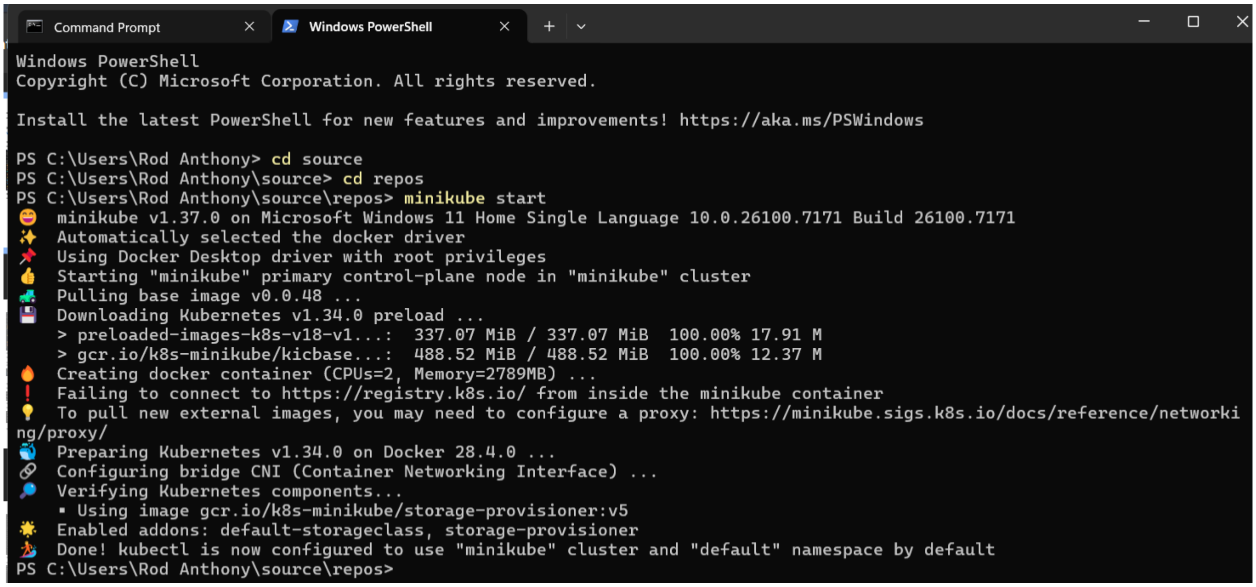
Rod Anthony C. Balaoro
IV - ACSAD

ELECTIVE 3
Assignment#5

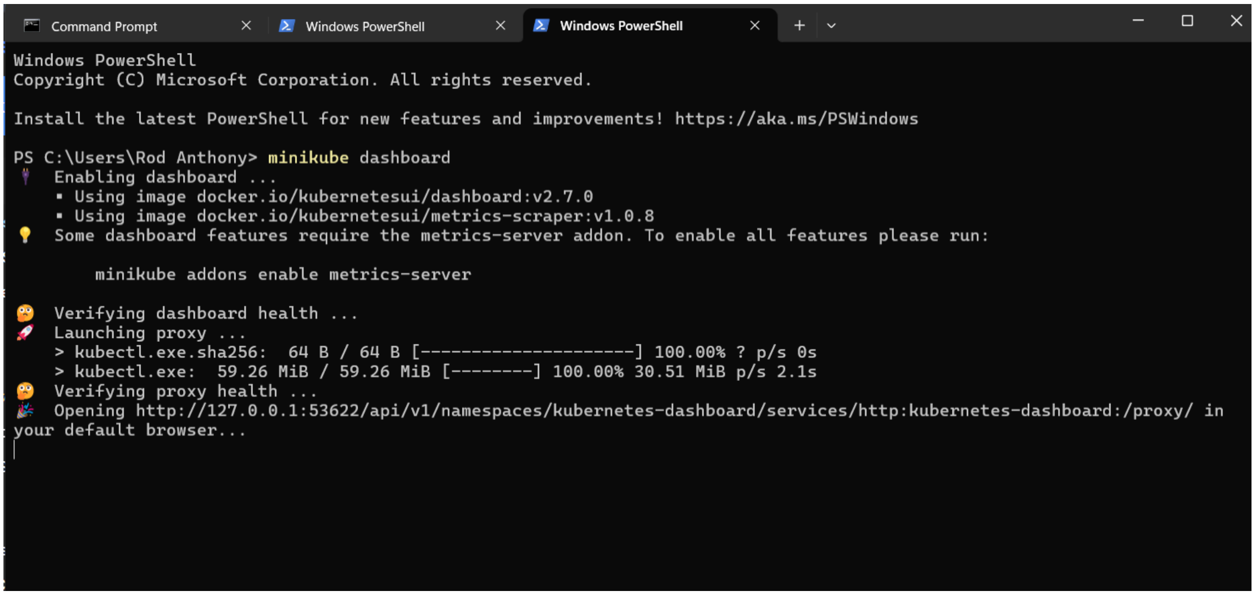
Hello Minikube:
https://kubernetes.io/docs/tutorials/hello-minikube

STEP 1 : Create a minikube cluster using terminal minikube start



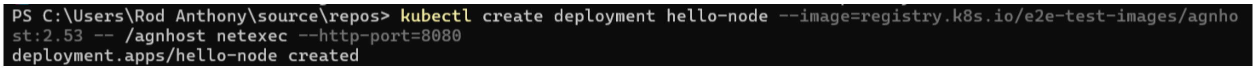
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Rod Anthony> cd source
PS C:\Users\Rod Anthony\source> cd repos
PS C:\Users\Rod Anthony\source\repos> minikube start
😄  minikube v1.37.0 on Microsoft Windows 11 Home Single Language 10.0.26100.7171 Build 26100.7171
✨  Automatically selected the docker driver
📌  Using Docker Desktop driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.48 ...
💾  Downloading Kubernetes v1.34.0 preload ...
    > preloaded-images-k8s-v18-v1...:  337.07 MiB / 337.07 MiB  100.00% 17.91 M
    > gcr.io/k8s-minikube/kicbase...:  488.52 MiB / 488.52 MiB  100.00% 12.37 M
🔥  Creating docker container (CPUs=2, Memory=2789MB) ...
❗  Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡  To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networki
ng/proxy/
🐳  Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\Rod Anthony\source\repos>
```

STEP 2: Open another terminal minikube dashboard



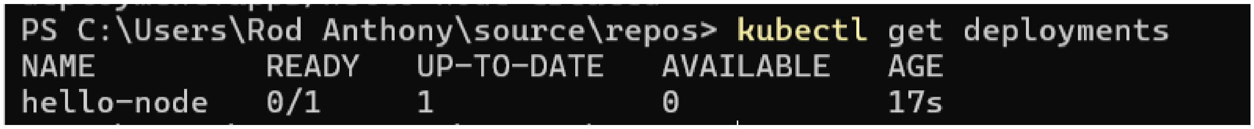
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Rod Anthony> minikube dashboard
🔌  Enabling dashboard ...
    ▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
    ▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
💡  Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server

🤔  Verifying dashboard health ...
🚀  Launching proxy ...
    > kubectl.exe.sha256:  64 B / 64 B [----------------------] 100.00% ? p/s 0s
    > kubectl.exe:  59.26 MiB / 59.26 MiB [--------] 100.00% 30.51 MiB p/s 2.1s
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:53622/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in
your default browser...
```

CREATE A DEPLOYMENT
STEP 3: Create a Deployment
kubectl create deployment hello-node --image=registry.k8s.io/e2e-test- images/agnhost:2.53 -- /agnhost netexec --http-port=8080



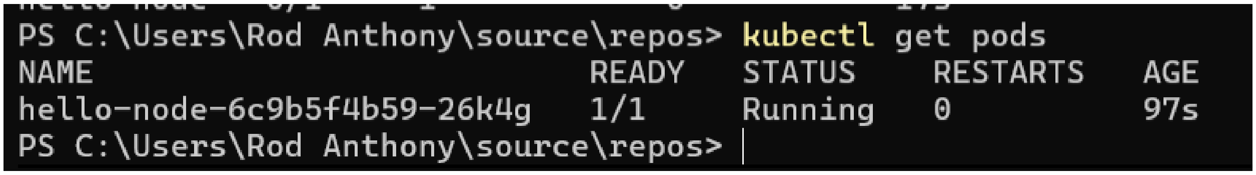
```
PS C:\Users\Rod Anthony\source\repos> kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnho
st:2.53 -- /agnhost netexec --http-port=8080
deployment.apps/hello-node created
```

STEP 4: View the deployment kubectl get deployments



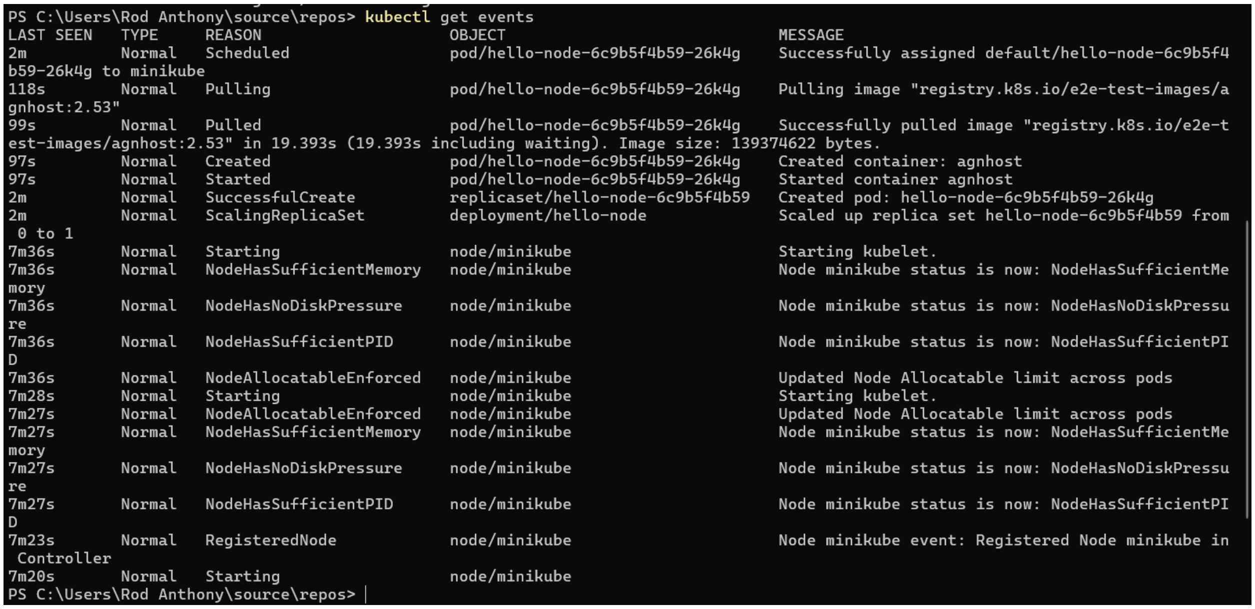
```
PS C:\Users\Rod Anthony\source\repos> kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   0/1     1            0           17s
```

STEP 5: View the pod kubectl get pods



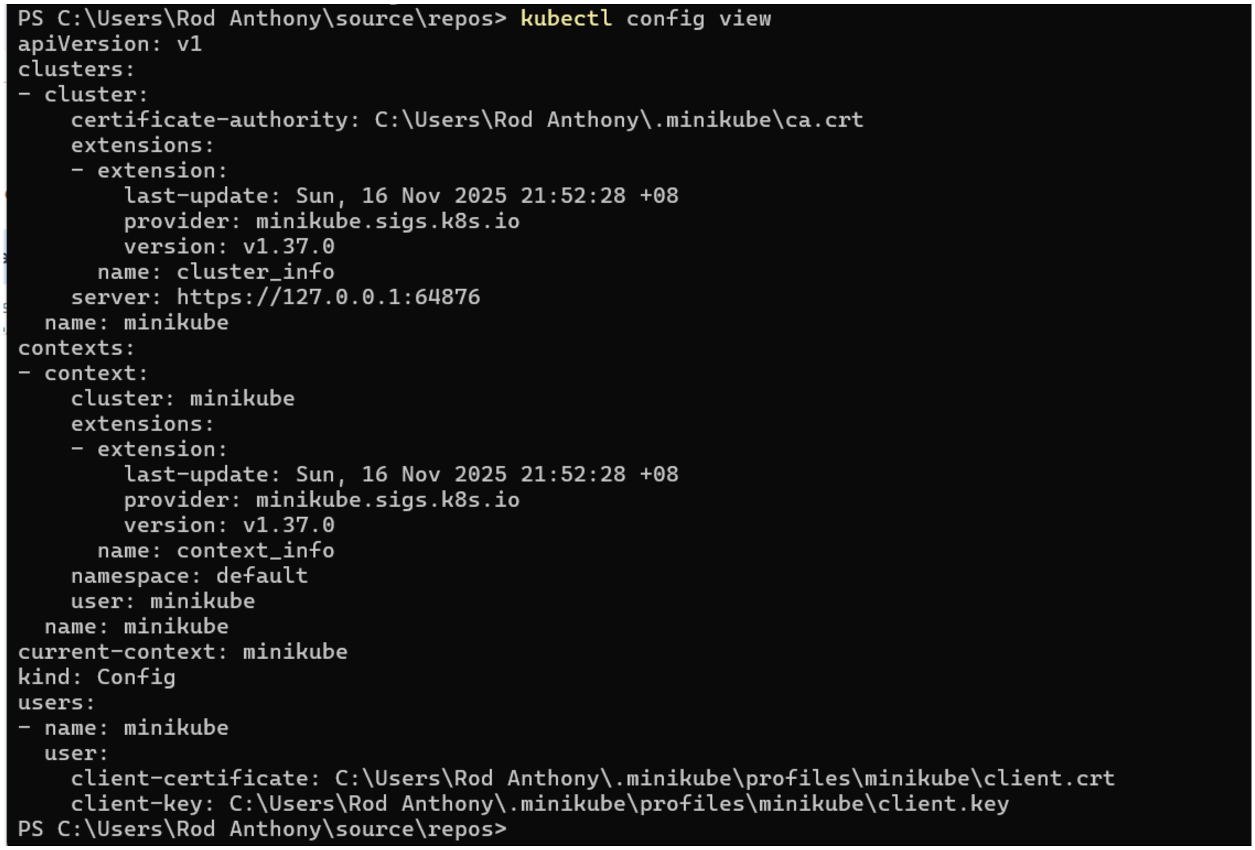
```
PS C:\Users\Rod Anthony\source\repos> kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-6c9b5f4b59-26k4g   1/1     Running   0          97s
PS C:\Users\Rod Anthony\source\repos> 
```

STEP 6: View cluster events



```
PS C:\Users\Rod Anthony\source\repos> kubectl get events
LAST SEEN   TYPE     REASON                    OBJECT                             MESSAGE
2m          Normal   Scheduled                 pod/hello-node-6c9b5f4b59-26k4g    Successfully assigned default/hello-node-6c9b5f4
b59-26k4g to minikube
118s        Normal   Pulling                   pod/hello-node-6c9b5f4b59-26k4g    Pulling image "registry.k8s.io/e2e-test-images/a
gnhost:2.53"
99s         Normal   Pulled                    pod/hello-node-6c9b5f4b59-26k4g    Successfully pulled image "registry.k8s.io/e2e-t
est-images/agnhost:2.53" in 19.393s (19.393s including waiting). Image size: 139374622 bytes.
97s         Normal   Created                   pod/hello-node-6c9b5f4b59-26k4g    Created container: agnhost
97s         Normal   Started                   pod/hello-node-6c9b5f4b59-26k4g    Started container agnhost
2m          Normal   SuccessfulCreate          replicaset/hello-node-6c9b5f4b59   Created pod: hello-node-6c9b5f4b59-26k4g
2m          Normal   ScalingReplicaSet         deployment/hello-node              Scaled up replica set hello-node-6c9b5f4b59 from
 0 to 1
7m36s       Normal   Starting                  node/minikube                      Starting kubelet.
7m36s       Normal   NodeHasSufficientMemory   node/minikube                      Node minikube status is now: NodeHasSufficientMe
mory
7m36s       Normal   NodeHasNoDiskPressure     node/minikube                      Node minikube status is now: NodeHasNoDiskPressu
re
7m36s       Normal   NodeHasSufficientPID      node/minikube                      Node minikube status is now: NodeHasSufficientPI
D
7m36s       Normal   NodeAllocatableEnforced   node/minikube                      Updated Node Allocatable limit across pods
7m28s       Normal   Starting                  node/minikube                      Starting kubelet.
7m27s       Normal   NodeAllocatableEnforced   node/minikube                      Updated Node Allocatable limit across pods
7m27s       Normal   NodeHasSufficientMemory   node/minikube                      Node minikube status is now: NodeHasSufficientMe
mory
7m27s       Normal   NodeHasNoDiskPressure     node/minikube                      Node minikube status is now: NodeHasNoDiskPressu
re
7m27s       Normal   NodeHasSufficientPID      node/minikube                      Node minikube status is now: NodeHasSufficientPI
D
7m23s       Normal   RegisteredNode            node/minikube                      Node minikube event: Registered Node minikube in
 Controller
7m20s       Normal   Starting                  node/minikube
PS C:\Users\Rod Anthony\source\repos> 
```

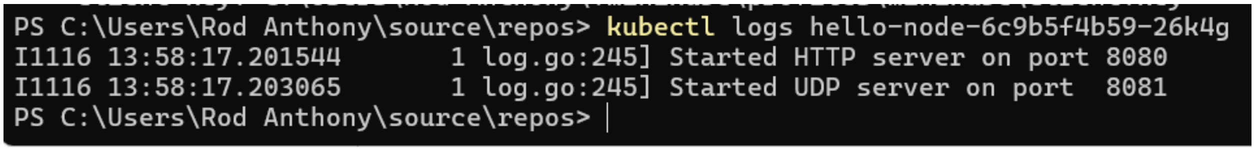
STEP 7: View the kubectl configuration: kubectl config view



```
PS C:\Users\Rod Anthony\source\repos> kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: C:\Users\Rod Anthony\.minikube\ca.crt
    extensions:
    - extension:
        last-update: Sun, 16 Nov 2025 21:52:28 +08
        provider: minikube.sigs.k8s.io
        version: v1.37.0
      name: cluster_info
    server: https://127.0.0.1:64876
  name: minikube
contexts:
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Sun, 16 Nov 2025 21:52:28 +08
        provider: minikube.sigs.k8s.io
        version: v1.37.0
      name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
users:
- name: minikube
  user:
    client-certificate: C:\Users\Rod Anthony\.minikube\profiles\minikube\client.crt
    client-key: C:\Users\Rod Anthony\.minikube\profiles\minikube\client.key
PS C:\Users\Rod Anthony\source\repos>
```

STEP 8: View application logs for a container in a pod (replace pod name with the one you got from kubectl get pods).

kubectl logs hello-node-6c9b5f4b59-b7sfd

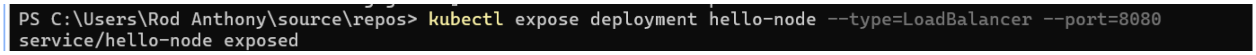


```
PS C:\Users\Rod Anthony\source\repos> kubectl logs hello-node-6c9b5f4b59-26k4g
I1116 13:58:17.201544       1 log.go:245] Started HTTP server on port 8080
I1116 13:58:17.203065       1 log.go:245] Started UDP server on port  8081
PS C:\Users\Rod Anthony\source\repos>
```

CREATE A SERVICE

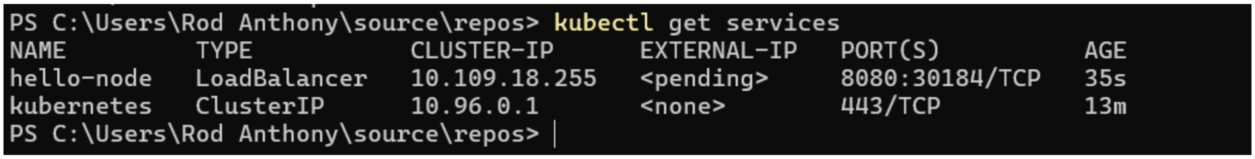
STEP 9: Expose the Pod to the public internet using the kubectl expose command:

kubectl expose deployment hello-node --type=LoadBalancer --port=8080



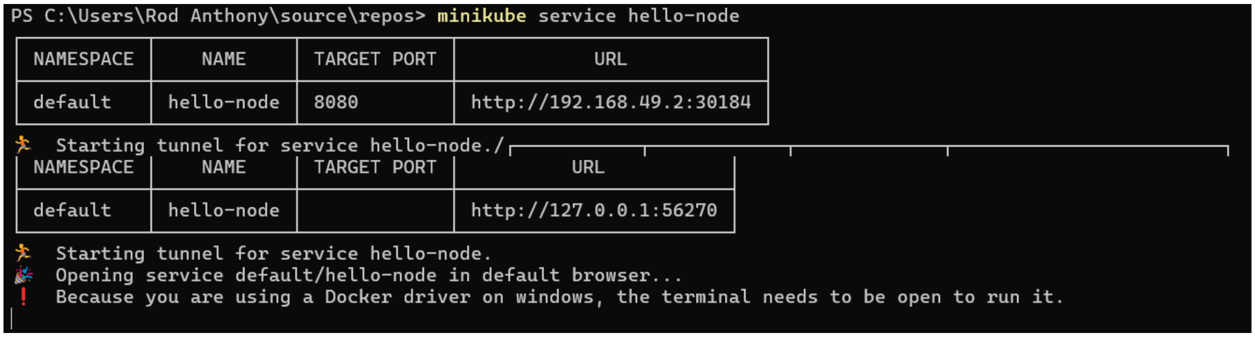
```
PS C:\Users\Rod Anthony\source\repos> kubectl expose deployment hello-node --type=LoadBalancer --port=8080
service/hello-node exposed
```

STEP 10: View the Service you created:

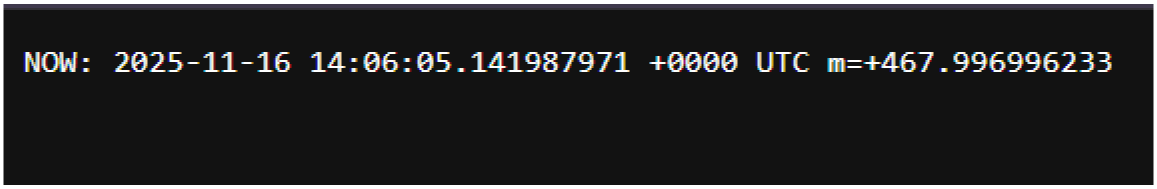


```
PS C:\Users\Rod Anthony\source\repos> kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-node   LoadBalancer   10.109.18.255   <pending>     8080:30184/TCP   35s
kubernetes   ClusterIP      10.96.0.1       <none>        443/TCP          13m
PS C:\Users\Rod Anthony\source\repos>
```

STEP 11: minikube service minikube service hello-node



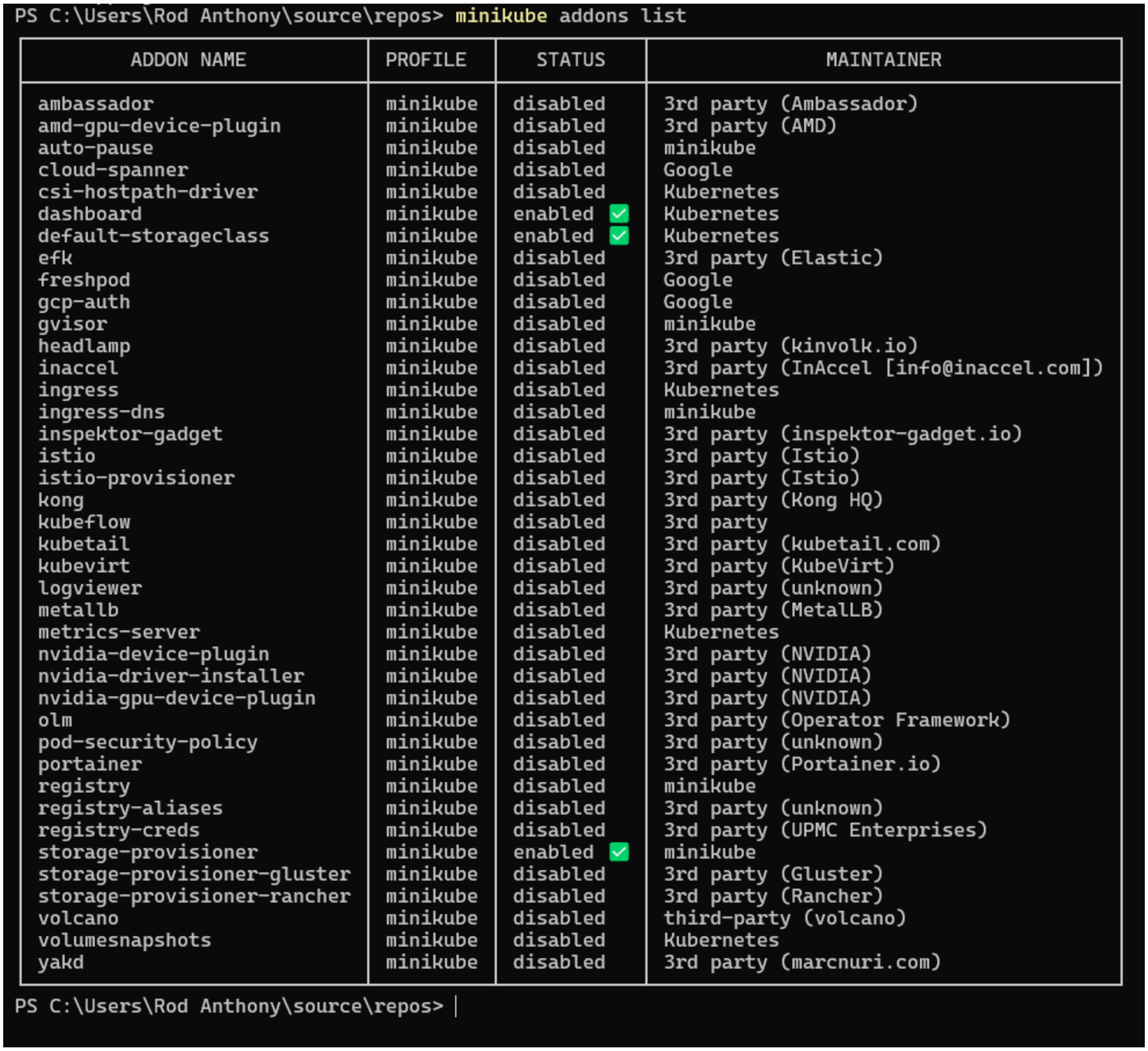
```
PS C:\Users\Rod Anthony\source\repos> minikube service hello-node
┌───────────┬────────────┬─────────────┬───────────────────────────┐
│ NAMESPACE │    NAME    │ TARGET PORT │            URL            │
├───────────┼────────────┼─────────────┼───────────────────────────┤
│ default   │ hello-node │ 8080        │ http://192.168.49.2:30184 │
└───────────┴────────────┴─────────────┴───────────────────────────┘
🏃  Starting tunnel for service hello-node./┌───────────┬────────────┬─────────────┬────────────────────────┐
│ NAMESPACE │    NAME    │ TARGET PORT │          URL           │
├───────────┼────────────┼─────────────┼────────────────────────┤
│ default   │ hello-node │             │ http://127.0.0.1:56270 │
└───────────┴────────────┴─────────────┴────────────────────────┘
🏃  Starting tunnel for service hello-node.
🎉  Opening service default/hello-node in default browser...
❗  Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```



```
NOW: 2025-11-16 14:06:05.141987971 +0000 UTC m=+467.996996233
```

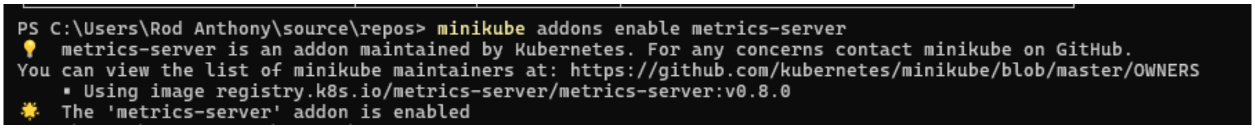
ENABLE ADD ONS

STEP 12: List the currently supported addons: minikube addons list



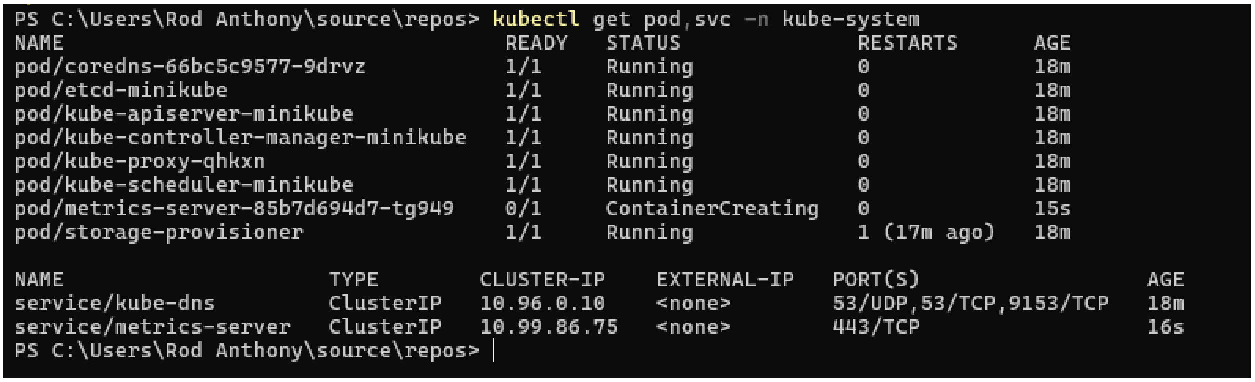
```
PS C:\Users\Rod Anthony\source\repos> minikube addons list
|-----------------------------|----------|------------|----------------------------------------|
|         ADDON NAME          | PROFILE  |   STATUS   |               MAINTAINER               |
|-----------------------------|----------|------------|----------------------------------------|
| ambassador                  | minikube | disabled   | 3rd party (Ambassador)                 |
| amd-gpu-device-plugin       | minikube | disabled   | 3rd party (AMD)                        |
| auto-pause                  | minikube | disabled   | minikube                               |
| cloud-spanner               | minikube | disabled   | Google                                 |
| csi-hostpath-driver         | minikube | disabled   | Kubernetes                             |
| dashboard                   | minikube | enabled ✅ | Kubernetes                             |
| default-storageclass        | minikube | enabled ✅ | Kubernetes                             |
| efk                         | minikube | disabled   | 3rd party (Elastic)                    |
| freshpod                    | minikube | disabled   | Google                                 |
| gcp-auth                    | minikube | disabled   | Google                                 |
| gvisor                      | minikube | disabled   | minikube                               |
| headlamp                    | minikube | disabled   | 3rd party (kinvolk.io)                 |
| inaccel                     | minikube | disabled   | 3rd party (InAccel [info@inaccel.com]) |
| ingress                     | minikube | disabled   | Kubernetes                             |
| ingress-dns                 | minikube | disabled   | minikube                               |
| inspektor-gadget            | minikube | disabled   | 3rd party (inspektor-gadget.io)        |
| istio                       | minikube | disabled   | 3rd party (Istio)                      |
| istio-provisioner           | minikube | disabled   | 3rd party (Istio)                      |
| kong                        | minikube | disabled   | 3rd party (Kong HQ)                    |
| kubeflow                    | minikube | disabled   | 3rd party                              |
| kubetail                    | minikube | disabled   | 3rd party (kubetail.com)               |
| kubevirt                    | minikube | disabled   | 3rd party (KubeVirt)                   |
| logviewer                   | minikube | disabled   | 3rd party (unknown)                    |
| metallb                     | minikube | disabled   | 3rd party (MetalLB)                    |
| metrics-server              | minikube | disabled   | Kubernetes                             |
| nvidia-device-plugin        | minikube | disabled   | 3rd party (NVIDIA)                     |
| nvidia-driver-installer     | minikube | disabled   | 3rd party (NVIDIA)                     |
| nvidia-gpu-device-plugin    | minikube | disabled   | 3rd party (NVIDIA)                     |
| olm                         | minikube | disabled   | 3rd party (Operator Framework)         |
| pod-security-policy         | minikube | disabled   | 3rd party (unknown)                    |
| portainer                   | minikube | disabled   | 3rd party (Portainer.io)               |
| registry                    | minikube | disabled   | minikube                               |
| registry-aliases            | minikube | disabled   | 3rd party (unknown)                    |
| registry-creds              | minikube | disabled   | 3rd party (UPMC Enterprises)           |
| storage-provisioner         | minikube | enabled ✅ | minikube                               |
| storage-provisioner-gluster | minikube | disabled   | 3rd party (Gluster)                    |
| storage-provisioner-rancher | minikube | disabled   | 3rd party (Rancher)                    |
| volcano                     | minikube | disabled   | third-party (volcano)                  |
| volumesnapshots             | minikube | disabled   | Kubernetes                             |
| yakd                        | minikube | disabled   | 3rd party (marcnuri.com)               |
|-----------------------------|----------|------------|----------------------------------------|
PS C:\Users\Rod Anthony\source\repos> |
```

STEP 13: Enable an addon, for example, metrics-server: minikube addons enable metrics-server



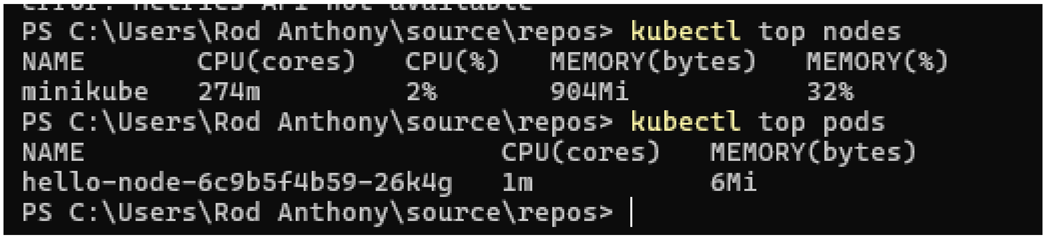
```
PS C:\Users\Rod Anthony\source\repos> minikube addons enable metrics-server
💡  metrics-server is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
    ▪ Using image registry.k8s.io/metrics-server/metrics-server:v0.8.0
🌟  The 'metrics-server' addon is enabled
```

STEP 14: View the Pod and Service you created by installing that addon:



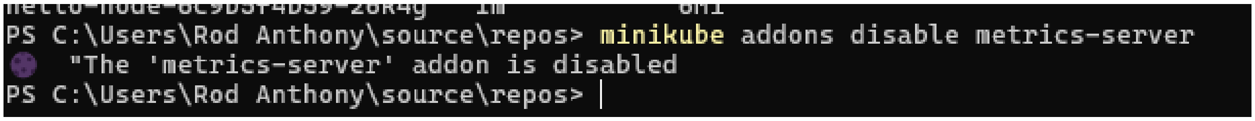
```
PS C:\Users\Rod Anthony\source\repos> kubectl get pod,svc -n kube-system
NAME                                   READY   STATUS              RESTARTS      AGE
pod/coredns-66bc5c9577-9drvz           1/1     Running             0             18m
pod/etcd-minikube                      1/1     Running             0             18m
pod/kube-apiserver-minikube            1/1     Running             0             18m
pod/kube-controller-manager-minikube   1/1     Running             0             18m
pod/kube-proxy-qhkxn                   1/1     Running             0             18m
pod/kube-scheduler-minikube            1/1     Running             0             18m
pod/metrics-server-85b7d694d7-tg949    0/1     ContainerCreating   0             15s
pod/storage-provisioner                1/1     Running             1 (17m ago)   18m

NAME                        TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)                  AGE
service/kube-dns            ClusterIP   10.96.0.10     <none>        53/UDP,53/TCP,9153/TCP   18m
service/metrics-server      ClusterIP   10.99.86.75    <none>        443/TCP                  16s
PS C:\Users\Rod Anthony\source\repos> |
```

STEP 14: Check the output from metrics-server: kubectl top pods



```
error: Metrics API not available
PS C:\Users\Rod Anthony\source\repos> kubectl top nodes
NAME       CPU(cores)   CPU(%)   MEMORY(bytes)   MEMORY(%)
minikube   274m         2%       904Mi           32%
PS C:\Users\Rod Anthony\source\repos> kubectl top pods
NAME                          CPU(cores)   MEMORY(bytes)
hello-node-6c9b5f4b59-26k4g   1m           6Mi
PS C:\Users\Rod Anthony\source\repos> |
```

STEP 15: Disable metrics-server: minikube addons disable metrics-server

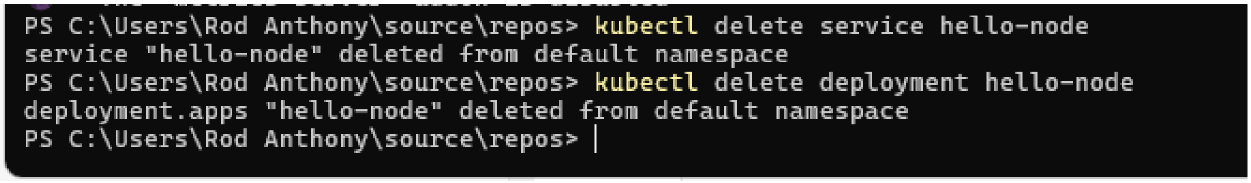


```
hello-node-6c9b5f4b59-26k4g   1m           6Mi
PS C:\Users\Rod Anthony\source\repos> minikube addons disable metrics-server
●  "The 'metrics-server' addon is disabled
PS C:\Users\Rod Anthony\source\repos> |
```

CLEAN UP

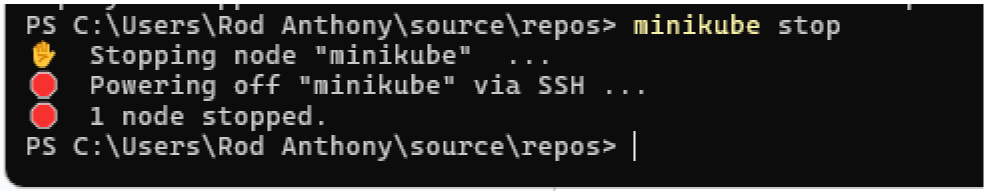
STEP 16: Cleaning up resources in the cluster kubectl delete service hello-node
kubectl delete deployment hello-node

STEP 17: Stop the Minikube cluster



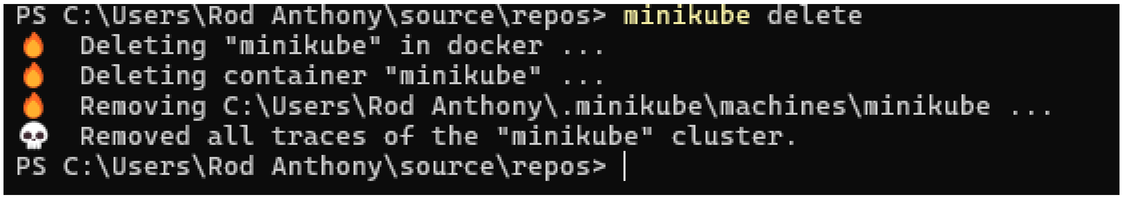
```
PS C:\Users\Rod Anthony\source\repos> kubectl delete service hello-node
service "hello-node" deleted from default namespace
PS C:\Users\Rod Anthony\source\repos> kubectl delete deployment hello-node
deployment.apps "hello-node" deleted from default namespace
PS C:\Users\Rod Anthony\source\repos> |
```

minikube stop



```
PS C:\Users\Rod Anthony\source\repos> minikube stop
✋  Stopping node "minikube"  ...
🛑  Powering off "minikube" via SSH ...
🛑  1 node stopped.
PS C:\Users\Rod Anthony\source\repos> |
```

STEP 18: delete the Minikube VM minikube delete

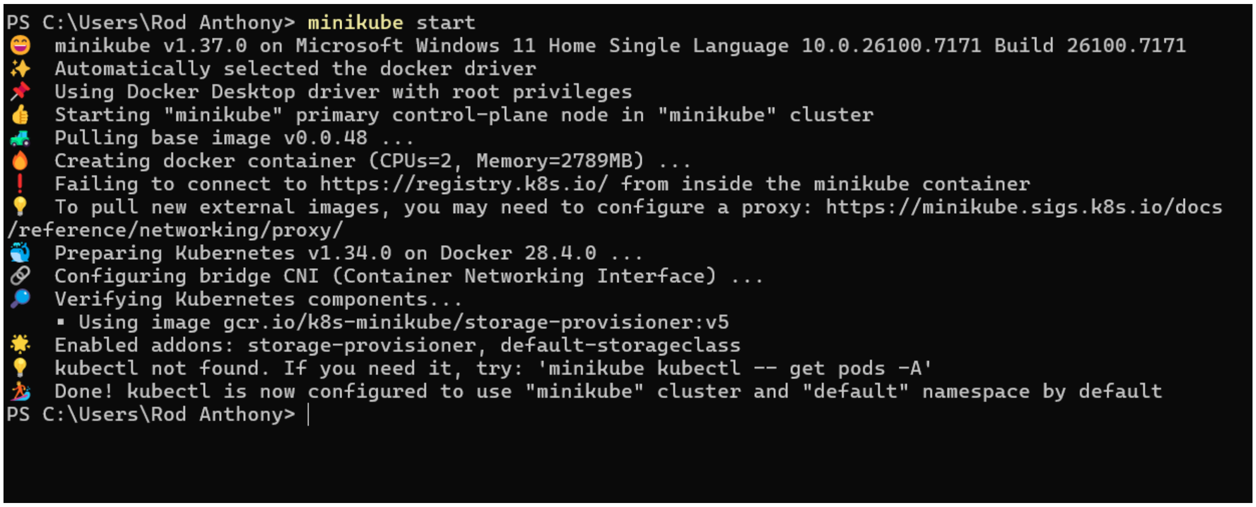


```
PS C:\Users\Rod Anthony\source\repos> minikube delete
🔥  Deleting "minikube" in docker ...
🔥  Deleting container "minikube" ...
🔥  Removing C:\Users\Rod Anthony\.minikube\machines\minikube ...
💀  Removed all traces of the "minikube" cluster.
PS C:\Users\Rod Anthony\source\repos> |
```

GET A SHELL TO A RUNNING CONTAINER

https://kubernetes.io/docs/tasks/debug/debug-application/get-shell-running- container

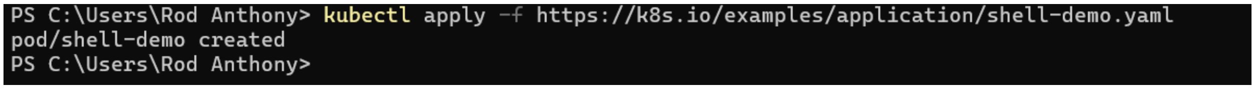
STEP 1: start minikube cluster



```
PS C:\Users\Rod Anthony> minikube start
😄  minikube v1.37.0 on Microsoft Windows 11 Home Single Language 10.0.26100.7171 Build 26100.7171
✨  Automatically selected the docker driver
📌  Using Docker Desktop driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.48 ...
🔥  Creating docker container (CPUs=2, Memory=2789MB) ...
❗  Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡  To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs
/reference/networking/proxy/
🐳  Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
💡  kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\Rod Anthony>
```

STEP 2: CREATE THE POD

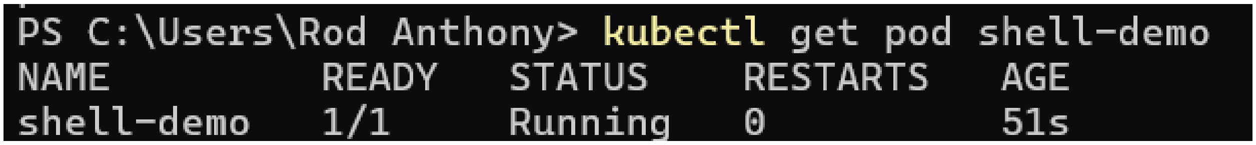
kubectl apply -f https://k8s.io/examples/application/shell-demo.yaml



```
PS C:\Users\Rod Anthony> kubectl apply -f https://k8s.io/examples/application/shell-demo.yaml
pod/shell-demo created
PS C:\Users\Rod Anthony>
```

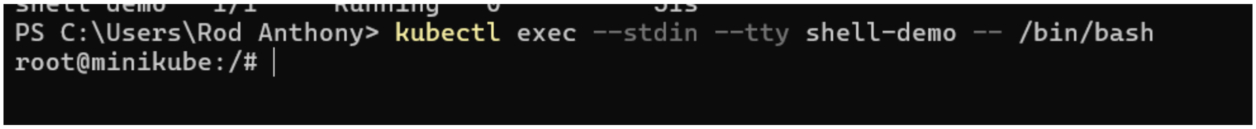
STEP 3: VERIFY THE CONTAINER IS RUNNING

kubectl get pod shell-demo



```
PS C:\Users\Rod Anthony> kubectl get pod shell-demo
NAME         READY   STATUS    RESTARTS   AGE
shell-demo   1/1     Running   0          51s
```

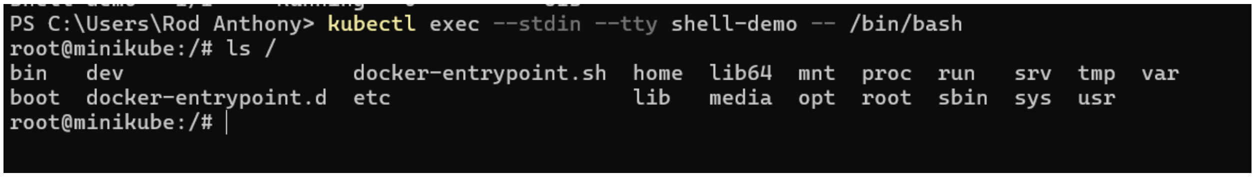
STEP 4: Get a shell to the running container: kubectl exec --stdin --tty shell-demo -- /bin/bash



```
PS C:\Users\Rod Anthony> kubectl exec --stdin --tty shell-demo -- /bin/bash
root@minikube:/#
```

STEP 5: In your shell, list the root directory:

ls /



```
PS C:\Users\Rod Anthony> kubectl exec --stdin --tty shell-demo -- /bin/bash
root@minikube:/# ls /
bin   dev                  docker-entrypoint.sh  home  lib64  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   media  opt  root  sbin  sys  usr
root@minikube:/#
```

STEP 6: experimenting with commands (run any to experiment) cat /proc/mounts

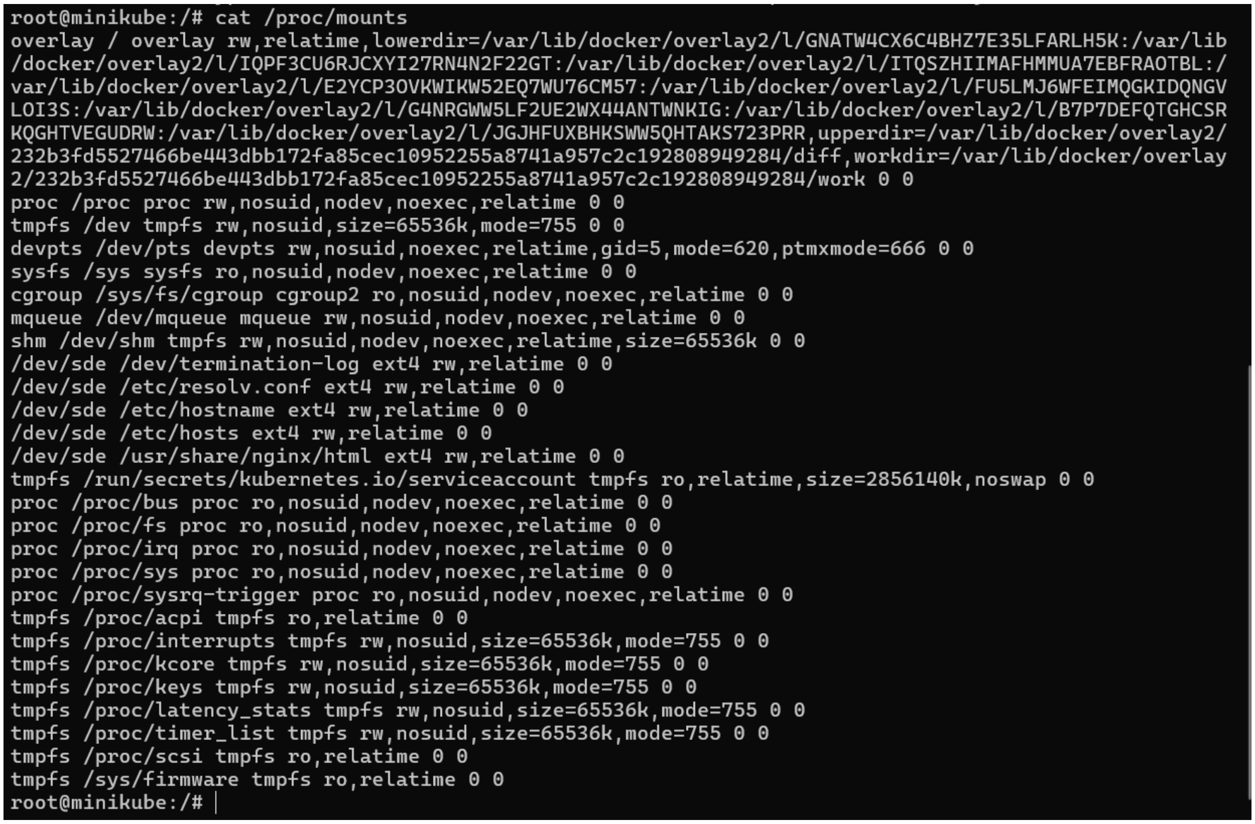
cat /proc/1/maps apt-get update

apt-get install -y tcpdump tcpdump

apt-get install -y lsof lsof

apt-get install -y procps ps aux

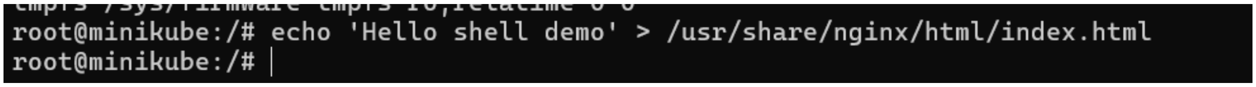
ps aux | grep nginx



```
root@minikube:/# cat /proc/mounts
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/GNATW4CX6C4BHZ7E35LFARLH5K:/var/lib
/docker/overlay2/l/IQPF3CU6RJCXYI27RN4N2F22GT:/var/lib/docker/overlay2/l/ITQSZHIIMAFHMMUA7EBFRAOTBL:/
var/lib/docker/overlay2/l/E2YCP3OVKWIKW52EQ7WU76CM57:/var/lib/docker/overlay2/l/FU5LMJ6WFEIMQGKIDQNGV
LOI3S:/var/lib/docker/overlay2/l/G4NRGWW5LF2UE2WX44ANTWNKIG:/var/lib/docker/overlay2/l/B7P7DEFQTGHCSR
KQGHTVEGUDRW:/var/lib/docker/overlay2/l/JGJHFUXBHKSWW5QHTAKS723PRR,upperdir=/var/lib/docker/overlay2/
232b3fd5527466be443dbb172fa85cec10952255a8741a957c2c192808949284/diff,workdir=/var/lib/docker/overlay
2/232b3fd5527466be443dbb172fa85cec10952255a8741a957c2c192808949284/work 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev tmpfs rw,nosuid,size=65536k,mode=755 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666 0 0
sysfs /sys sysfs ro,nosuid,nodev,noexec,relatime 0 0
cgroup /sys/fs/cgroup cgroup2 ro,nosuid,nodev,noexec,relatime 0 0
mqueue /dev/mqueue mqueue rw,nosuid,nodev,noexec,relatime 0 0
shm /dev/shm tmpfs rw,nosuid,nodev,noexec,relatime,size=65536k 0 0
/dev/sde /dev/termination-log ext4 rw,relatime 0 0
/dev/sde /etc/resolv.conf ext4 rw,relatime 0 0
/dev/sde /etc/hostname ext4 rw,relatime 0 0
/dev/sde /etc/hosts ext4 rw,relatime 0 0
/dev/sde /usr/share/nginx/html ext4 rw,relatime 0 0
tmpfs /run/secrets/kubernetes.io/serviceaccount tmpfs ro,relatime,size=2856140k,noswap 0 0
proc /proc/bus proc ro,nosuid,nodev,noexec,relatime 0 0
proc /proc/fs proc ro,nosuid,nodev,noexec,relatime 0 0
proc /proc/irq proc ro,nosuid,nodev,noexec,relatime 0 0
proc /proc/sys proc ro,nosuid,nodev,noexec,relatime 0 0
proc /proc/sysrq-trigger proc ro,nosuid,nodev,noexec,relatime 0 0
tmpfs /proc/acpi tmpfs ro,relatime 0 0
tmpfs /proc/interrupts tmpfs rw,nosuid,size=65536k,mode=755 0 0
tmpfs /proc/kcore tmpfs rw,nosuid,size=65536k,mode=755 0 0
tmpfs /proc/keys tmpfs rw,nosuid,size=65536k,mode=755 0 0
tmpfs /proc/latency_stats tmpfs rw,nosuid,size=65536k,mode=755 0 0
tmpfs /proc/timer_list tmpfs rw,nosuid,size=65536k,mode=755 0 0
tmpfs /proc/scsi tmpfs ro,relatime 0 0
tmpfs /sys/firmware tmpfs ro,relatime 0 0
root@minikube:/# |
```

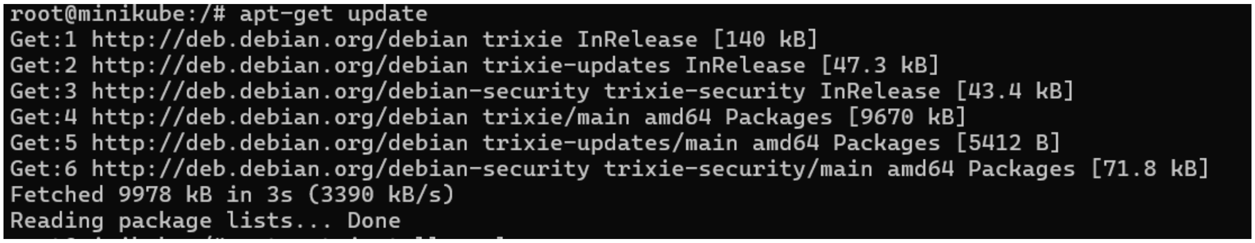
WRITING THE ROOT PAGE FOR NGINX

STEP 7: create an index.html



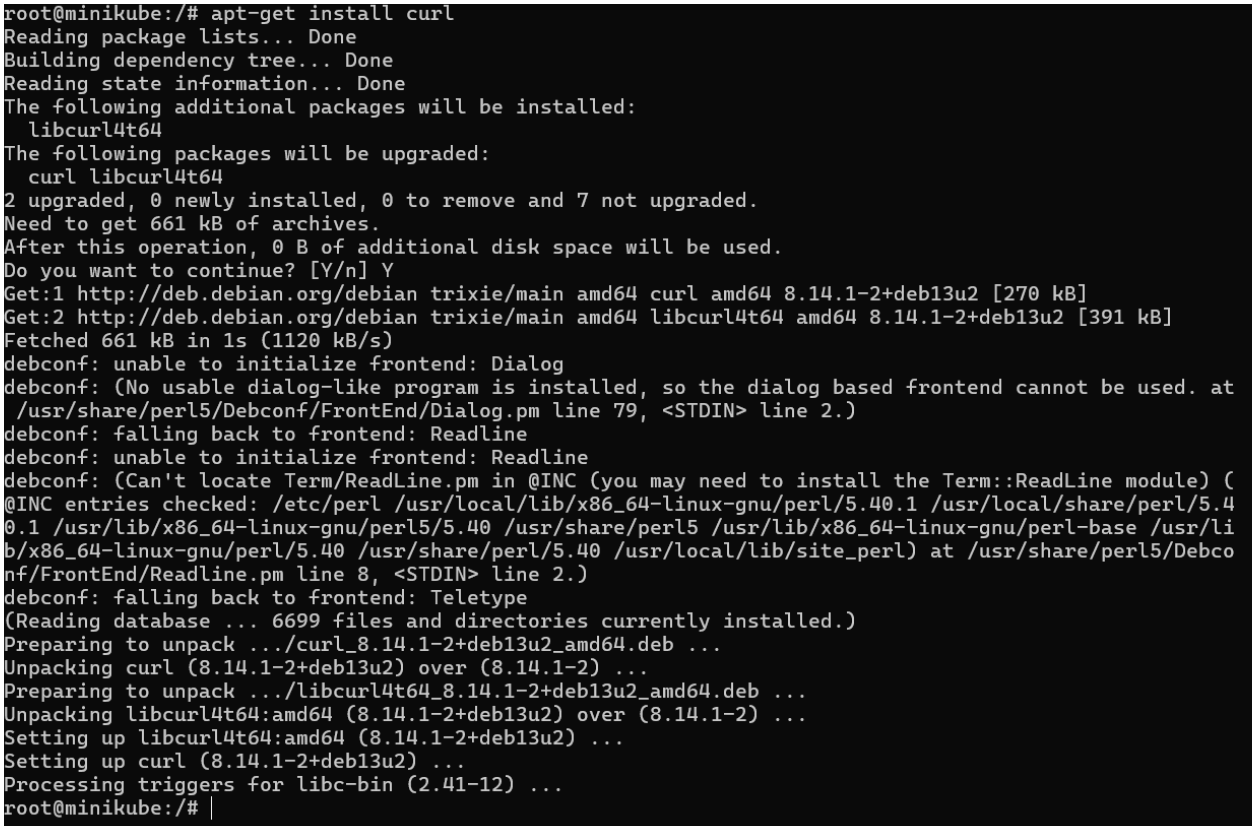
```
root@minikube:/# echo 'Hello shell demo' > /usr/share/nginx/html/index.html
root@minikube:/# |
```

STEP 8: send a GET request to the nginx server: apt-get update



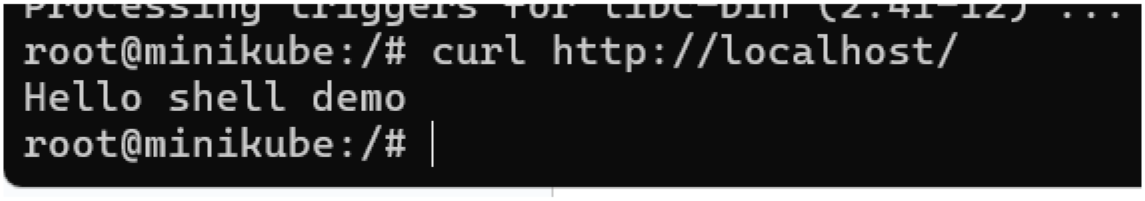
```
root@minikube:/# apt-get update
Get:1 http://deb.debian.org/debian trixie InRelease [140 kB]
Get:2 http://deb.debian.org/debian trixie-updates InRelease [47.3 kB]
Get:3 http://deb.debian.org/debian-security trixie-security InRelease [43.4 kB]
Get:4 http://deb.debian.org/debian trixie/main amd64 Packages [9670 kB]
Get:5 http://deb.debian.org/debian trixie-updates/main amd64 Packages [5412 B]
Get:6 http://deb.debian.org/debian-security trixie-security/main amd64 Packages [71.8 kB]
Fetched 9978 kB in 3s (3390 kB/s)
Reading package lists... Done
```

apt-get install curl



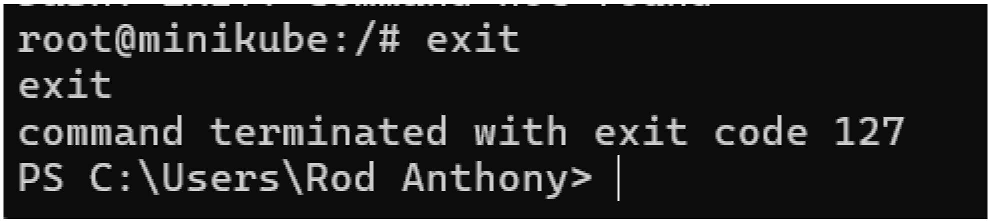
```
root@minikube:/# apt-get install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcurl4t64
The following packages will be upgraded:
  curl libcurl4t64
2 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
Need to get 661 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://deb.debian.org/debian trixie/main amd64 curl amd64 8.14.1-2+deb13u2 [270 kB]
Get:2 http://deb.debian.org/debian trixie/main amd64 libcurl4t64 amd64 8.14.1-2+deb13u2 [391 kB]
Fetched 661 kB in 1s (1120 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at
 /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 79, <STDIN> line 2.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (Can't locate Term/ReadLine.pm in @INC (you may need to install the Term::ReadLine module) (
@INC entries checked: /etc/perl /usr/local/lib/x86_64-linux-gnu/perl/5.40.1 /usr/local/share/perl/5.4
0.1 /usr/lib/x86_64-linux-gnu/perl5/5.40 /usr/share/perl5 /usr/lib/x86_64-linux-gnu/perl-base /usr/li
b/x86_64-linux-gnu/perl/5.40 /usr/share/perl/5.40 /usr/local/lib/site_perl) at /usr/share/perl5/Debco
nf/FrontEnd/Readline.pm line 8, <STDIN> line 2.)
debconf: falling back to frontend: Teletype
(Reading database ... 6699 files and directories currently installed.)
Preparing to unpack .../curl_8.14.1-2+deb13u2_amd64.deb ...
Unpacking curl (8.14.1-2+deb13u2) over (8.14.1-2) ...
Preparing to unpack .../libcurl4t64_8.14.1-2+deb13u2_amd64.deb ...
Unpacking libcurl4t64:amd64 (8.14.1-2+deb13u2) over (8.14.1-2) ...
Setting up libcurl4t64:amd64 (8.14.1-2+deb13u2) ...
Setting up curl (8.14.1-2+deb13u2) ...
Processing triggers for libc-bin (2.41-12) ...
root@minikube:/#
```

curl http://localhost/



```
Processing triggers for libc-bin (2.41-12) ...
root@minikube:/# curl http://localhost/
Hello shell demo
root@minikube:/#
```

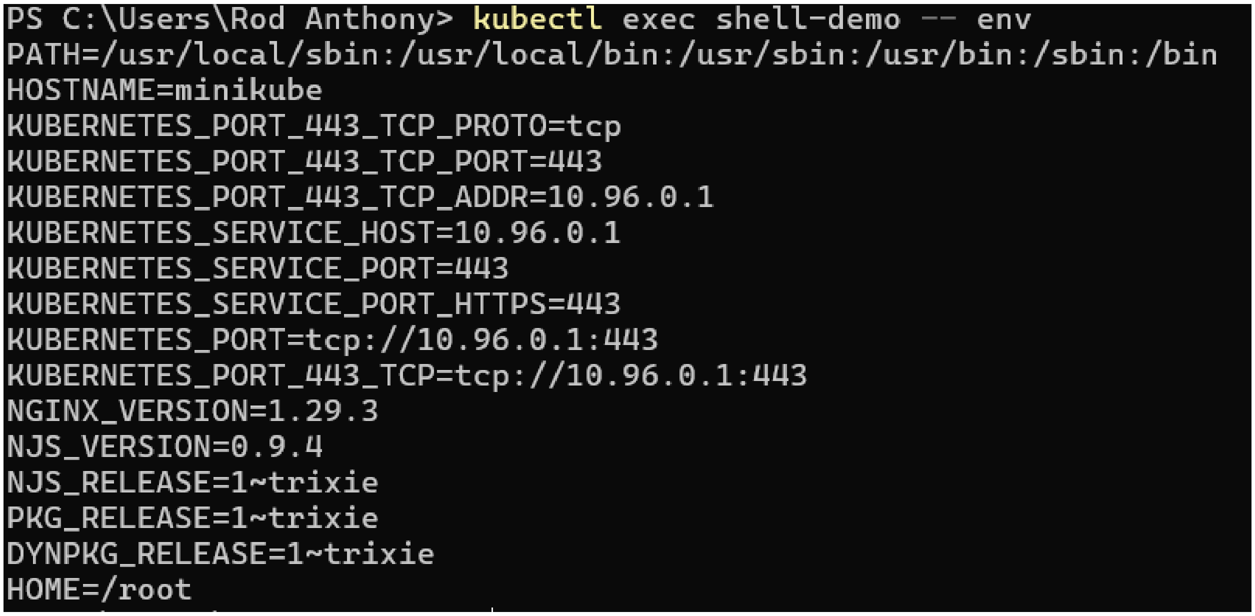
STEP 9: Exit exit



```
root@minikube:/# exit
exit
command terminated with exit code 127
PS C:\Users\Rod Anthony>
```

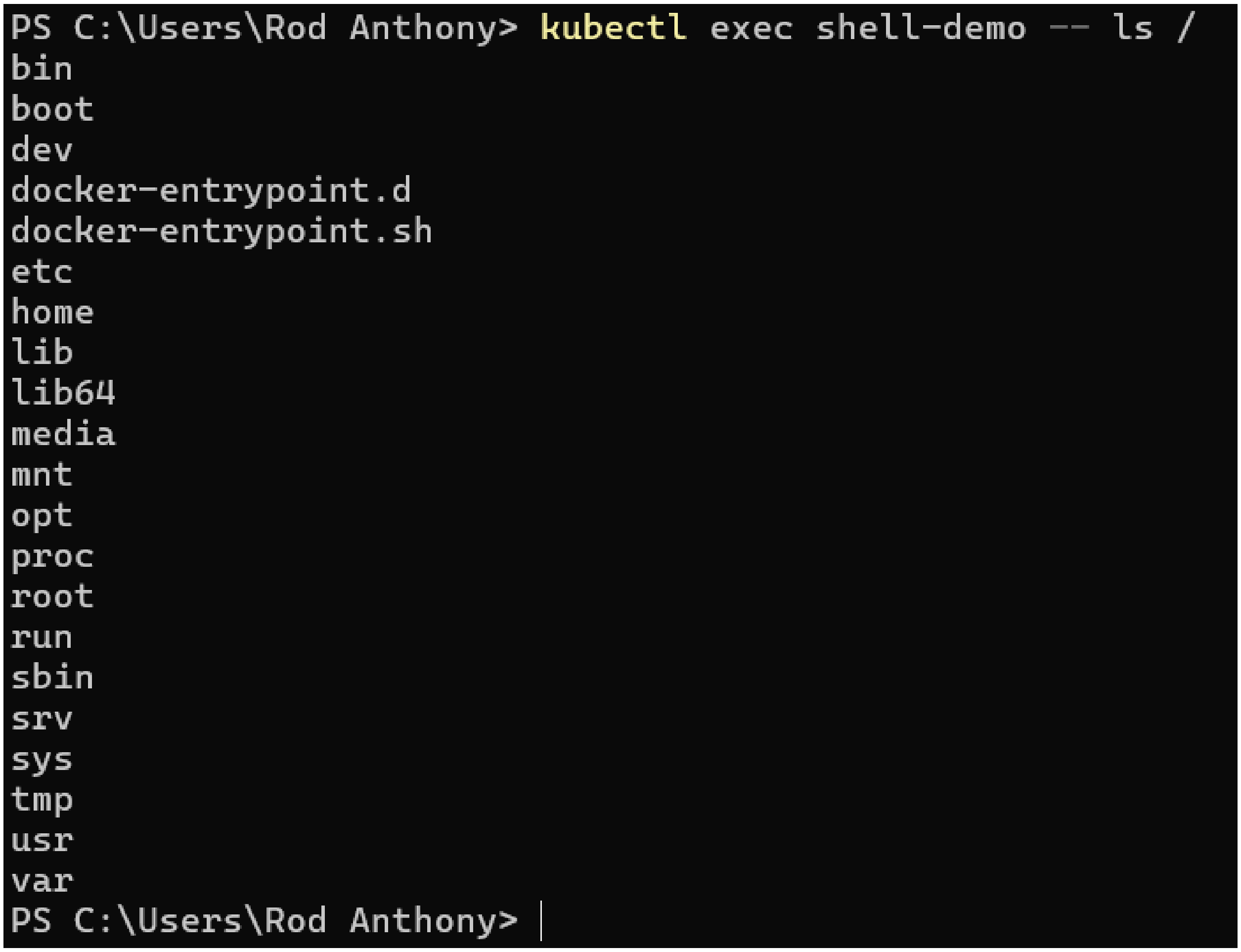
RUNNING INDIVIDUAL COMMANDS IN A CONTAINER

STEP 1: list the environment variables in the running container:



```
PS C:\Users\Rod Anthony> kubectl exec shell-demo -- env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=minikube
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
NGINX_VERSION=1.29.3
NJS_VERSION=0.9.4
NJS_RELEASE=1~trixie
PKG_RELEASE=1~trixie
DYNPKG_RELEASE=1~trixie
HOME=/root
```

STEP 2: Experiment with running other commands kubectl exec shell-demo -- cat /proc/1/mounts kubectl exec shell-demo -- ps aux
kubectl exec shell-demo -- ls /



```
PS C:\Users\Rod Anthony> kubectl exec shell-demo -- ls /
bin
boot
dev
docker-entrypoint.d
docker-entrypoint.sh
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
PS C:\Users\Rod Anthony>
```

OPENING A SHELL WHEN A POD HAS MORE THAN ONE CONTAINER

kubectl exec -i -t my-pod --container main-app -- /bin/bash

DEPLOYING WORDPRESS AND MYSQL WITH PERSISTENT VOLUMES

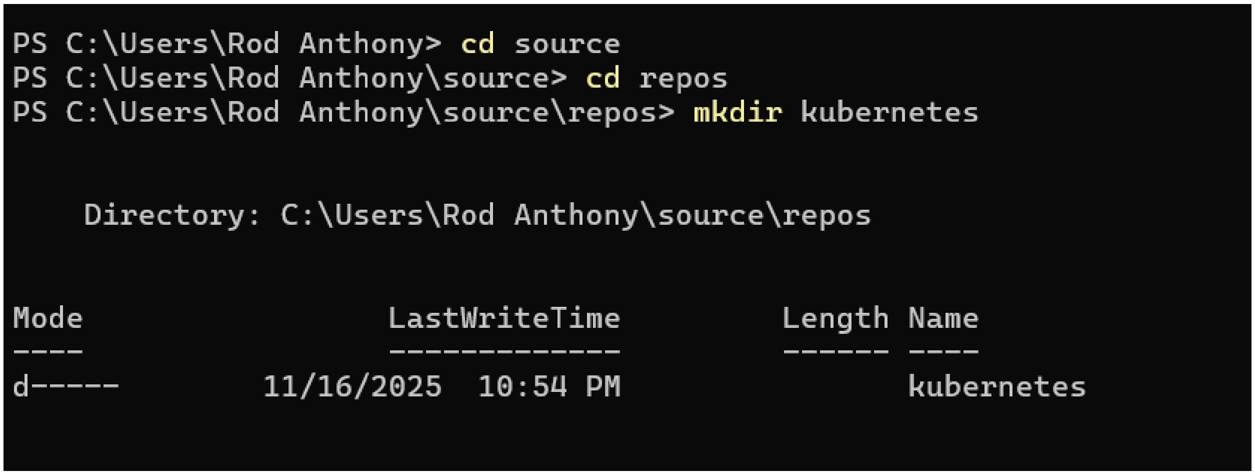
https://kubernetes.io/docs/tutorials/stateful-application/mysql-wordpress- persistent-volume

Objectives

- Create PersistentVolumeClaims and PersistentVolumes
- Create a kustomization.yaml with
  - a Secret generator

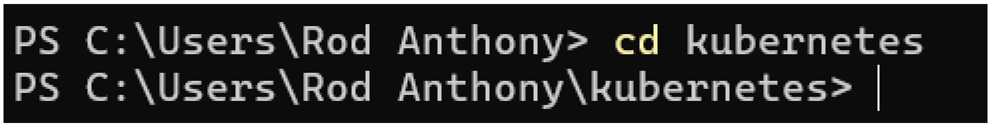
  - MySQL resource configs
  - WordPress resource configs
- Apply the kustomization directory by kubectl apply -k ./
- Clean up

STEP 1: make directory to keep file structure clean cd C:/Users/User/source/repo mkdir kubernetes



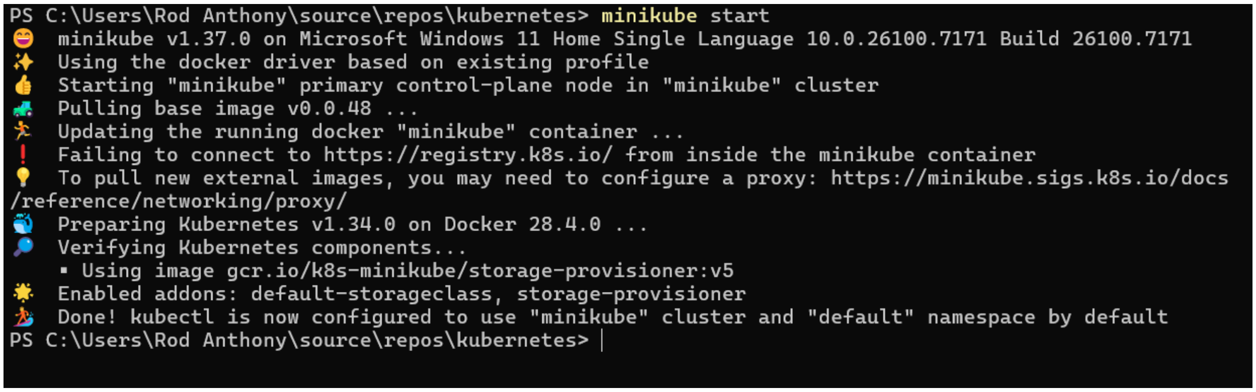
```
PS C:\Users\Rod Anthony> cd source
PS C:\Users\Rod Anthony\source> cd repos
PS C:\Users\Rod Anthony\source\repos> mkdir kubernetes


    Directory: C:\Users\Rod Anthony\source\repos


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/16/2025  10:54 PM                kubernetes
```

cd kubernetes

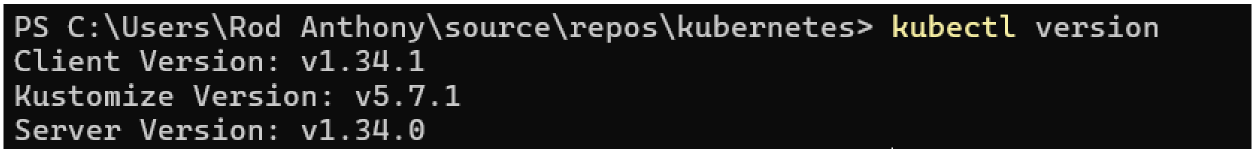


```
PS C:\Users\Rod Anthony> cd kubernetes
PS C:\Users\Rod Anthony\kubernetes> |
```

STEP 2: start minikube and verify kubectl version minikube start, kubectl version



```
PS C:\Users\Rod Anthony\source\repos\kubernetes> minikube start
😄  minikube v1.37.0 on Microsoft Windows 11 Home Single Language 10.0.26100.7171 Build 26100.7171
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.48 ...
🏃  Updating the running docker "minikube" container ...
❗  Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡  To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs
/reference/networking/proxy/
🐳  Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\Rod Anthony\source\repos\kubernetes> |
```

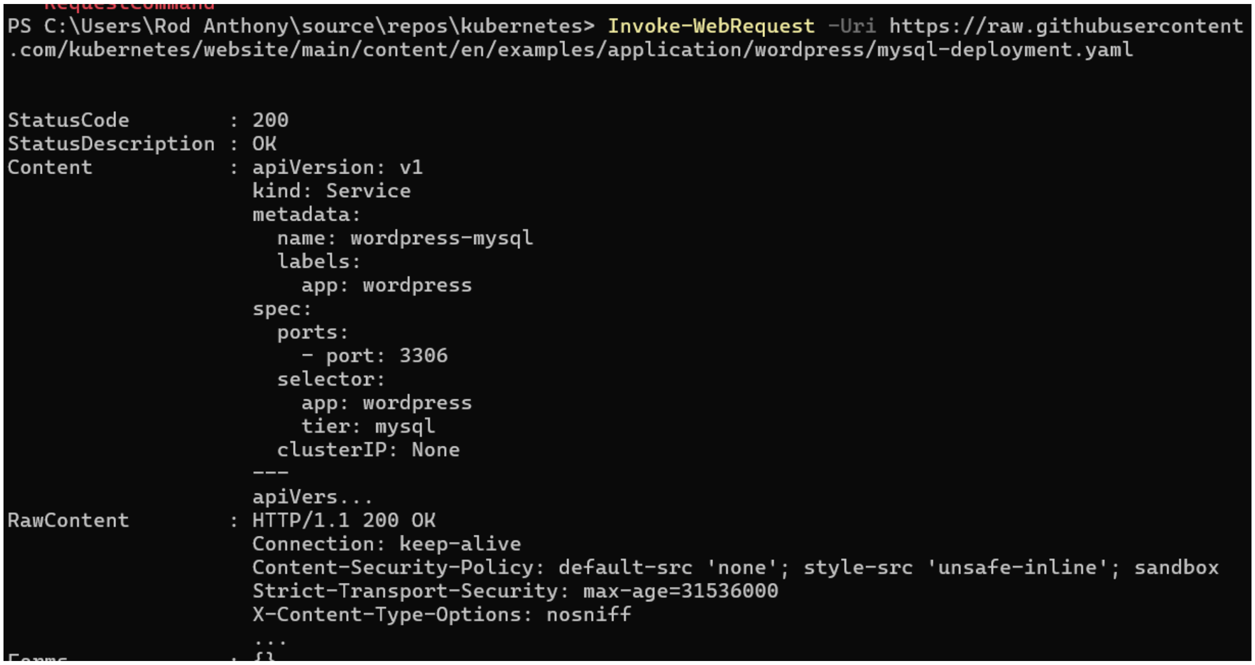


```
PS C:\Users\Rod Anthony\source\repos\kubernetes> kubectl version
Client Version: v1.34.1
Kustomize Version: v5.7.1
Server Version: v1.34.0
```

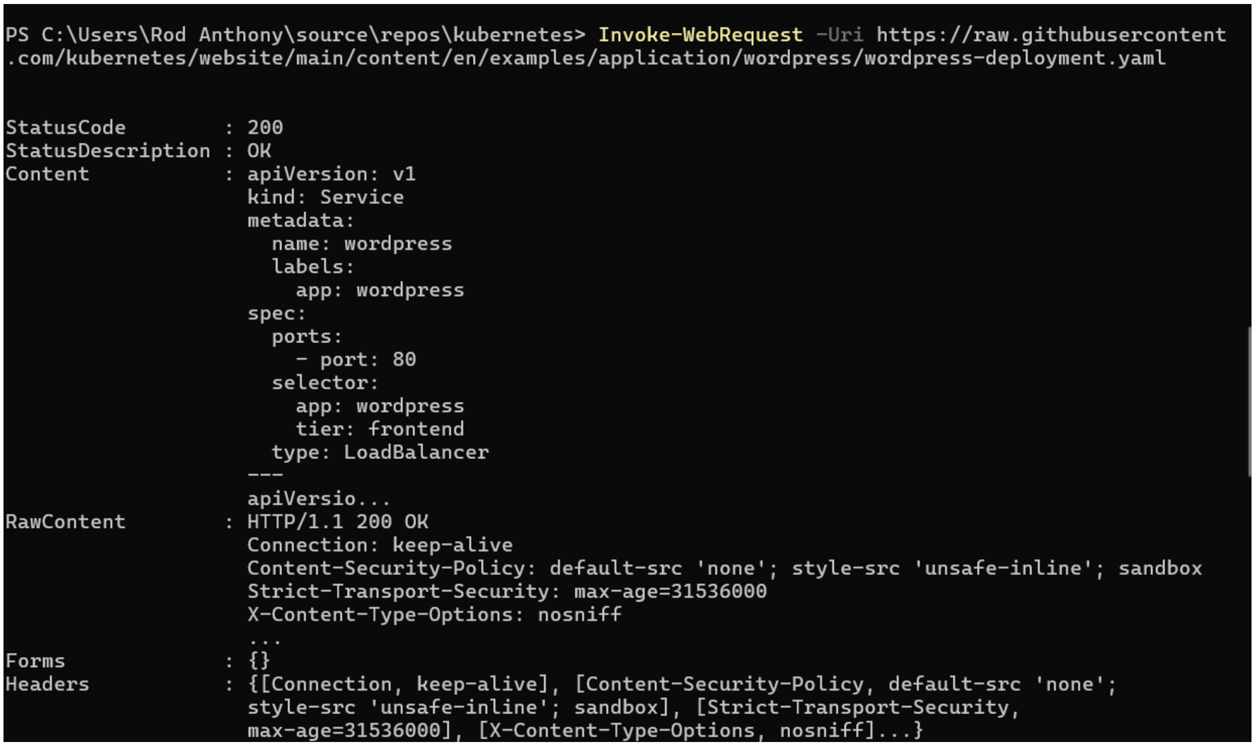
STEP 3: Download Deployment YAMLs Download MySQL and WordPress manifests:

# Download MySQL deployment YAML
Invoke-WebRequest -Uri "https://k8s.io/examples/application/wordpress/mysql-deployment.yaml" -OutFile "mysql-deployment.yaml"

# Download WordPress deployment YAML
Invoke-WebRequest -Uri "https://k8s.io/examples/application/wordpress/wordpress-deployment.yaml" -OutFile "wordpress-deployment.yaml"



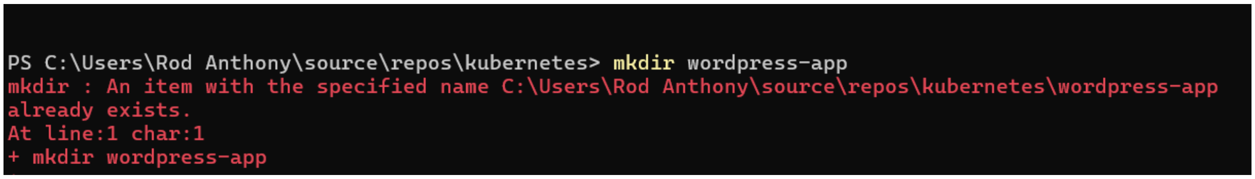
```
    RequestCommand
PS C:\Users\Rod Anthony\source\repos\kubernetes> Invoke-WebRequest -Uri https://raw.githubusercontent
.com/kubernetes/website/main/content/en/examples/application/wordpress/mysql-deployment.yaml


StatusCode        : 200
StatusDescription : OK
Content           : apiVersion: v1
                    kind: Service
                    metadata:
                      name: wordpress-mysql
                      labels:
                        app: wordpress
                    spec:
                      ports:
                        - port: 3306
                      selector:
                        app: wordpress
                        tier: mysql
                      clusterIP: None
                    ---
                    apiVers...
RawContent        : HTTP/1.1 200 OK
                    Connection: keep-alive
                    Content-Security-Policy: default-src 'none'; style-src 'unsafe-inline'; sandbox
                    Strict-Transport-Security: max-age=31536000
                    X-Content-Type-Options: nosniff
                    ...
Forms             : {}
```



```
PS C:\Users\Rod Anthony\source\repos\kubernetes> Invoke-WebRequest -Uri https://raw.githubusercontent
.com/kubernetes/website/main/content/en/examples/application/wordpress/wordpress-deployment.yaml


StatusCode        : 200
StatusDescription : OK
Content           : apiVersion: v1
                    kind: Service
                    metadata:
                      name: wordpress
                      labels:
                        app: wordpress
                    spec:
                      ports:
                        - port: 80
                      selector:
                        app: wordpress
                        tier: frontend
                      type: LoadBalancer
                    ---
                    apiVersio...
RawContent        : HTTP/1.1 200 OK
                    Connection: keep-alive
                    Content-Security-Policy: default-src 'none'; style-src 'unsafe-inline'; sandbox
                    Strict-Transport-Security: max-age=31536000
                    X-Content-Type-Options: nosniff
                    ...
Forms             : {}
Headers           : {[Connection, keep-alive], [Content-Security-Policy, default-src 'none';
                    style-src 'unsafe-inline'; sandbox], [Strict-Transport-Security,
                    max-age=31536000], [X-Content-Type-Options, nosniff]...}
```

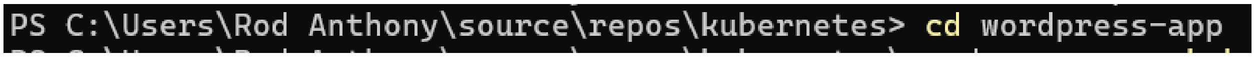
STEP 4: create another directory named wordpress-app mkdir wordpress-app

STEP 5: move YAML files inside the wordpress-app mv mysql-deployment.yaml wordpress-app/ mv wordpress-deployment.yaml wordpress-app/



```
PS C:\Users\Rod Anthony\source\repos\kubernetes> mkdir wordpress-app
mkdir : An item with the specified name C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app
already exists.
At line:1 char:1
+ mkdir wordpress-app
```

STEP 6: change directory to wordpress-app

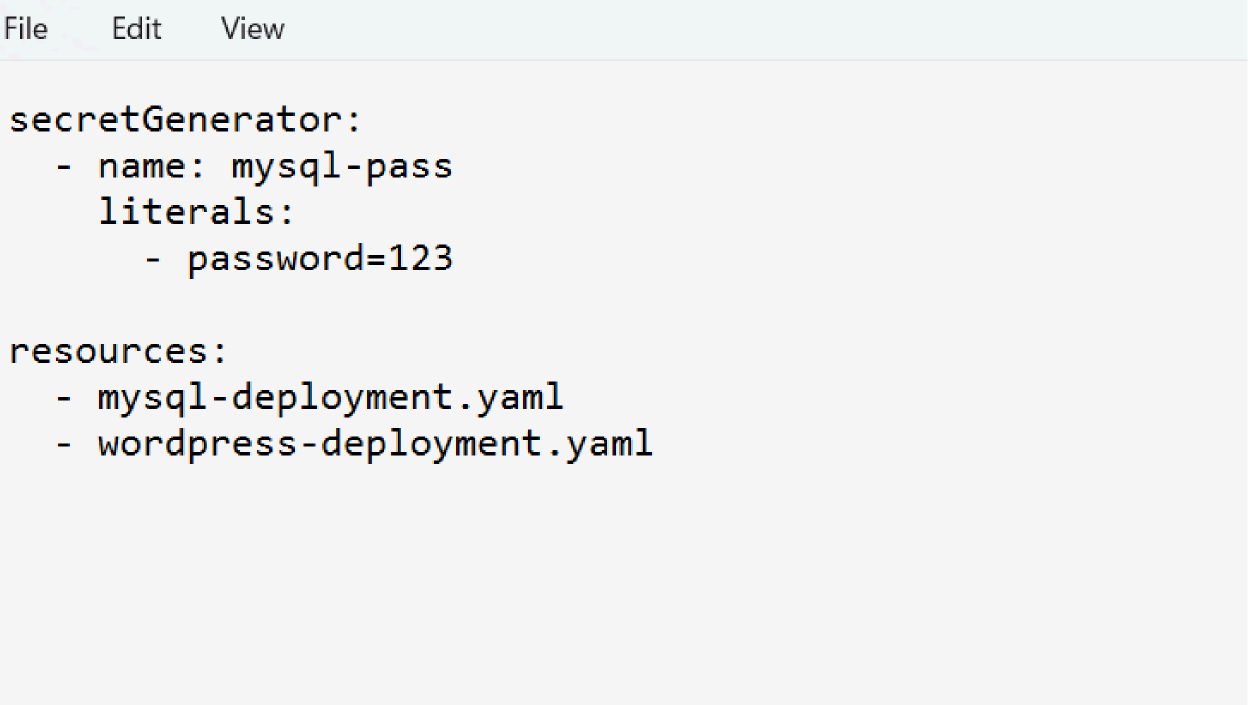


```
PS C:\Users\Rod Anthony\source\repos\kubernetes> cd wordpress-app
```

STEP 7: Create a Secret generator in kustomization.yaml

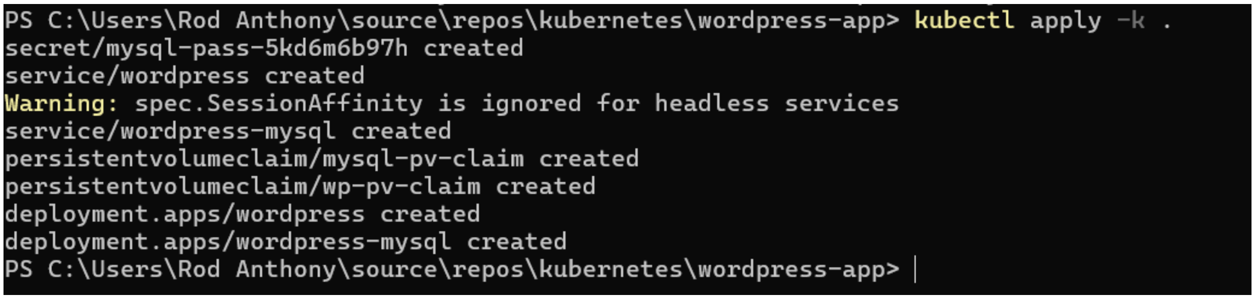
notepad kustomization.yaml secretGenerator:

- name: mysql-pass literals:
- password=123 resources:
- mysql-deployment.yaml
- wordpress-deployment.yaml



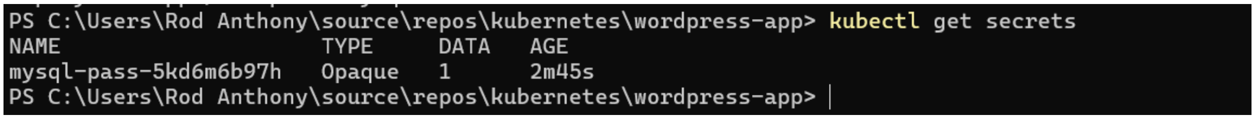
```
secretGenerator:
  - name: mysql-pass
    literals:
      - password=123

resources:
  - mysql-deployment.yaml
  - wordpress-deployment.yaml
```

STEP 8:Deploy WordPress and MySQL using Kustomize: kubectl apply -k ./



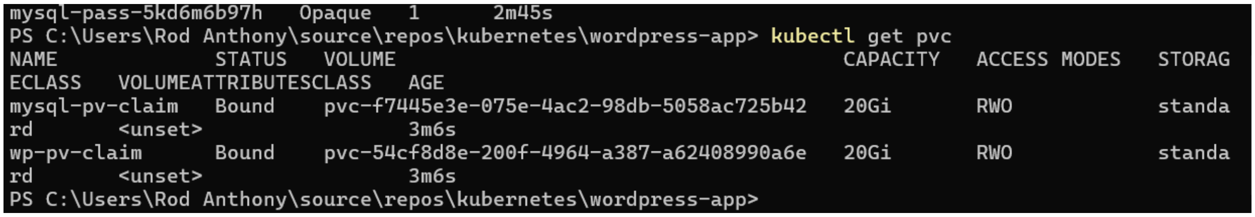
```
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> kubectl apply -k .
secret/mysql-pass-5kd6m6b97h created
service/wordpress created
Warning: spec.SessionAffinity is ignored for headless services
service/wordpress-mysql created
persistentvolumeclaim/mysql-pv-claim created
persistentvolumeclaim/wp-pv-claim created
deployment.apps/wordpress created
deployment.apps/wordpress-mysql created
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app>
```

STEP 9: Check Secrets kubectl get secrets



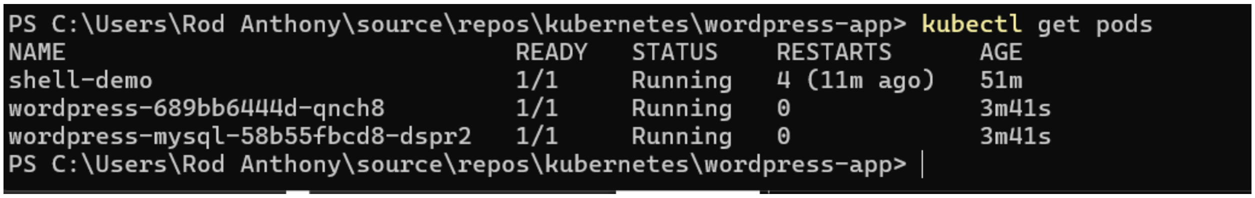
```
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> kubectl get secrets
NAME                    TYPE     DATA   AGE
mysql-pass-5kd6m6b97h   Opaque   1      2m45s
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app>
```

STEP 10: Check PersistentVolumeClaims (PVCs): kubectl get pvc



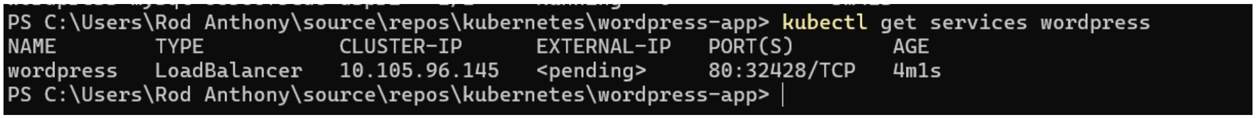
```
mysql-pass-5kd6m6b97h   Opaque   1      2m45s
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> kubectl get pvc
NAME             STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAG
ECLASS   VOLUMEATTRIBUTESCLASS   AGE
mysql-pv-claim   Bound    pvc-f7445e3e-075e-4ac2-98db-5058ac725b42   20Gi       RWO            standa
rd       <unset>                 3m6s
wp-pv-claim      Bound    pvc-54cf8d8e-200f-4964-a387-a62408990a6e   20Gi       RWO            standa
rd       <unset>                 3m6s
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app>
```

STEP 11: Check pods (wordpress and mysql should be 1/1 ready



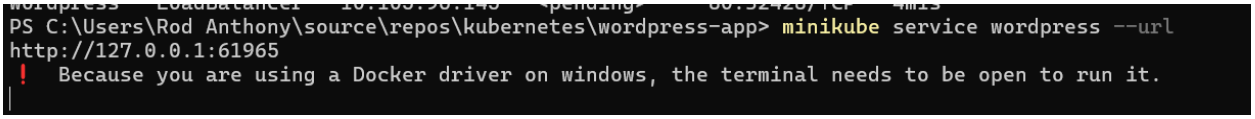
```
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> kubectl get pods
NAME                              READY   STATUS    RESTARTS       AGE
shell-demo                        1/1     Running   4 (11m ago)    51m
wordpress-689bb6444d-qnch8        1/1     Running   0              3m41s
wordpress-mysql-58b55fbcd8-dspr2  1/1     Running   0              3m41s
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> |
```

STEP 12: Check services kubectl get services wordpress

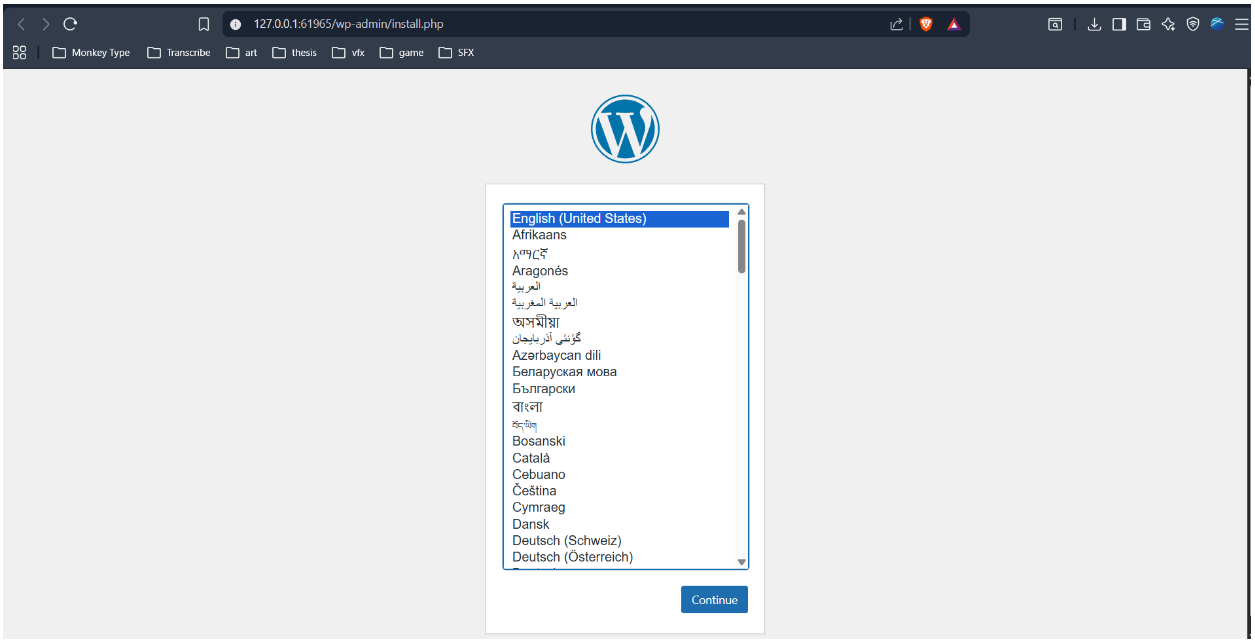


```
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> kubectl get services wordpress
NAME        TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
wordpress   LoadBalancer   10.105.96.145   <pending>     80:32428/TCP   4m1s
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> |
```

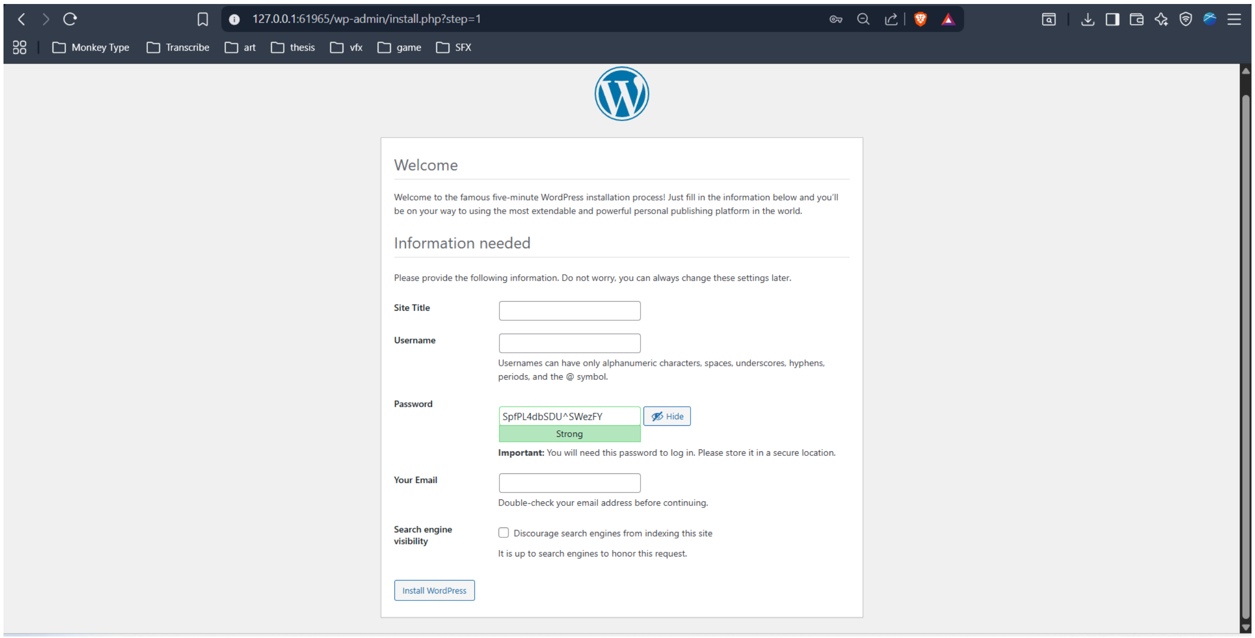
STEP 13: Access WordPress



```
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> minikube service wordpress --url
http://127.0.0.1:61965
❗  Because you are using a Docker driver on windows, the terminal needs to be open to run it.
|
```



English (United States)
Afrikaans
አማርኛ
Aragonés
العربية
العربية المغربية
অসমীয়া
گؤنئی آذربایجان
Azərbaycan dili
Беларуская мова
Български
বাংলা
བོད་ཡིག
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch (Österreich)



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password SpfPL4dbSDU^SWezFY Hide

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

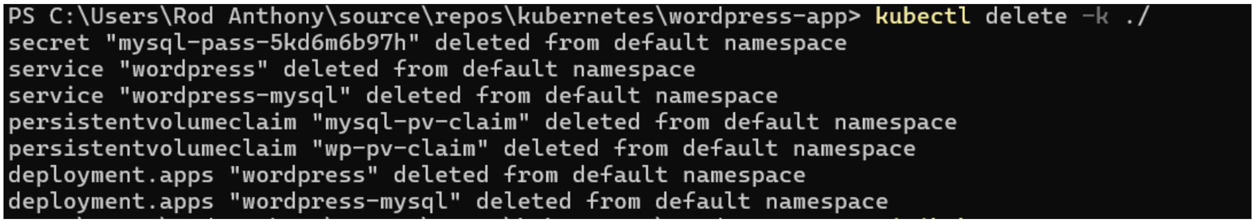
Double-check your email address before continuing.

Search engine visibility □ Discourage search engines from indexing this site

It is up to search engines to honor this request.

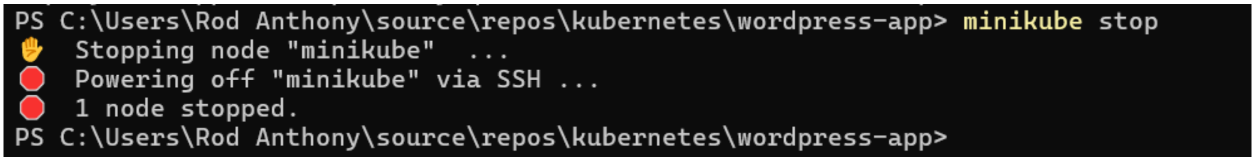
Install WordPress

STEP 14: Clean up kubectl delete -k ./



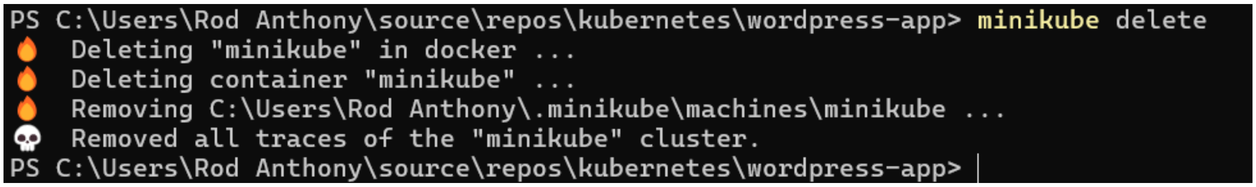
```
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> kubectl delete -k ./
secret "mysql-pass-5kd6m6b97h" deleted from default namespace
service "wordpress" deleted from default namespace
service "wordpress-mysql" deleted from default namespace
persistentvolumeclaim "mysql-pv-claim" deleted from default namespace
persistentvolumeclaim "wp-pv-claim" deleted from default namespace
deployment.apps "wordpress" deleted from default namespace
deployment.apps "wordpress-mysql" deleted from default namespace
```

minikube stop



```
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> minikube stop
✋  Stopping node "minikube"  ...
🛑  Powering off "minikube" via SSH ...
🛑  1 node stopped.
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app>
```

minikube delete



```
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app> minikube delete
🔥  Deleting "minikube" in docker ...
🔥  Deleting container "minikube" ...
🔥  Removing C:\Users\Rod Anthony\.minikube\machines\minikube ...
💀  Removed all traces of the "minikube" cluster.
PS C:\Users\Rod Anthony\source\repos\kubernetes\wordpress-app>
```