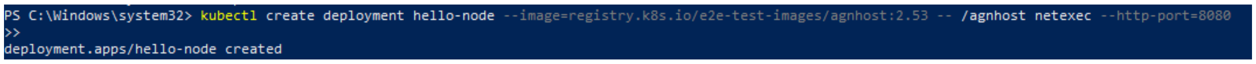
Villena, Lans Clarence S.
IV - ACSAD

Assignment# 5 - Kubernetes Home Lab Activity

Part 1 - Hello Minikube Activity

1.1 Creating First Deployment:

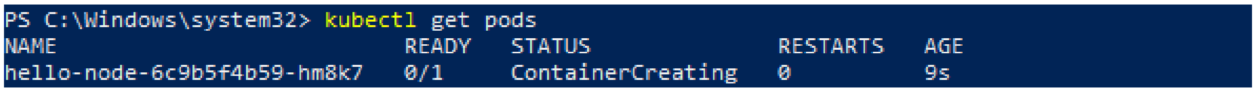
kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.53 -- /agnhost netexec --http-port=8080



```
PS C:\Windows\system32> kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.53 -- /agnhost netexec --http-port=8080
>>
deployment.apps/hello-node created
```

1.2 Check Pods:

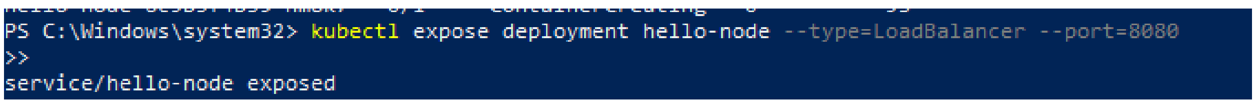
kubectl get pods



```
PS C:\Windows\system32> kubectl get pods
NAME                          READY   STATUS              RESTARTS   AGE
hello-node-6c9b5f4b59-hm8k7   0/1     ContainerCreating   0          9s
```

1.3 Expose Service:

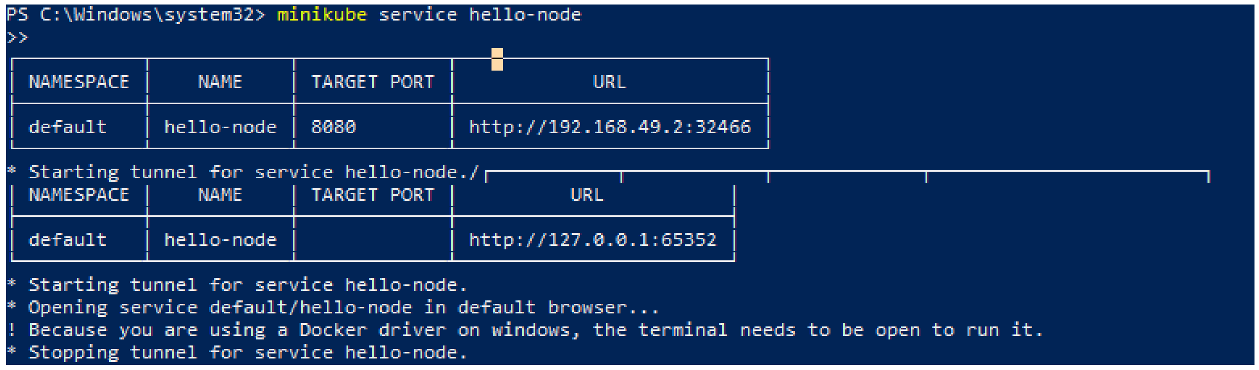
kubectl expose deployment hello-node --type=LoadBalancer --port=8080



```
PS C:\Windows\system32> kubectl expose deployment hello-node --type=LoadBalancer --port=8080
>>
service/hello-node exposed
```

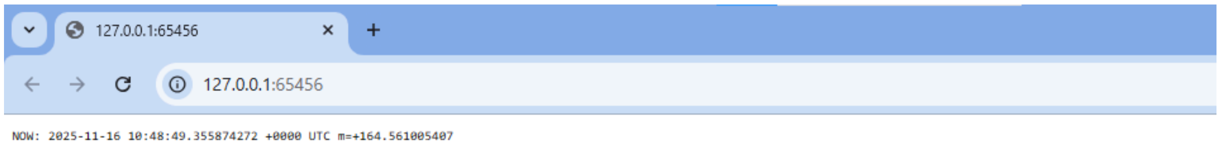
1.4 Open App in Browser:

minikube service hello-node



```
PS C:\Windows\system32> minikube service hello-node
>>
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | hello-node | 8080        | http://192.168.49.2:32466 |
|-----------|------------|-------------|---------------------------|
* Starting tunnel for service hello-node./
| NAMESPACE |    NAME    | TARGET PORT |           URL          |
|-----------|------------|-------------|------------------------|
| default   | hello-node |             | http://127.0.0.1:65352 |
|-----------|------------|-------------|------------------------|
* 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.
* Stopping tunnel for service hello-node.
```

Browser Screenshot:

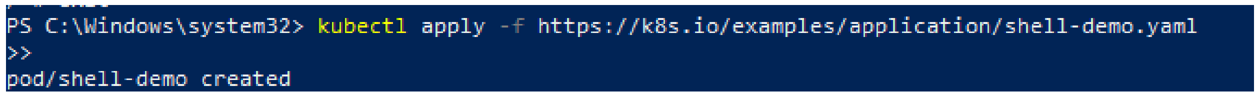


```
NOW: 2025-11-16 10:48:49.355874272 +0000 UTC m=+164.561005407
```

Part 2 - Get a Shell to a Running Container

2.1 Create the Pod:

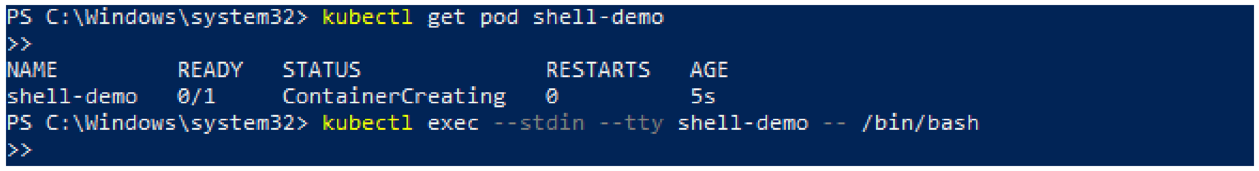
kubectl apply -f https://k8s.io/examples/application/shell-demo.yaml



```
PS C:\Windows\system32> kubectl apply -f https://k8s.io/examples/application/shell-demo.yaml
>>
pod/shell-demo created
```

2.2 Verify if Pod is Running:

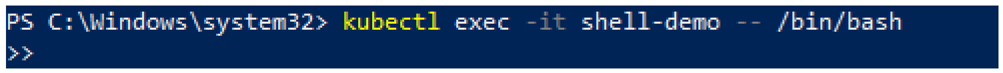
kubectl get pod shell-demo



```
PS C:\Windows\system32> kubectl get pod shell-demo
>>
NAME         READY   STATUS              RESTARTS   AGE
shell-demo   0/1     ContainerCreating   0          5s
PS C:\Windows\system32> kubectl exec --stdin --tty shell-demo -- /bin/bash
>>
```

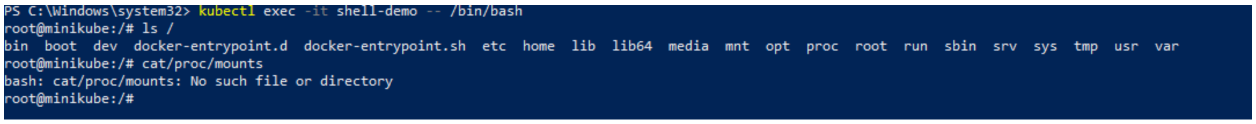
2.3 Get a shell inside a container:

kubectl exec -it shell-demo -- /bin/bash



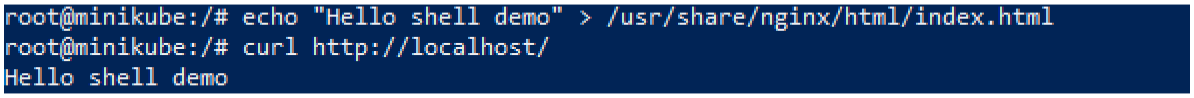
```
PS C:\Windows\system32> kubectl exec -it shell-demo -- /bin/bash
>>
```

2.4 Inside shell, run some commands:
ls /
cat/proc/mounts



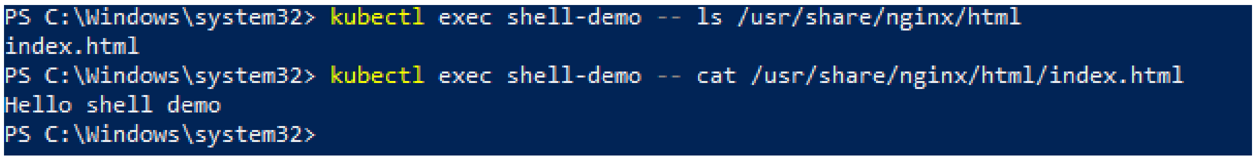
```
PS C:\Windows\system32> kubectl exec -it shell-demo -- /bin/bash
root@minikube:/# ls /
bin  boot  dev  docker-entrypoint.d  docker-entrypoint.sh  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@minikube:/# cat/proc/mounts
bash: cat/proc/mounts: No such file or directory
root@minikube:/#
```

2.5 Writing root page page for nginx:
echo "Hello shell demo" > /usr/share/nginx/html/index.html
curl http://localhost/
exit



```
root@minikube:/# echo "Hello shell demo" > /usr/share/nginx/html/index.html
root@minikube:/# curl http://localhost/
Hello shell demo
```

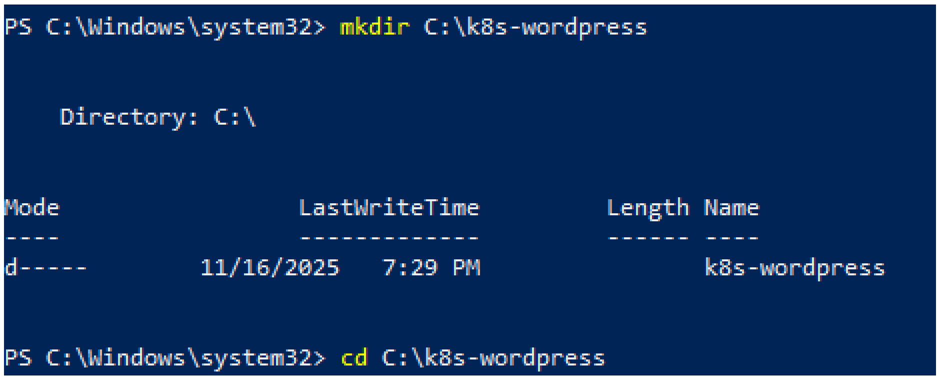
2.7 Running individual commands in a container:
kubectl exec shell-demo -- ls /usr/share/nginx/html
kubectl exec shell-demo -- cat /usr/share/nginx/html/index.html



```
PS C:\Windows\system32> kubectl exec shell-demo -- ls /usr/share/nginx/html
index.html
PS C:\Windows\system32> kubectl exec shell-demo -- cat /usr/share/nginx/html/index.html
Hello shell demo
PS C:\Windows\system32>
```

Part 3 - Deploying Wordpress and MySQL with Persistent Volumes

3.1 Create a working directory:
mkdir C:\k8s-wordpress
cd C:\k8s-wordpress



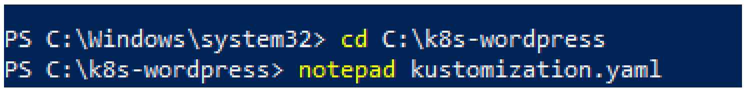
```
PS C:\Windows\system32> mkdir C:\k8s-wordpress


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/16/2025   7:29 PM                k8s-wordpress


PS C:\Windows\system32> cd C:\k8s-wordpress
```

3.2 Create kustomization.yaml with a Secret for MySQL:
notepad kustomization.yaml

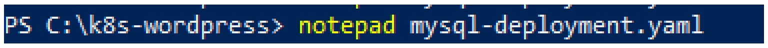


```
PS C:\Windows\system32> cd C:\k8s-wordpress
PS C:\k8s-wordpress> notepad kustomization.yaml
```

3.3 In the new file paste:

```
secretGenerator:
- name: mysql-pass
  literals:
  - password=YOUR_PASSWORD (create a strong password)
resources:
- mysql-deployment.yaml
- wordpress-deployment.yaml
```

3.4 MySQL Deployment - Create file name mysql-deployment.yaml:
notepad mysql-deployment.yaml



```
PS C:\k8s-wordpress> notepad mysql-deployment.yaml
```

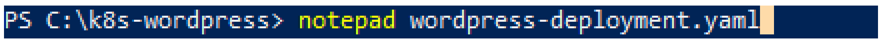
Input the content from tutorial:

```
apiVersion: v1
kind: Service
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  ports:
    - port: 3306
  selector:
    app: wordpress
    tier: mysql
  clusterIP: None
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
    spec:
      containers:
      - image: mysql:8.0
        name: mysql
        env:
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-pass
              key: password
        - name: MYSQL_DATABASE
          value: wordpress
        - name: MYSQL_USER
          value: wordpress
        - name: MYSQL_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-pass
              key: password
        ports:
        - containerPort: 3306
          name: mysql
        volumeMounts:
```

```
        - name: mysql-persistent-storage
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-persistent-storage
        persistentVolumeClaim:
          claimName: mysql-pv-claim
```

3.5 Create a file named wordpress-deployment.yaml in the same folder:

notepad wordpress-deployment.yaml



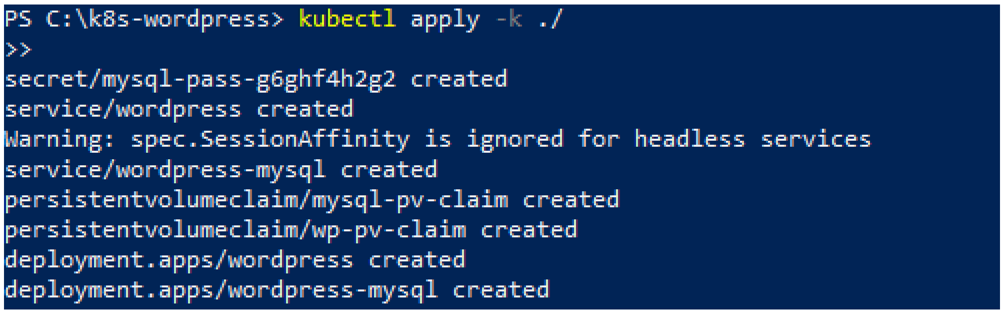
```
PS C:\k8s-wordpress> notepad wordpress-deployment.yaml
```

Input the content from tutorial:

```
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
    - port: 80
  selector:
    app: wordpress
    tier: frontend
  type: LoadBalancer
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: frontend
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: frontend
    spec:
      containers:
      - image: wordpress:6.2.1-apache
        name: wordpress
        env:
        - name: WORDPRESS_DB_HOST
          value: wordpress-mysql
        - name: WORDPRESS_DB_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-pass
              key: password
        - name: WORDPRESS_DB_USER
          value: wordpress
        ports:
        - containerPort: 80
          name: wordpress
        volumeMounts:
        - name: wordpress-persistent-storage
          mountPath: /var/www/html
      volumes:
      - name: wordpress-persistent-storage
        persistentVolumeClaim:
          claimName: wp-pv-claim
```

3.6 Apply everything using kubectl:

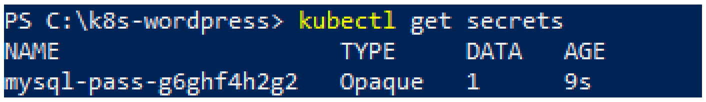
PS C:\k8s-wordpress> kubectl apply -k ./



```
PS C:\k8s-wordpress> kubectl apply -k ./
>>
secret/mysql-pass-g6ghf4h2g2 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
```

3.7 Verify Kubernetes resources are created:

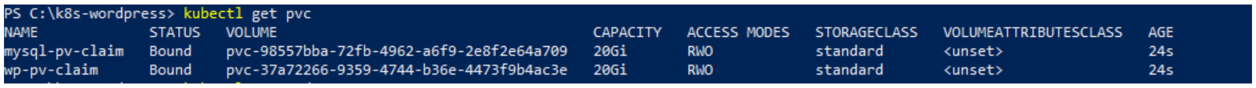
kubectl get secrets



```
PS C:\k8s-wordpress> kubectl get secrets
NAME                    TYPE      DATA   AGE
mysql-pass-g6ghf4h2g2   Opaque    1      9s
```

3.8 Check PersistentVolumeClaims (status must become 'bound'):

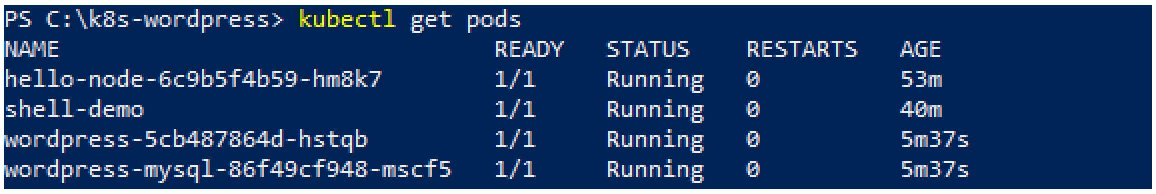
kubectl get pvc



```
PS C:\k8s-wordpress> kubectl get pvc
NAME             STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mysql-pv-claim   Bound    pvc-98557bba-72fb-4962-a6f9-2e8f2e64a709   20Gi       RWO            standard       <unset>                 24s
wp-pv-claim      Bound    pvc-37a72266-9359-4744-b36e-4473f9b4ac3e   20Gi       RWO            standard       <unset>                 24s
```

3.9 Check Pods (Both must have status of 'Running'):

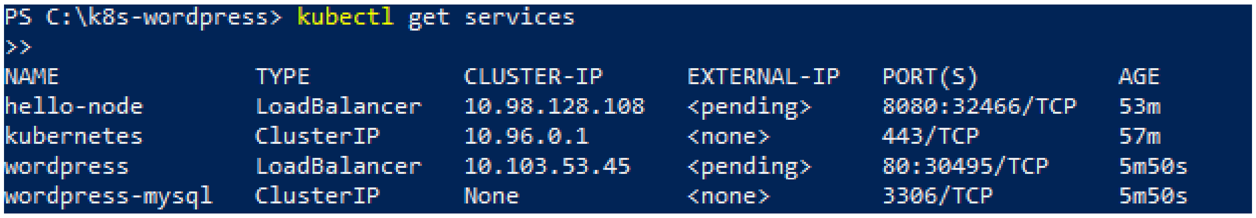
kubectl get pods



```
PS C:\k8s-wordpress> kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
hello-node-6c9b5f4b59-hm8k7        1/1     Running   0          53m
shell-demo                         1/1     Running   0          40m
wordpress-5cb487864d-hstqb         1/1     Running   0          5m37s
wordpress-mysql-86f49cf948-mscf5   1/1     Running   0          5m37s
```

3.10 Check Services:

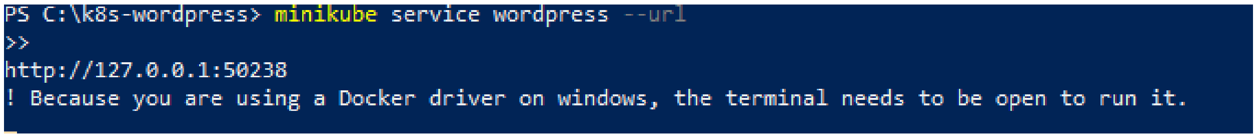
kubectl get services



```
PS C:\k8s-wordpress> kubectl get services
>>
NAME              TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-node        LoadBalancer   10.98.128.108   <pending>     8080:32466/TCP   53m
kubernetes        ClusterIP      10.96.0.1       <none>        443/TCP          57m
wordpress         LoadBalancer   10.103.53.45    <pending>     80:30495/TCP     5m50s
wordpress-mysql   ClusterIP      None            <none>        3306/TCP         5m50s
```

3.11 Get the URL for WordPress (on Minikube):

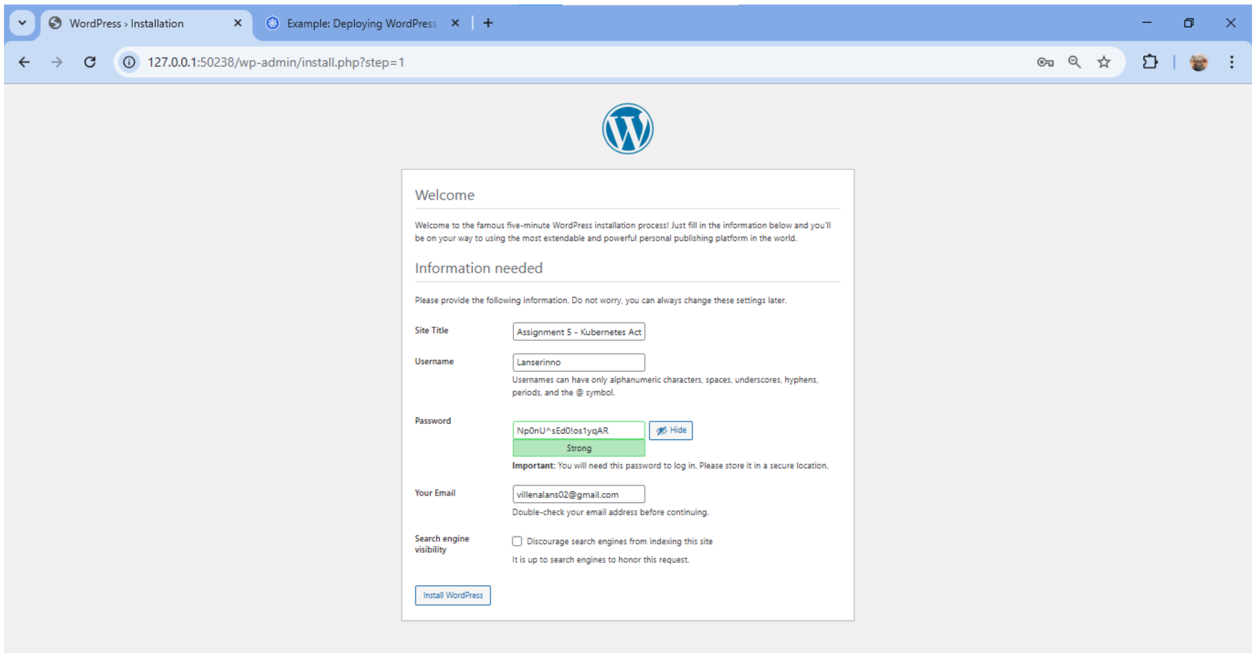
minikube service wordpress --url



```
PS C:\k8s-wordpress> minikube service wordpress --url
>>
http://127.0.0.1:50238
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

3.12 Complete the WordPress Setup from browser:

Input any information here:


WordPress › Installation
Example: Deploying WordPress
127.0.0.1:50238/wp-admin/install.php?step=1
Hide
Strong
Install WordPress

Welcome

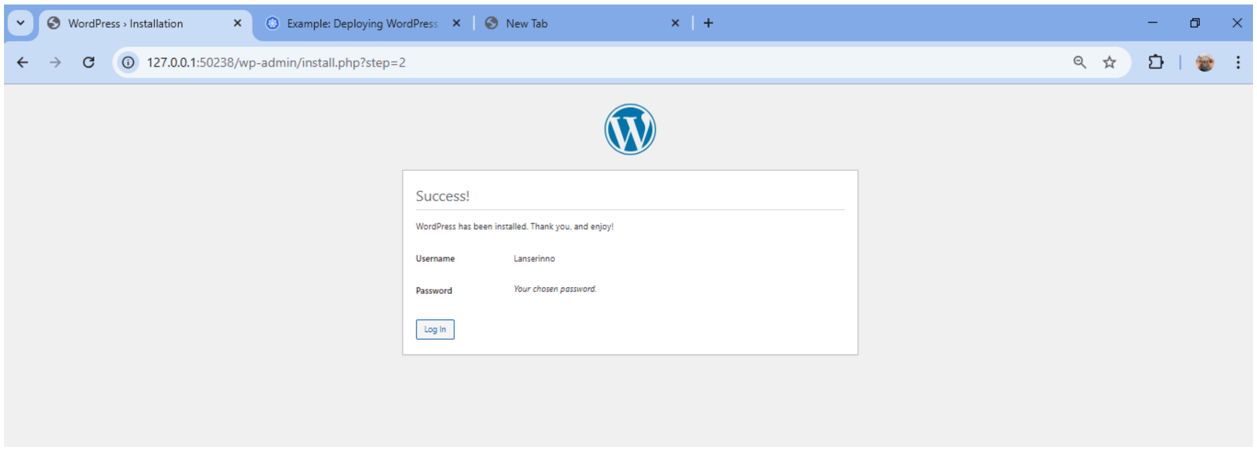
Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

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

Site Title	Assignment 5 - Kubernetes Act
Username	Lanserinno Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password	Np0nU^sEd0!os1yqAR **Important:** You will need this password to log in. Please store it in a secure location.
Your Email	villenalans02@gmail.com 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.

Press login:

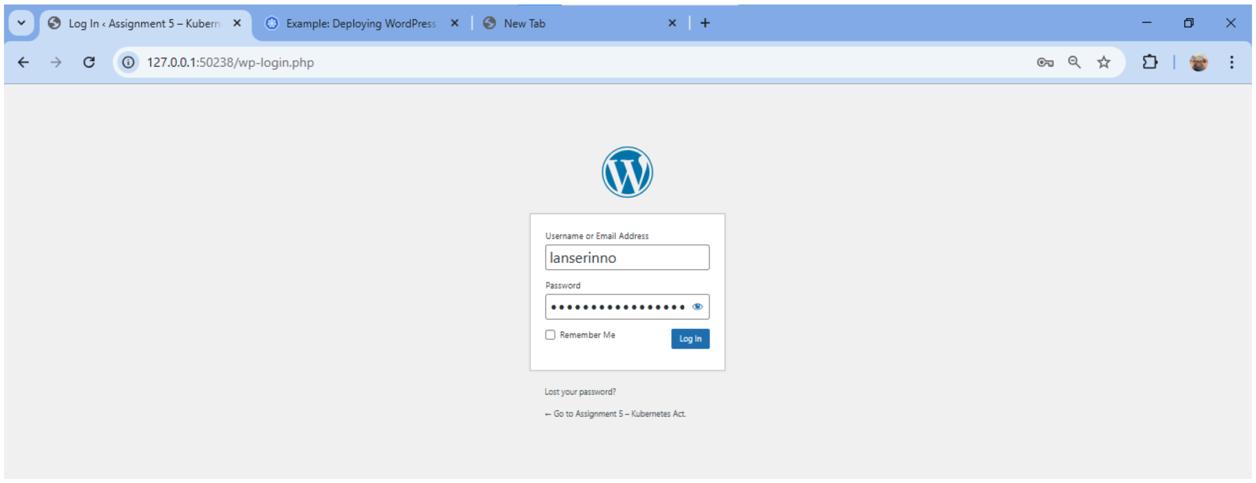

WordPress › Installation
Example: Deploying WordPress
New Tab
127.0.0.1:50238/wp-admin/install.php?step=2
Log In

Success!

WordPress has been installed. Thank you, and enjoy!

Username	Lanserinno
Password	*Your chosen password.*

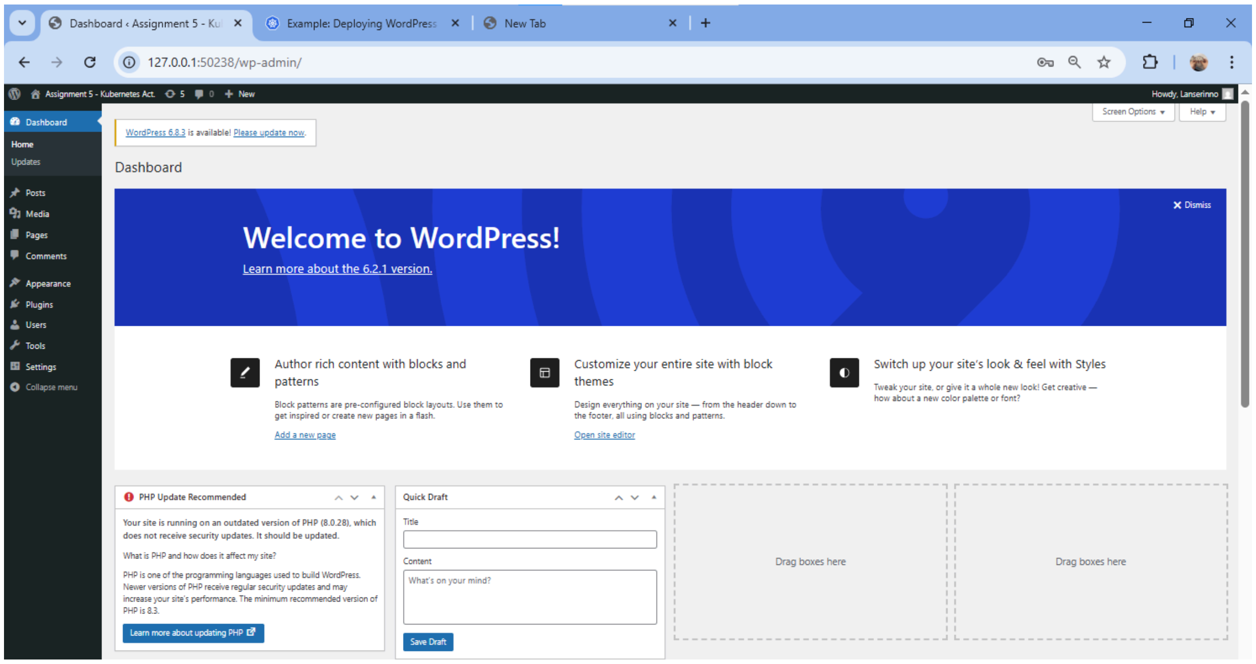
Enter the email and password you inputted on sign-up:


Log In ‹ Assignment 5 – Kubern
Example: Deploying WordPress
New Tab
Username or Email Address
Password
Remember Me
Log In
Lost your password?
← Go to Assignment 5 – Kubernetes Act.

127.0.0.1:50238/wp-login.php

Username or Email Address: lanserinno

Wordpress dashboard:



127.0.0.1:50238/wp-admin/

WordPress 6.8.3 is available! Please update now.

Dashboard

Welcome to WordPress!

Learn more about the 6.2.1 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?

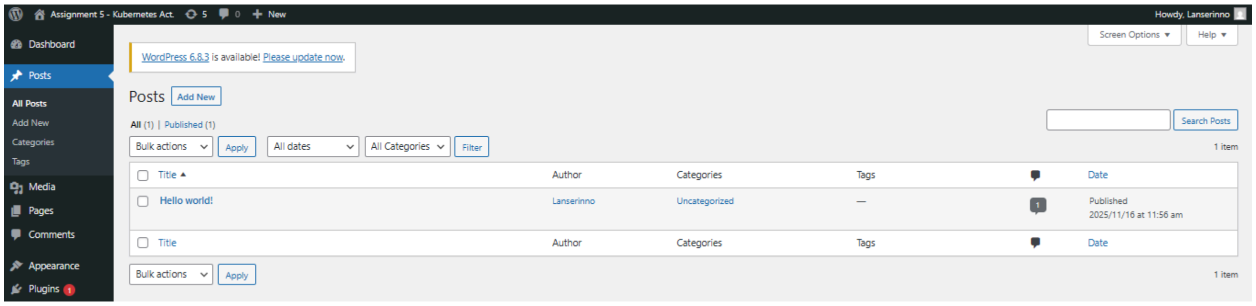
PHP is one of the programming languages used to build WordPress. Newer versions of PHP receive regular security updates and may increase your site's performance. The minimum recommended version of PHP is 8.3.

Quick Draft

Title

Content

What's on your mind?



WordPress 6.8.3 is available! Please update now.

Posts

All (1) | Published (1)

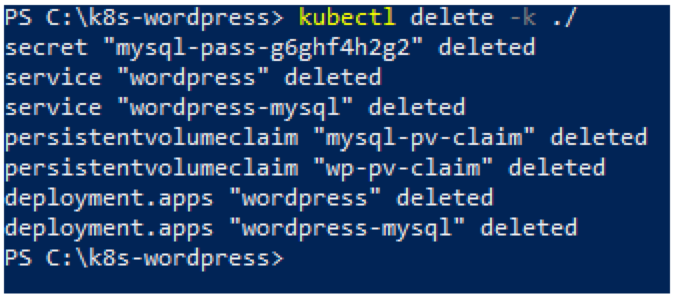
1 item

Title	Author	Categories	Tags		Date
Hello world!	Lanserinno	Uncategorized	—	1	Published 2025/11/16 at 11:56 am
Title	Author	Categories	Tags		Date

1 item

3.13 Clean up resources when done:

kubectl delete -k ./



```
PS C:\k8s-wordpress> kubectl delete -k ./
secret "mysql-pass-g6ghf4h2g2" deleted
service "wordpress" deleted
service "wordpress-mysql" deleted
persistentvolumeclaim "mysql-pv-claim" deleted
persistentvolumeclaim "wp-pv-claim" deleted
deployment.apps "wordpress" deleted
deployment.apps "wordpress-mysql" deleted
PS C:\k8s-wordpress>
```