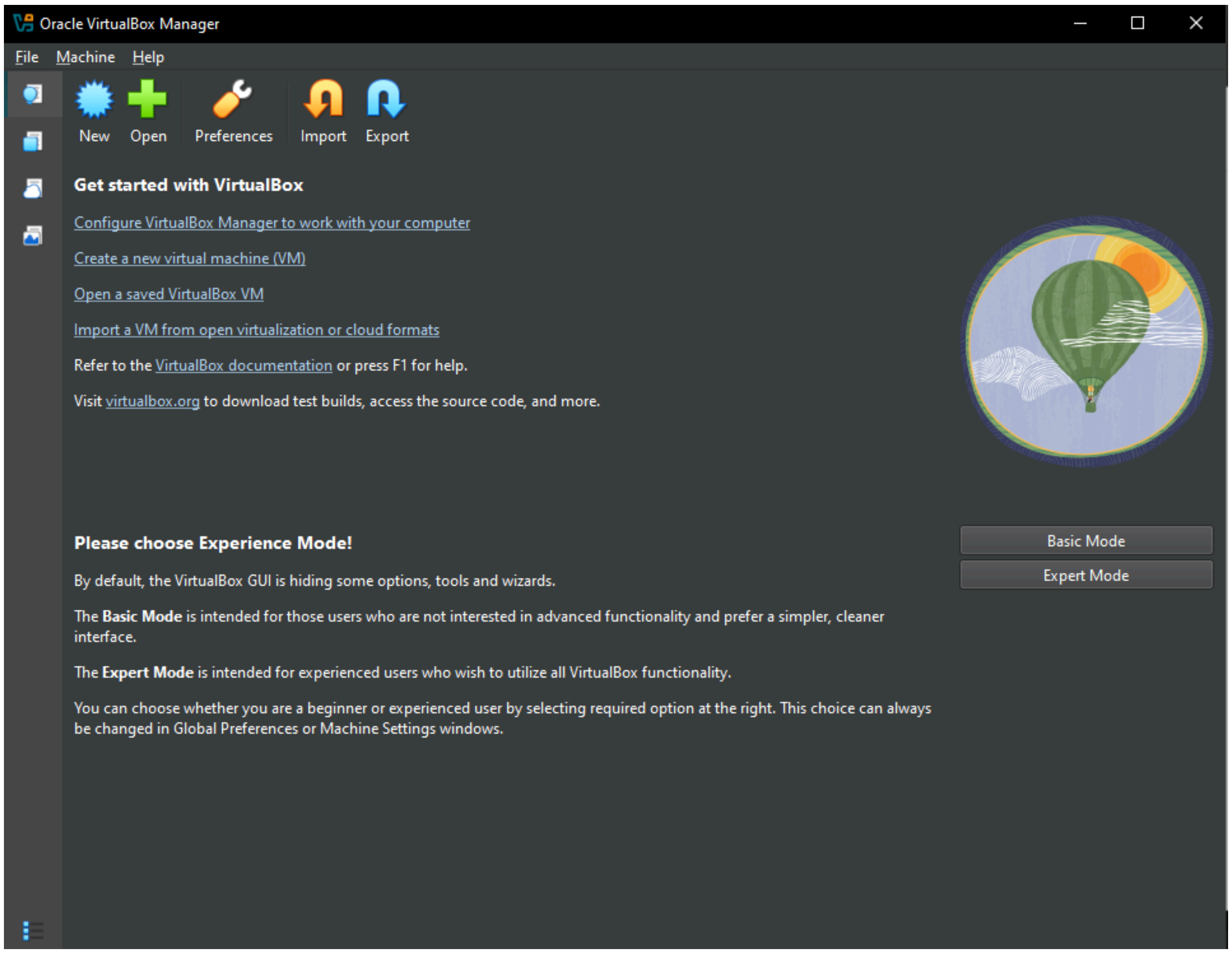
Jeremiah Dela Vega
IV-BCSAD

Elective 3
Assignment# 5 - Kubernetes Home Lab Activity

Hello Minikube:

Tools needed:
Minikube
Kubectl
Virtual Box

Virtual Box Main Screen (after installation)



Get started with VirtualBox

Configure VirtualBox Manager to work with your computer

Create a new virtual machine (VM)

Open a saved VirtualBox VM

Import a VM from open virtualization or cloud formats

Refer to the VirtualBox documentation or press F1 for help.

Visit virtualbox.org to download test builds, access the source code, and more.

Please choose Experience Mode!

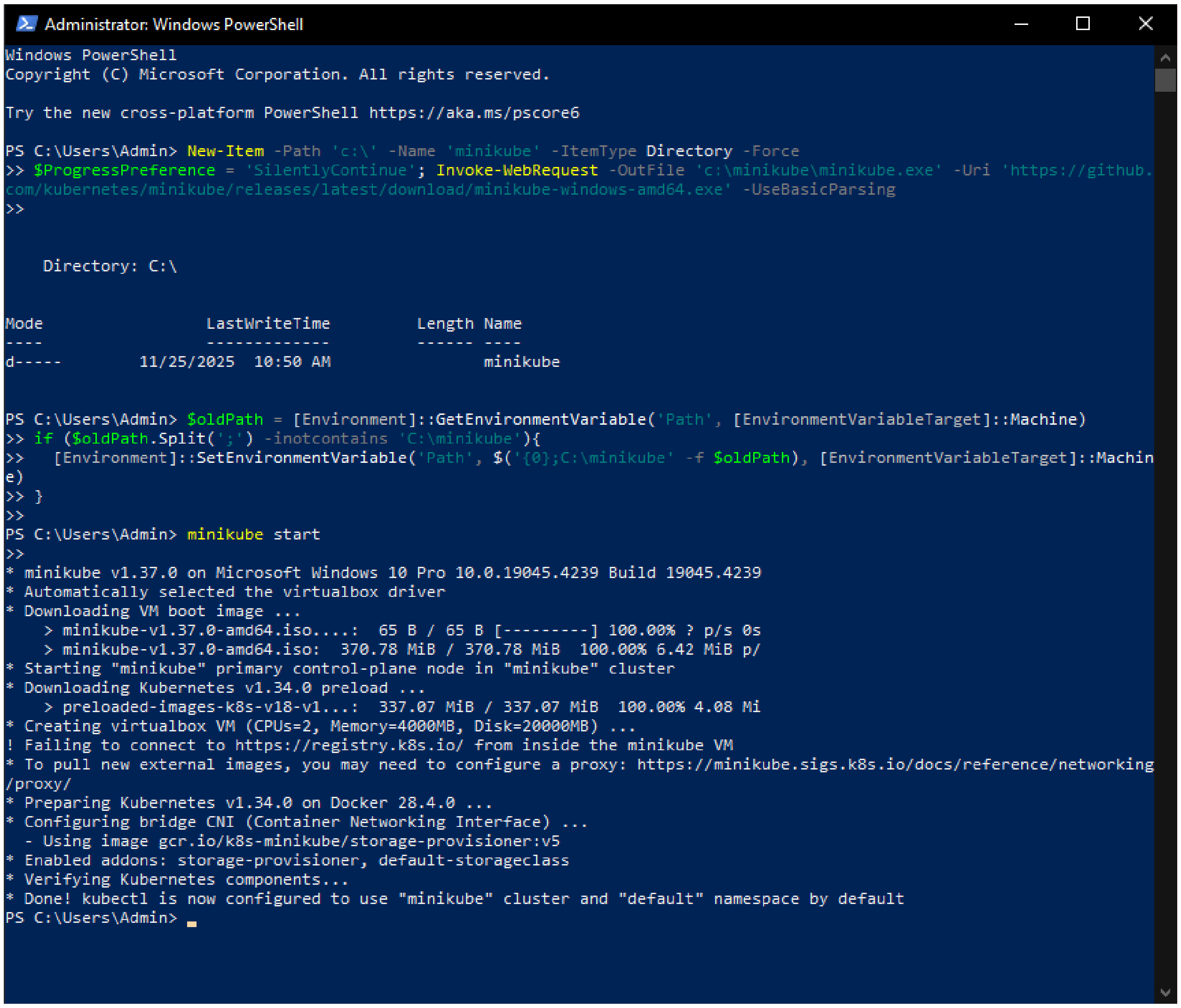
By default, the VirtualBox GUI is hiding some options, tools and wizards.

The **Basic Mode** is intended for those users who are not interested in advanced functionality and prefer a simpler, cleaner interface.

The **Expert Mode** is intended for experienced users who wish to utilize all VirtualBox functionality.

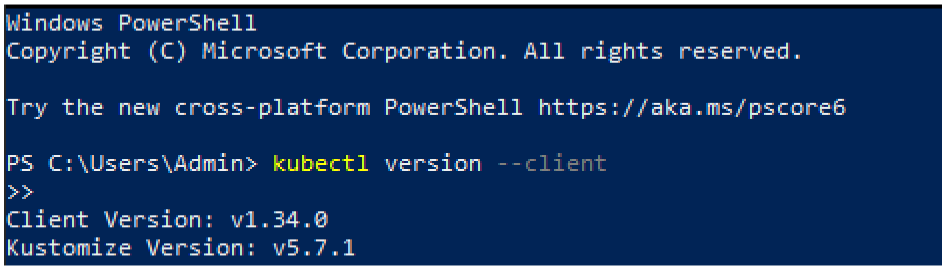
You can choose whether you are a beginner or experienced user by selecting required option at the right. This choice can always be changed in Global Preferences or Machine Settings windows.

Start the Minikube using the ‘minikube start’ command



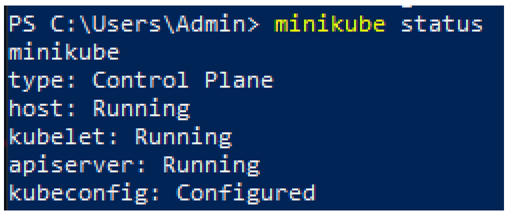
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Admin> New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force
>> $ProgressPreference = 'SilentlyContinue'; Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.
com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe' -UseBasicParsing
>>


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/25/2025  10:50 AM                minikube


PS C:\Users\Admin> $oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::Machine)
>> if ($oldPath.Split(';') -inotcontains 'C:\minikube'){
>>   [Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube' -f $oldPath), [EnvironmentVariableTarget]::Machin
e)
>> }
>>
PS C:\Users\Admin> minikube start
>>
* minikube v1.37.0 on Microsoft Windows 10 Pro 10.0.19045.4239 Build 19045.4239
* Automatically selected the virtualbox driver
* Downloading VM boot image ...
    > minikube-v1.37.0-amd64.iso....:  65 B / 65 B [---------] 100.00% ? p/s 0s
    > minikube-v1.37.0-amd64.iso:  370.78 MiB / 370.78 MiB  100.00% 6.42 MiB p/
* Starting "minikube" primary control-plane node in "minikube" cluster
* Downloading Kubernetes v1.34.0 preload ...
    > preloaded-images-k8s-v18-v1...:  337.07 MiB / 337.07 MiB  100.00% 4.08 Mi
* Creating virtualbox VM (CPUs=2, Memory=4000MB, Disk=20000MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube VM
* 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) ...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Verifying Kubernetes components...
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\Admin>
```

Check kubectl version



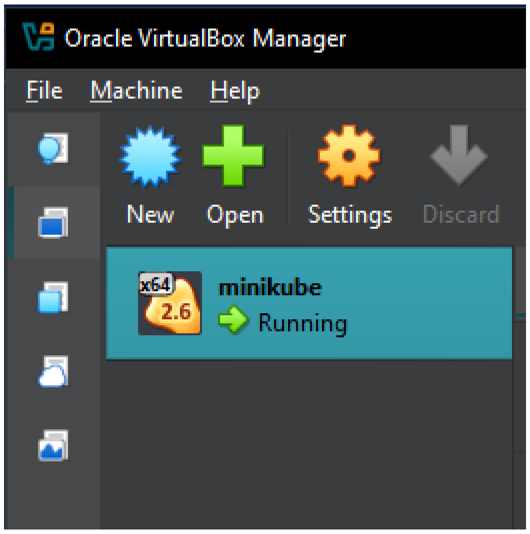
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Admin> kubectl version --client
>>
Client Version: v1.34.0
Kustomize Version: v5.7.1
```

Check if minikube is running using minikube status command

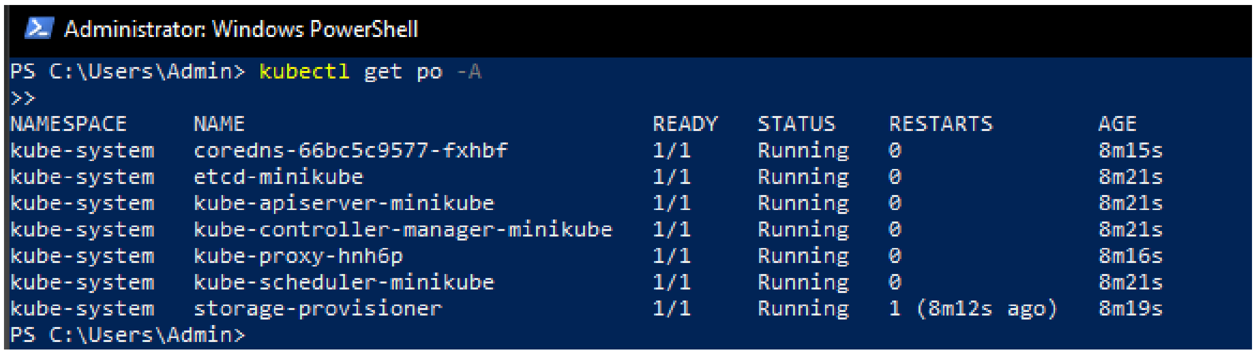


```
PS C:\Users\Admin> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

Minikube will run on Virtual Box automatically after starting it on terminal



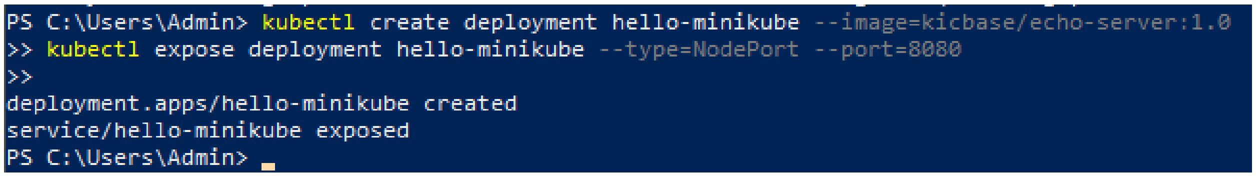
Access new cluster using this command



```
Administrator: Windows PowerShell
PS C:\Users\Admin> kubectl get po -A
>>
NAMESPACE     NAME                               READY   STATUS    RESTARTS         AGE
kube-system   coredns-66bc5c9577-fxhbf           1/1     Running   0                8m15s
kube-system   etcd-minikube                      1/1     Running   0                8m21s
kube-system   kube-apiserver-minikube            1/1     Running   0                8m21s
kube-system   kube-controller-manager-minikube   1/1     Running   0                8m21s
kube-system   kube-proxy-hnh6p                   1/1     Running   0                8m16s
kube-system   kube-scheduler-minikube            1/1     Running   0                8m21s
kube-system   storage-provisioner                1/1     Running   1 (8m12s ago)    8m19s
PS C:\Users\Admin>
```

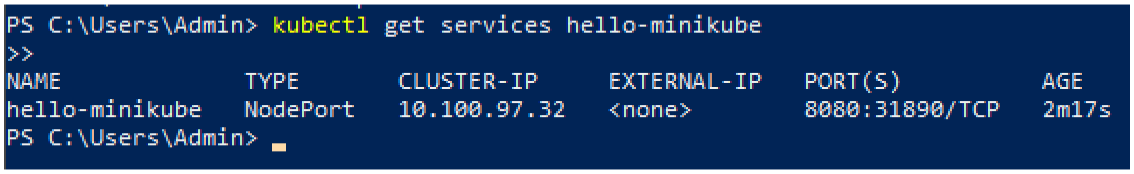
Next step is Deployment

Use this command to make a sample deployment and expose it on port 8080



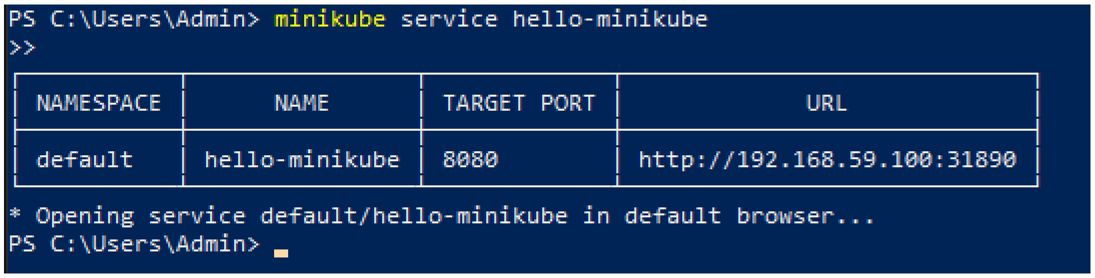
```
PS C:\Users\Admin> kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
>> kubectl expose deployment hello-minikube --type=NodePort --port=8080
>>
deployment.apps/hello-minikube created
service/hello-minikube exposed
PS C:\Users\Admin>
```

Then use this command to retrieve the information about Kubernetes services



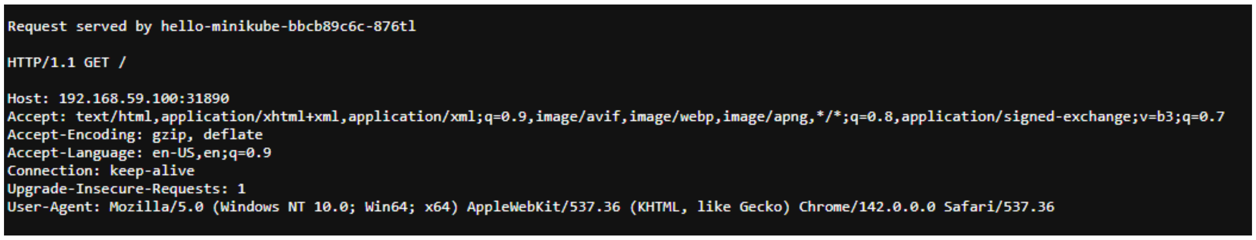
```
PS C:\Users\Admin> kubectl get services hello-minikube
>>
NAME             TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort   10.100.97.32   <none>        8080:31890/TCP   2m17s
PS C:\Users\Admin>
```

Then use this command to let minikube launch a web browser in order to access this service



```
PS C:\Users\Admin> minikube service hello-minikube
>>
|-----------|----------------|-------------|-----------------------------|
| NAMESPACE |      NAME      | TARGET PORT |             URL             |
|-----------|----------------|-------------|-----------------------------|
| default   | hello-minikube | 8080        | http://192.168.59.100:31890 |
|-----------|----------------|-------------|-----------------------------|
* Opening service default/hello-minikube in default browser...
PS C:\Users\Admin>
```

Now, we can see the detail of the application and this is the proof that the system is working



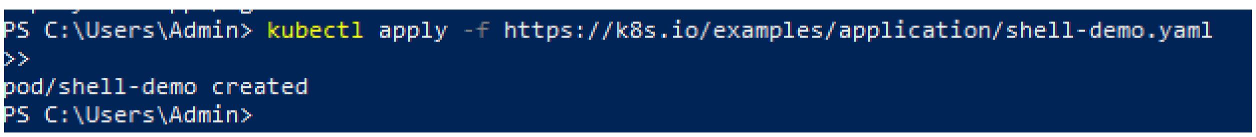
```
Request served by hello-minikube-bbcb89c6c-876tl

HTTP/1.1 GET /

Host: 192.168.59.100:31890
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
```

Get a Shell to a Running Container:

In this exercise, create a Pod running Nginx, but since Task 3 focuses on WordPress and MySQL with persistent volumes, I will skip steps that are not necessary for learning Pod creation and container access.

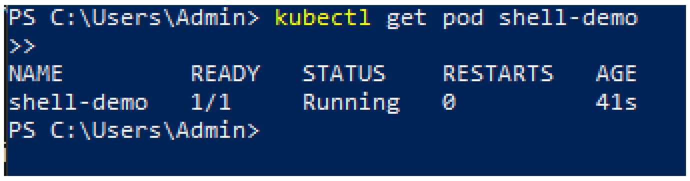
Step 1: Create the Pod

Create the shell-demo Pod:



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

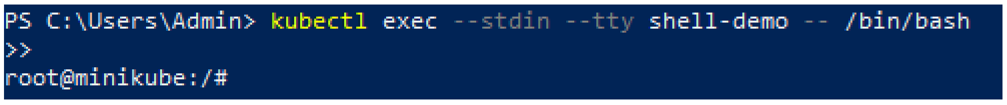
Verify that the Pod is running:



```
PS C:\Users\Admin> kubectl get pod shell-demo
>>
NAME         READY   STATUS    RESTARTS   AGE
shell-demo   1/1     Running   0          41s
PS C:\Users\Admin>
```

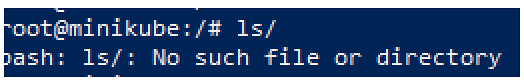
Step 2: Open a Shell in the Container

Open an interactive shell:



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

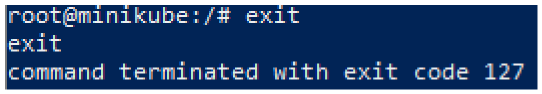
Inside the container, you can check basic info if needed:



```
oot@minikube:/# ls/
ash: ls/: No such file or directory
```

Skipped: Installing additional tools (tcpdump, lsof, procps) and running commands like ps aux | grep nginx because Task 3 focuses on multi-container deployments and persistent volumes, not container debugging.

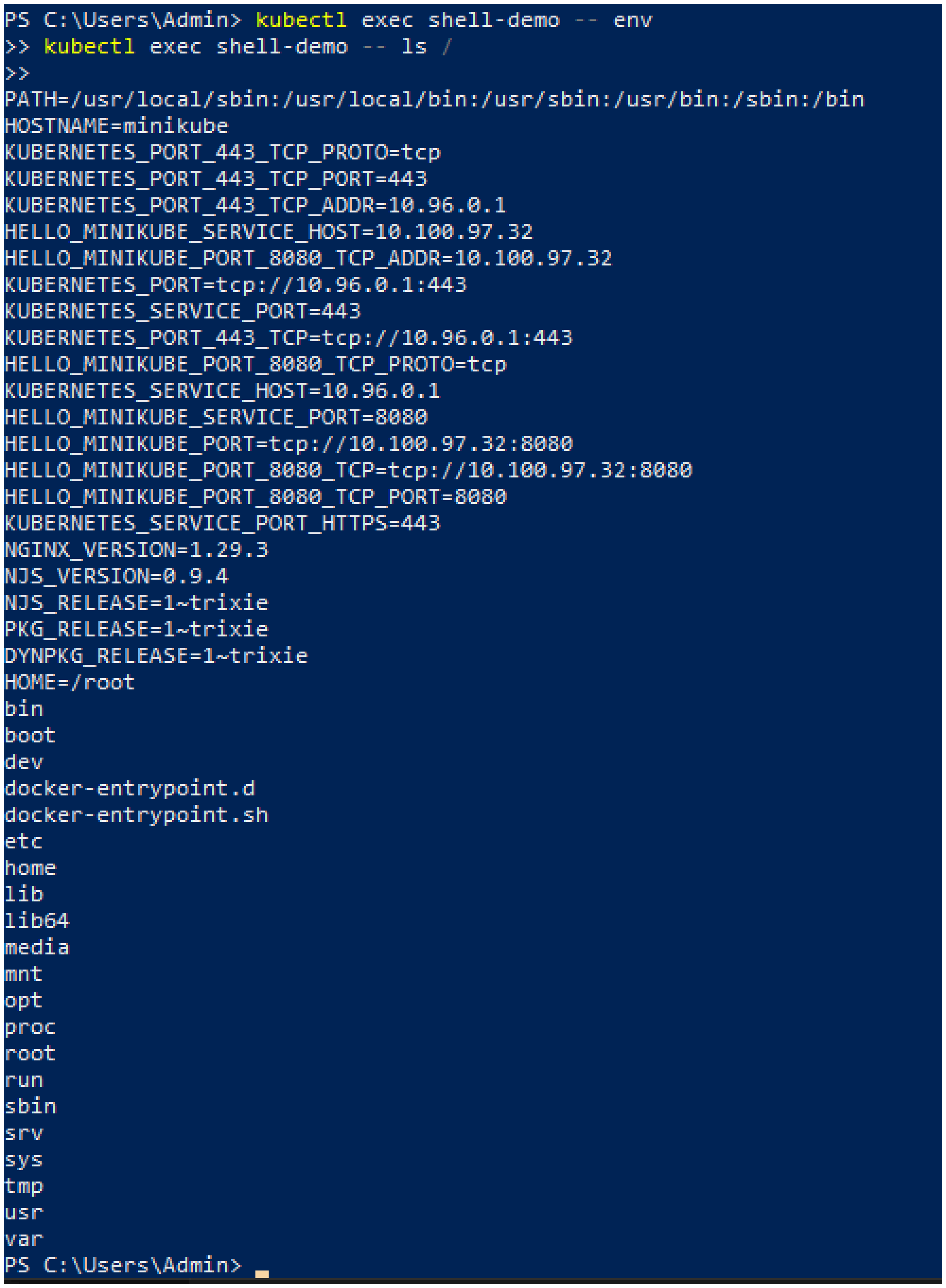
Exit the shell when done:



```
root@minikube:/# exit
exit
command terminated with exit code 127
```

Step 3: Running Individual Commands (Optional)

You can still run simple commands without opening a shell:



```
PS C:\Users\Admin> kubectl exec shell-demo -- env
>> kubectl exec shell-demo -- ls /
>>
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
HELLO_MINIKUBE_SERVICE_HOST=10.100.97.32
HELLO_MINIKUBE_PORT_8080_TCP_ADDR=10.100.97.32
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_SERVICE_PORT=443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
HELLO_MINIKUBE_PORT_8080_TCP_PROTO=tcp
KUBERNETES_SERVICE_HOST=10.96.0.1
HELLO_MINIKUBE_SERVICE_PORT=8080
HELLO_MINIKUBE_PORT=tcp://10.100.97.32:8080
HELLO_MINIKUBE_PORT_8080_TCP=tcp://10.100.97.32:8080
HELLO_MINIKUBE_PORT_8080_TCP_PORT=8080
KUBERNETES_SERVICE_PORT_HTTPS=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
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\Admin> _
```

Skipped: Any further container experimentation because it's not required for deploying WordPress and MySQL.

Step 4: Multi-Container Pods (Optional Reference)

If a Pod has more than one container, you can specify which container to access:

‘kubectl exec -i -t my-pod --container main-app -- /bin/bash’

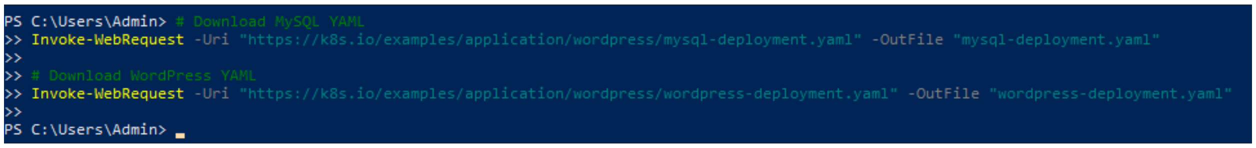
Skipped: Practicing this now because WordPress/MySQL deployment will naturally involve multi-container Pods, and we will cover it in Task 3.

Deploying Wordpress and MySQL with persistent volumes:

This task deploys a WordPress site with a MySQL backend using PersistentVolumes (PV) and PersistentVolumeClaims (PVC) on Minikube.

Step 1: Download Deployment Files

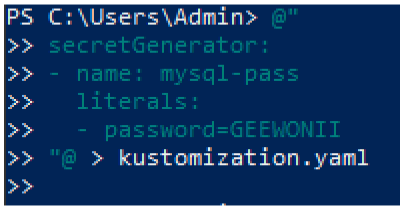
Download the MySQL and WordPress deployment YAMLs:



```
PS C:\Users\Admin> # Download MySQL YAML
>> Invoke-WebRequest -Uri "https://k8s.io/examples/application/wordpress/mysql-deployment.yaml" -OutFile "mysql-deployment.yaml"
>>
>> # Download WordPress YAML
>> Invoke-WebRequest -Uri "https://k8s.io/examples/application/wordpress/wordpress-deployment.yaml" -OutFile "wordpress-deployment.yaml"
>>
PS C:\Users\Admin> _
```

Step 2: Create a Secret for MySQL Password

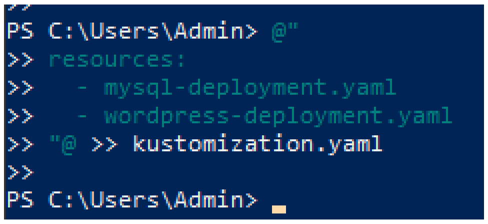
Create kustomization.yaml with a Secret generator:



```
PS C:\Users\Admin> @"
>> secretGenerator:
>> - name: mysql-pass
>>   literals:
>>   - password=GEEWONII
>> "@ > kustomization.yaml
>>
```

Step 3: Add Deployment Resources to Kustomization

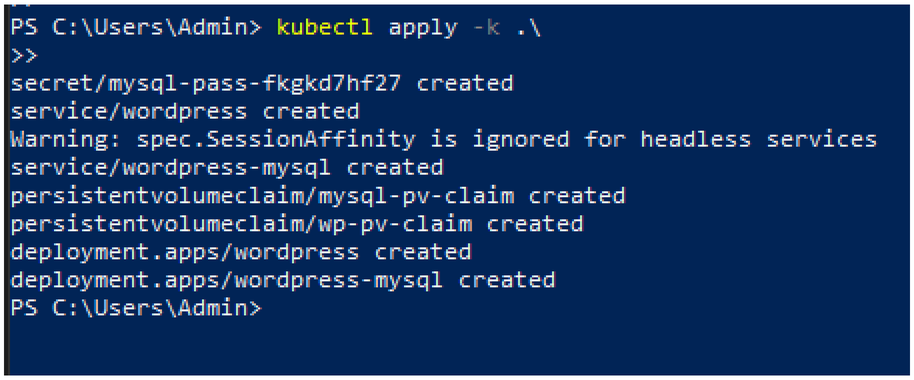
Append the MySQL and WordPress YAML files to kustomization.yaml:



```
PS C:\Users\Admin> @"
>> resources:
>>   - mysql-deployment.yaml
>>   - wordpress-deployment.yaml
>> "@ >> kustomization.yaml
>>
PS C:\Users\Admin> _
```

Step 4: Apply the Kustomization

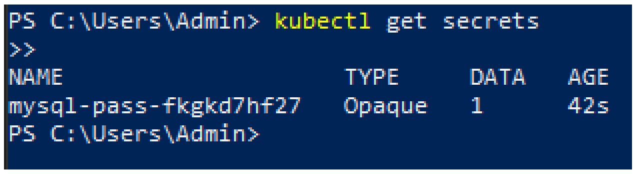
Deploy all resources to the cluster:



```
PS C:\Users\Admin> kubectl apply -k .\
>>
secret/mysql-pass-fkgkd7hf27 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\Admin>
```

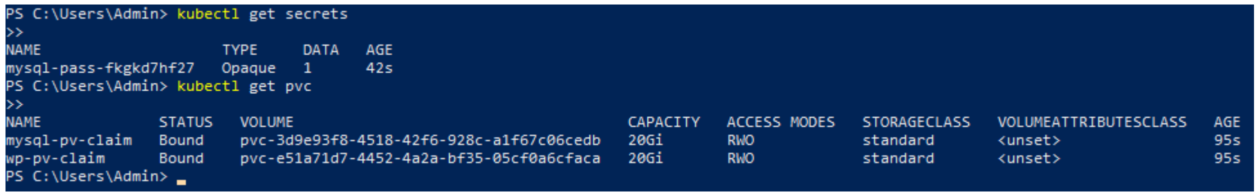
Step 5: Verify Resources

1. Check that the Secret exists:



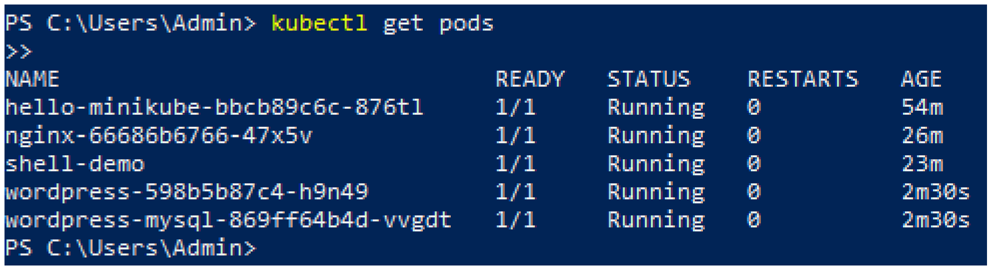
```
PS C:\Users\Admin> kubectl get secrets
>>
NAME                    TYPE     DATA   AGE
mysql-pass-fkgkd7hf27   Opaque   1      42s
PS C:\Users\Admin>
```

2. Check that PersistentVolumeClaims are bound:



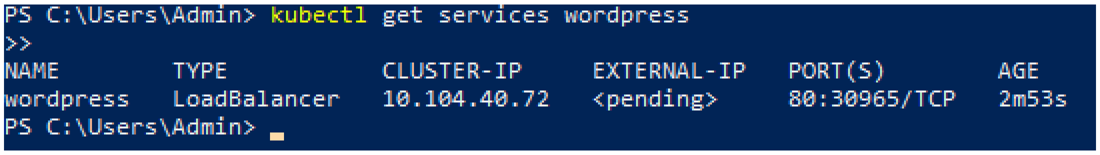
```
PS C:\Users\Admin> kubectl get secrets
>>
NAME                    TYPE     DATA   AGE
mysql-pass-fkgkd7hf27   Opaque   1      42s
PS C:\Users\Admin> kubectl get pvc
>>
NAME             STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mysql-pv-claim   Bound    pvc-3d9e93f8-4518-42f6-928c-a1f67c06cedb   20Gi       RWO            standard       <unset>                 95s
wp-pv-claim      Bound    pvc-e51a71d7-4452-4a2a-bf35-05cf0a6cfaca   20Gi       RWO            standard       <unset>                 95s
PS C:\Users\Admin> _
```

3. Check that Pods are running:



```
PS C:\Users\Admin> kubectl get pods
>>
NAME                               READY   STATUS    RESTARTS   AGE
hello-minikube-bbcb89c6c-876tl     1/1     Running   0          54m
nginx-66686b6766-47x5v             1/1     Running   0          26m
shell-demo                         1/1     Running   0          23m
wordpress-598b5b87c4-h9n49         1/1     Running   0          2m30s
wordpress-mysql-869ff64b4d-vvgdt   1/1     Running   0          2m30s
PS C:\Users\Admin>
```

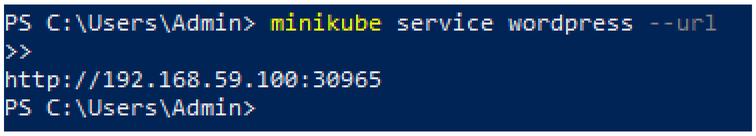
4. Check that the WordPress Service exists:



```
PS C:\Users\Admin> kubectl get services wordpress
>>
NAME         TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)        AGE
wordpress    LoadBalancer   10.104.40.72   <pending>      80:30965/TCP   2m53s
PS C:\Users\Admin> _
```

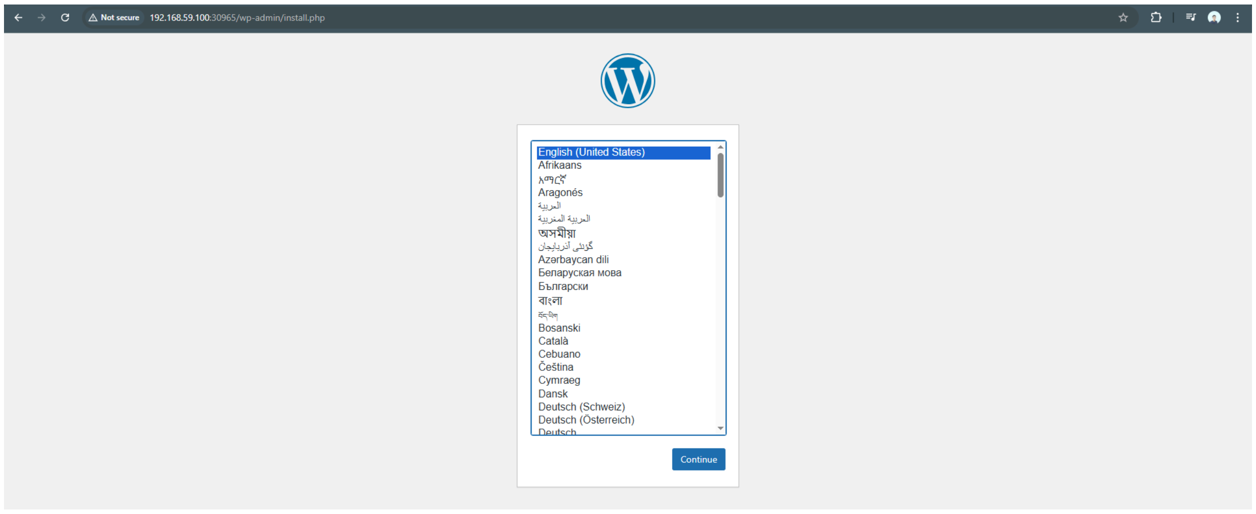
5. Get the WordPress URL:

   This returns a URL like:



```
PS C:\Users\Admin> minikube service wordpress --url
>>
http://192.168.59.100:30965
PS C:\Users\Admin>
```

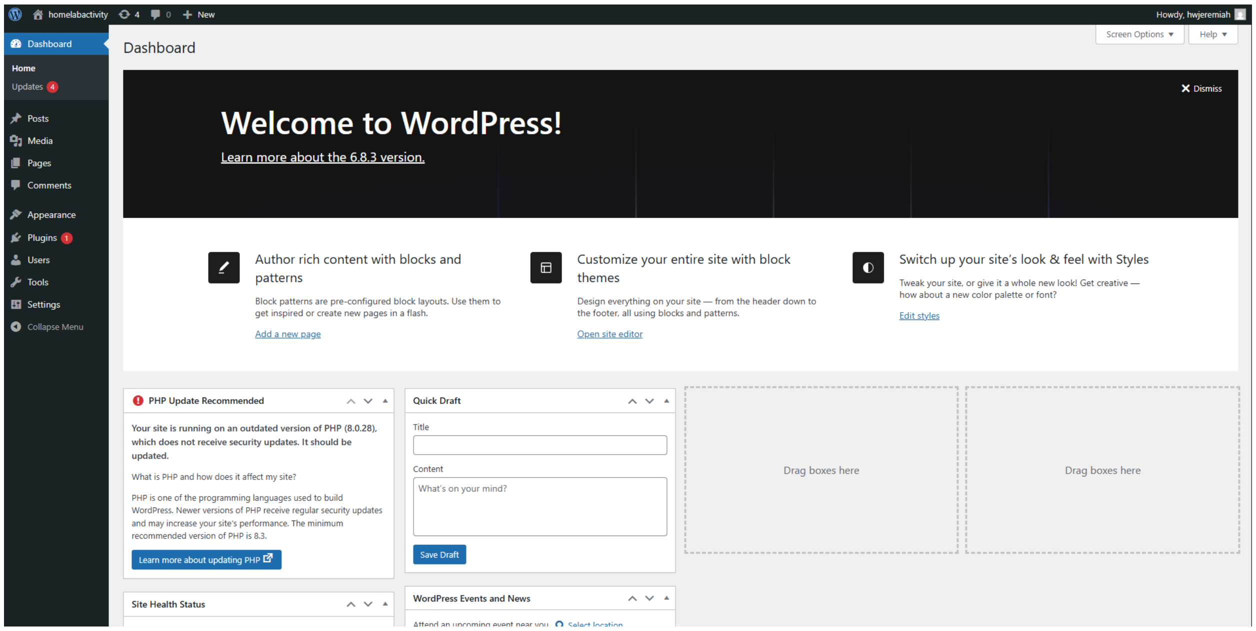
Open this URL in your browser to access the WordPress setup page.



Step 6: Complete WordPress Setup

Create your WordPress admin username and password.

Finish the installation to secure your site.



Dashboard

Welcome to WordPress!

Learn more about the 6.8.3 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

PHP Update Recommended

Your site is running on an outdated version of PHP (8.0.28), which does not receive security updates. It should be updated.

What is PHP and how does it affect my site?

PHP is one of the programming languages used to build WordPress. Newer versions of PHP receive regular security updates and may increase your site's performance. The minimum recommended version of PHP is 8.3.

Quick Draft

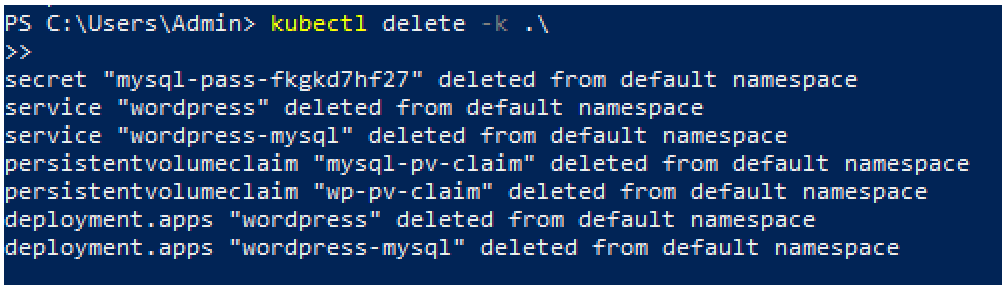
Site Health Status

WordPress Events and News

Attend an upcoming event near you.

Step 7: Clean Up

1. After finishing, delete all resources created by the kustomization:



```
PS C:\Users\Admin> kubectl delete -k .\
>>
secret "mysql-pass-fkgkd7hf27" 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
```

2. Stop minikube (if you want to keep files for later)
3. Delete minikube (Delete the virtual machine and its data)