0.0, alpine:3.13.4, and elastic/logstash:7.13.3.

```
root@controlplane:~$ kubectl describe pod backend -n r61
Name:             backend
Namespace:        r61
Priority:         0
Service Account:  default
Node:             node01/172.17.252.6
Start Time:       Mon, 29 May 2023 05:01:24 +0000
Labels:           tier=backend
Annotations:      <none>
Status:           Running
IP:               10.244.1.4
IPs:
  IP:  10.244.1.4
Containers:
  hello:
    Container ID:   containerd://b308a5cc96ad13c70774e0410
    Image:          bmuschko/nodejs-hello-world:1.0.0
```

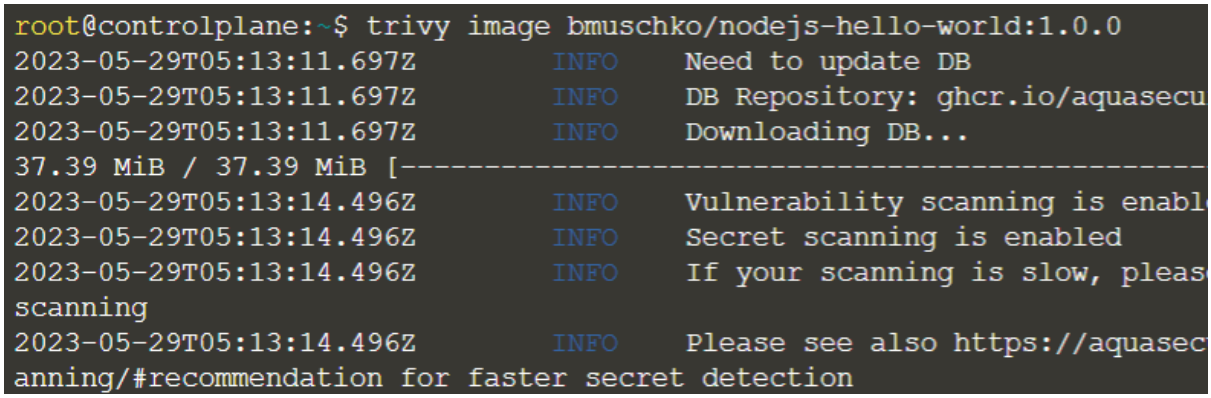
```
root@controlplane:~$ kubectl describe pod logstash -n r61
Name:             logstash
Namespace:        r61
Priority:         0
Service Account:  default
Node:             node01/172.17.252.6
Start Time:       Mon, 29 May 2023 05:01:24 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.1.2
IPs:
  IP:  10.244.1.2
Containers:
  logstash:
    Container ID:   containerd://b6afa4b897a3b149ff5776e8c
    Image:          elastic/logstash:7.13.3
```

```
root@controlplane:~$ kubectl describe pod loop -n r61
Name:             loop
Namespace:        r61
Priority:         0
Service Account:  default
Node:             node01/172.17.252.6
Start Time:       Mon, 29 May 2023 05:01:24 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.1.3
IPs:
  IP:  10.244.1.3
Containers:
  loop:
    Container ID:  containerd://efc5467751d703996936df
    Image:         alpine:3.13.4
```

The container will expose the used container image.

Scanning a Container Image with Trivy
From the command line, execute Trivy against the container images used by the Pods. Delete all Pods that have CRITICAL vulnerabilities. Which of the Pods are still running?

Use the Trivy executable to check vulnerabilities for all images:



$ trivy image bmuschko/nodejs-hello-world:1.0.0
$ trivy image alpine:3.13.4
$ trivy image elastic/logstash:7.13.3

If you look closely at the list of vulnerabilities, you will find that all images contain issues with CRITICAL severity. As a result, delete all Pods. Use the --force to avoid having to wait for a graceful deletion of the Pod:

$ kubectl delete pod backend -n r61 --force
$ kubectl delete pod logstash -n r61 --force
$ kubectl delete pod loop -n r61 --force
You should end up with zero Pods in the namespace r61:

$ kubectl get pods -n r61
No resources found in r61 namespace.