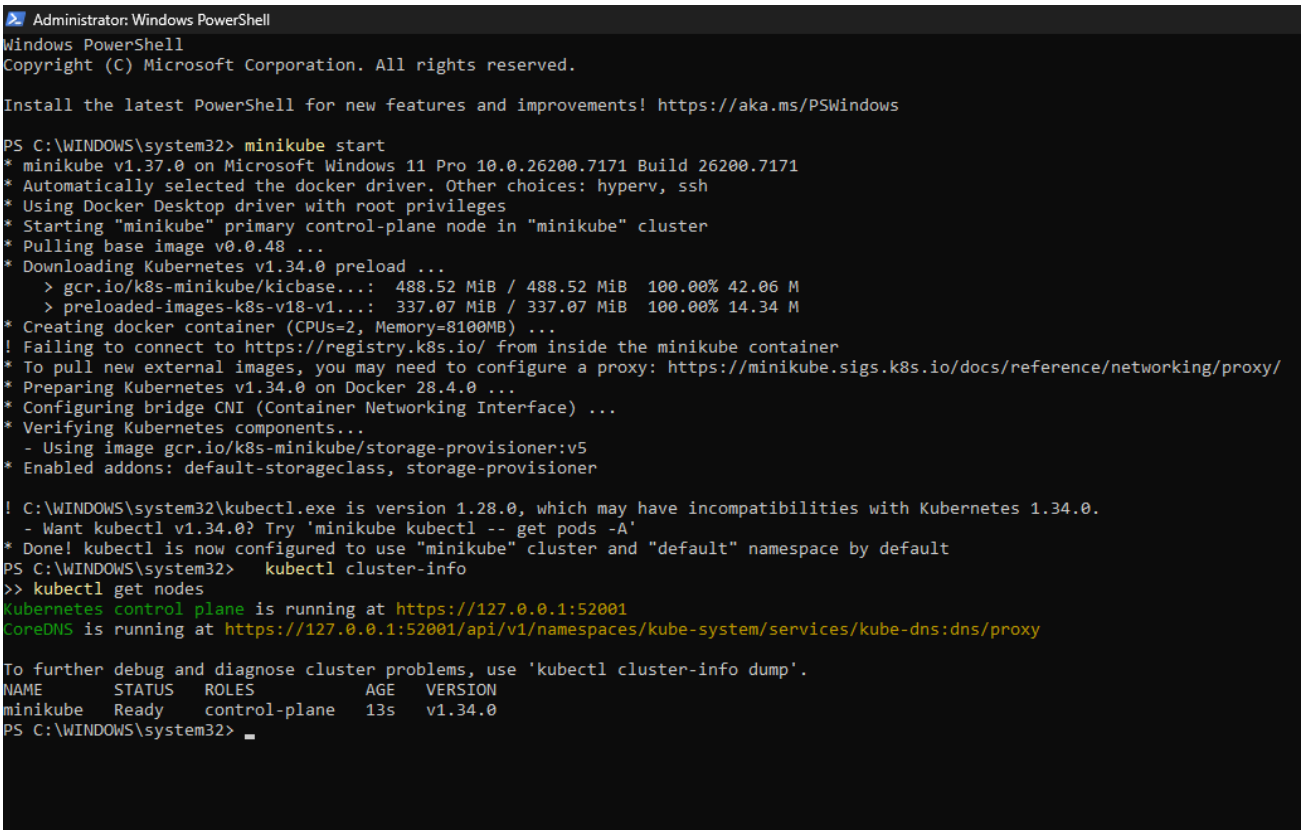
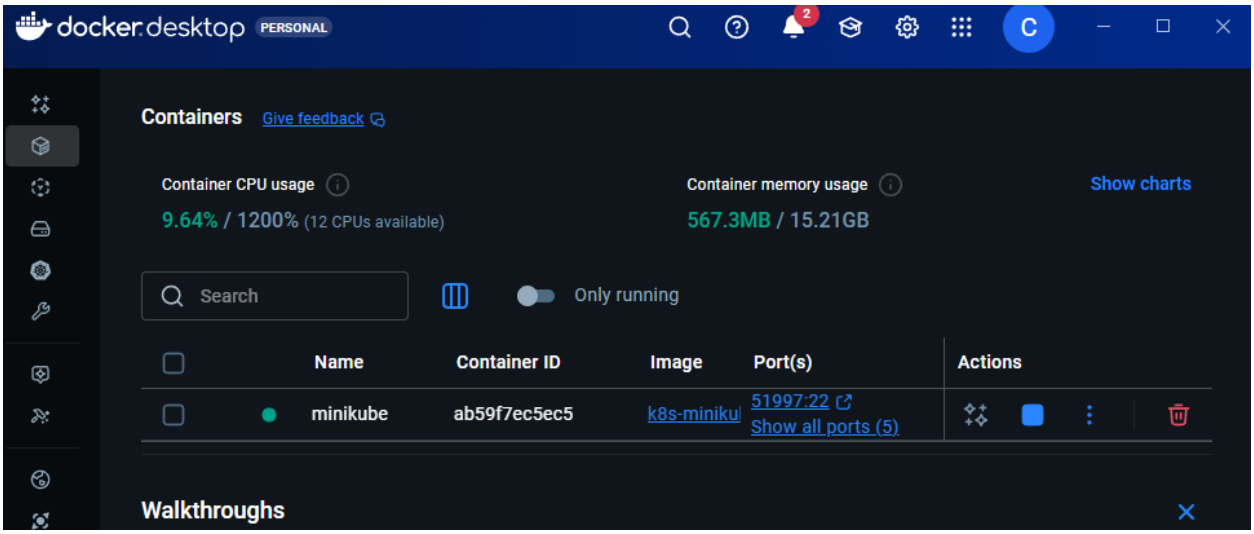
CABUS, CLEMENT HAROLD MIGUEL

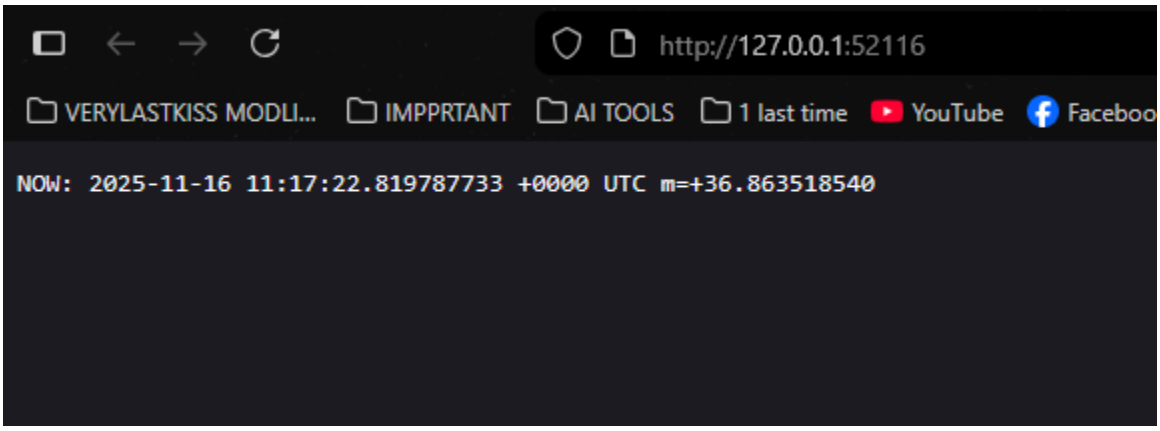
IV-ACSAD

- Hello Minikube:

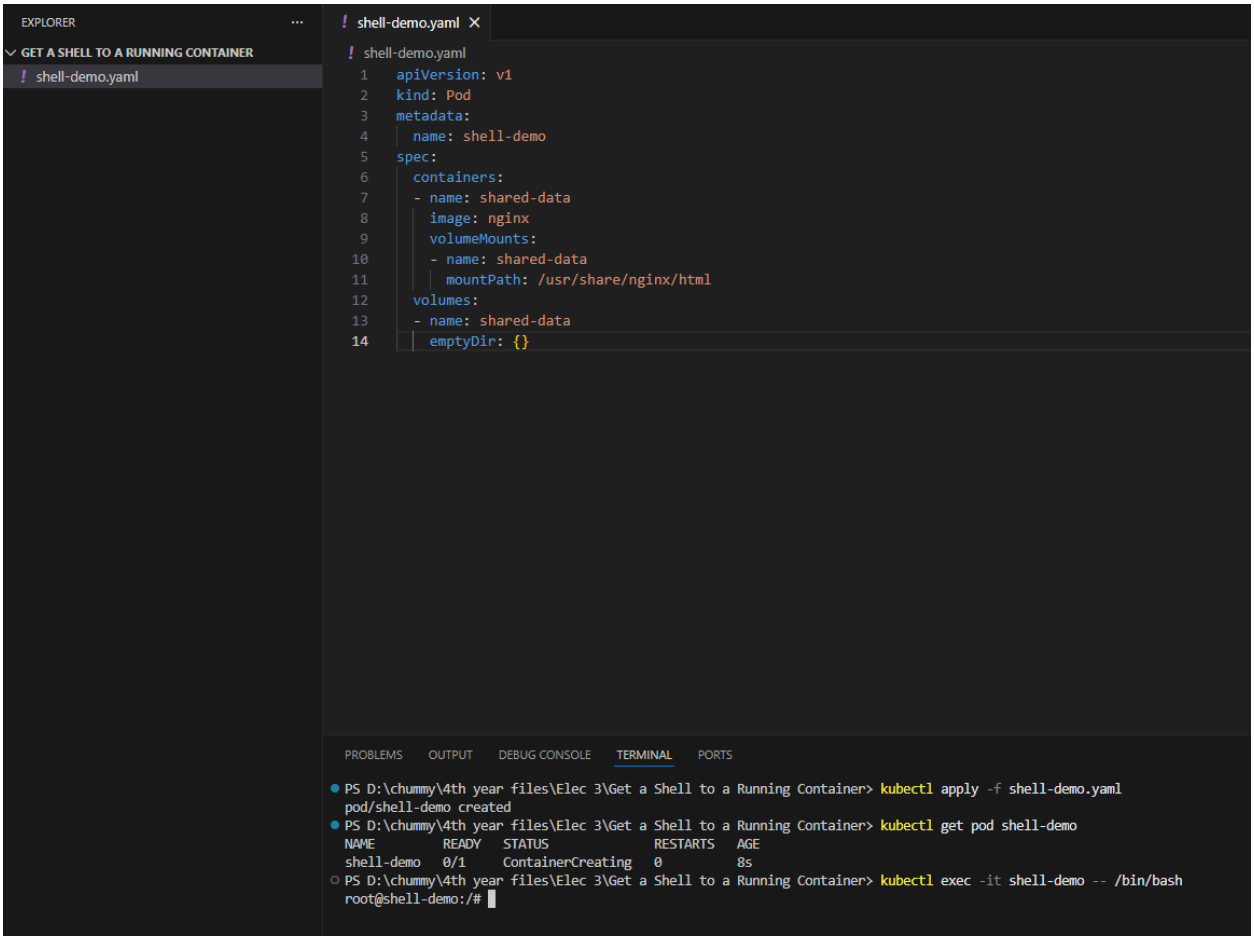


```
PS C:\WINDOWS\system32> kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080
deployment.apps/hello-node created
PS C:\WINDOWS\system32> kubectl get deployments
>> kubectl get pods
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   1/1     1            1           14s
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-bf5bf8ccf-dbzs4    1/1     Running   0          14s
PS C:\WINDOWS\system32> kubectl expose deployment hello-node --type=LoadBalancer --port=8080
service/hello-node exposed
PS C:\WINDOWS\system32> kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-node   LoadBalancer   10.106.40.100   <pending>     8080:31944/TCP   4s
kubernetes   ClusterIP      10.96.0.1       <none>        443/TCP          64s
PS C:\WINDOWS\system32> minikube service hello-node
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | hello-node | 8080        | http://192.168.49.2:31944 |
|-----------|------------|-------------|---------------------------|
* Starting tunnel for service hello-node.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | hello-node |             | http://127.0.0.1:52116 |
|-----------|------------|-------------|------------------------|
* 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.
```



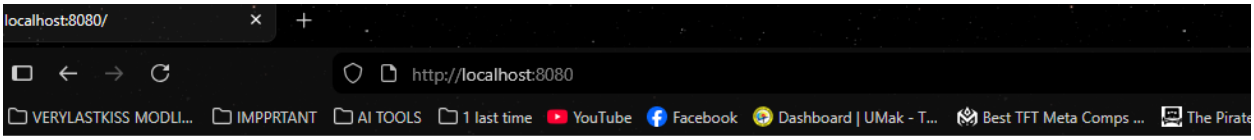


- Get a Shell to a Runing Container:



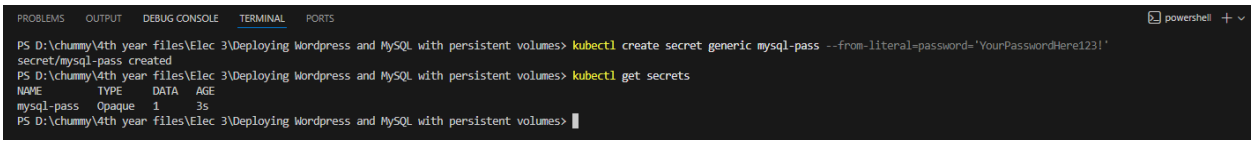
```
root@shell-demo:/# pwd
/
root@shell-demo:/# ls -la
total 72
drwxr-xr-x   1 root root 4096 Nov 16 11:21 .
drwxr-xr-x   1 root root 4096 Nov 16 11:21 ..
-rwxr-xr-x   1 root root    0 Nov 16 11:21 .dockerenv
lrwxrwxrwx   1 root root    7 Aug 24 16:20 bin -> usr/bin
drwxr-xr-x   2 root root 4096 Aug 24 16:20 boot
drwxr-xr-x   5 root root  360 Nov 16 11:21 dev
drwxr-xr-x   1 root root 4096 Nov  4 04:06 docker-entrypoint.d
-rwxr-xr-x   1 root root 1620 Nov  4 04:05 docker-entrypoint.sh
drwxr-xr-x   1 root root 4096 Nov 16 11:21 etc
drwxr-xr-x   2 root root 4096 Aug 24 16:20 home
lrwxrwxrwx   1 root root    7 Aug 24 16:20 lib -> usr/lib
lrwxrwxrwx   1 root root    9 Aug 24 16:20 lib64 -> usr/lib64
drwxr-xr-x   2 root root 4096 Nov  3 20:44 media
drwxr-xr-x   2 root root 4096 Nov  3 20:44 mnt
drwxr-xr-x   2 root root 4096 Nov  3 20:44 opt
dr-xr-xr-x 297 root root    0 Nov 16 11:21 proc
drwx------   2 root root 4096 Nov  3 20:44 root
drwxr-xr-x   1 root root 4096 Nov 16 11:21 run
lrwxrwxrwx   1 root root    8 Aug 24 16:20 sbin -> usr/sbin
drwxr-xr-x   2 root root 4096 Nov  3 20:44 srv
dr-xr-xr-x  13 root root    0 Nov 16 11:21 sys
drwxrwxrwt   2 root root 4096 Nov  3 20:44 tmp
drwxr-xr-x   1 root root 4096 Nov  3 20:44 usr
drwxr-xr-x   1 root root 4096 Nov  3 20:44 var
root@shell-demo:/# cd /usr/share/nginx/html
root@shell-demo:/usr/share/nginx/html# echo "Hello from inside the container!" > index.html
root@shell-demo:/usr/share/nginx/html# cat index.html
Hello from inside the container!
root@shell-demo:/usr/share/nginx/html# exit
exit
PS D:\chummy\4th year files\Elec 3\Get a Shell to a Running Container>
```

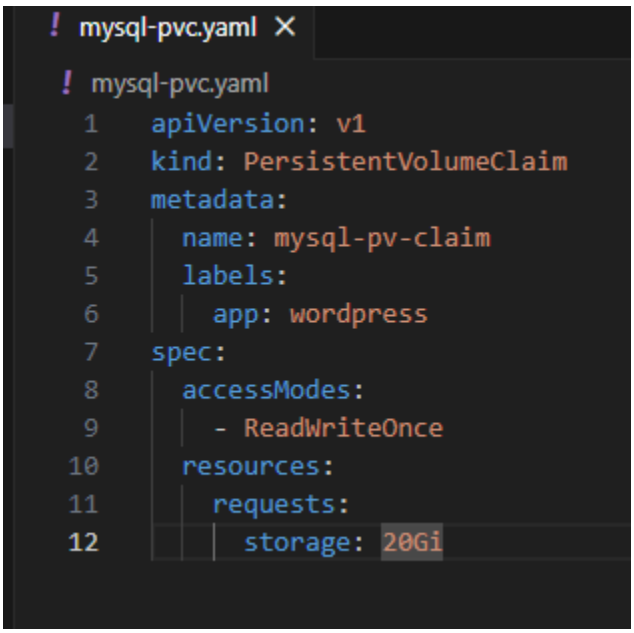
```
PS D:\chummy\4th year files\Elec 3\Get a Shell to a Running Container> kubectl exec shell-demo -- cat /usr/share/nginx/html/index.html
Hello from inside the container!
PS D:\chummy\4th year files\Elec 3\Get a Shell to a Running Container> # Forward port to access nginx
>> kubectl port-forward shell-demo 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
Handling connection for 8080
```

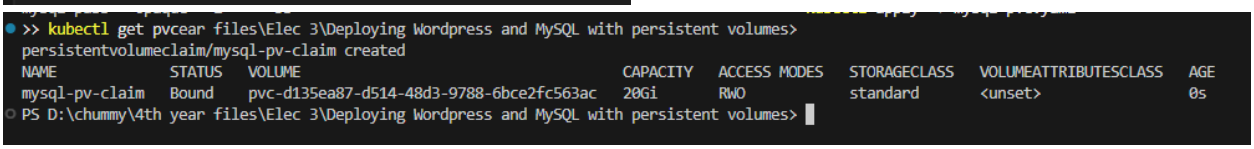


Hello from inside the container!

- Deploying Wordpress and MySQL with persistent volumes:







mysql-pvc.yaml | wordpress-pvc.yaml

wordpress-pvc.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
```

```
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl apply -f wordpress-pvc.yaml
>> kubectl get pvc
persistentvolumeclaim/wp-pv-claim created
NAME             STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mysql-pv-claim   Bound    pvc-d135ea87-d514-48d3-9788-6bce2fc563ac   20Gi       RWO            standard       <unset>                 26s
wp-pv-claim      Bound    pvc-3fd054dc-7056-49f1-ae32-ca6d40dd89f6   20Gi       RWO            standard       <unset>                 0s
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes>
```

mysql-pvc.yaml | wordpress-pvc.yaml | mysql-deployment.yaml

mysql-deployment.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  ports:
    - port: 3306
  selector:
    app: wordpress
    tier: mysql
  clusterIP: None
---
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
```

mysql-pvc.yaml | wordpress-pvc.yaml | mysql-deployment.yaml

mysql-deployment.yaml

```yaml
      - 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
```

```
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl apply -f mysql-deployment.yaml
Warning: spec.SessionAffinity is ignored for headless services
service/wordpress-mysql created
deployment.apps/wordpress-mysql created
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes>
```

```
deployment.apps/wordpress-mysql created
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl get pods
NAME                               READY   STATUS              RESTARTS   AGE
shell-demo                         1/1     Running             0          10m
two-containers                     2/2     Running             0          5m14s
wordpress-mysql-5ccb49cfb-vvl9b    0/1     ContainerCreating   0          13s
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl logs deployment/wordpress-mysql
Error from server (BadRequest): container "mysql" in pod "wordpress-mysql-5ccb49cfb-vvl9b" is waiting to start: ContainerCreating
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl get pods
NAME                               READY   STATUS              RESTARTS   AGE
shell-demo                         1/1     Running             0          10m
two-containers                     2/2     Running             0          5m28s
wordpress-mysql-5ccb49cfb-vvl9b    0/1     ContainerCreating   0          27s
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
shell-demo                         1/1     Running   0          10m
two-containers                     2/2     Running   0          5m29s
wordpress-mysql-5ccb49cfb-vvl9b    1/1     Running   0          28s
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
shell-demo                         1/1     Running   0          10m
two-containers                     2/2     Running   0          5m30s
wordpress-mysql-5ccb49cfb-vvl9b    1/1     Running   0          29s
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl logs deployment/wordpress-mysql
2025-11-16 11:31:49+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.44-1.el9 started.
2025-11-16 11:31:49+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2025-11-16 11:31:49+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.44-1.el9 started.
2025-11-16 11:31:49+00:00 [Note] [Entrypoint]: Initializing database files
2025-11-16T11:31:49.584866Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
2025-11-16T11:31:49.584972Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.44) initializing of server in progress as process 81
2025-11-16T11:31:49.592210Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2025-11-16T11:31:50.225400Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2025-11-16T11:31:51.500725Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider switching off the --initialize-insecure option.
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes>
```

mysql-pvc.yaml | wordpress-pvc.yaml | mysql-deployment.yaml | wordpress-deployment.yaml

wordpress-deployment.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
    - port: 80
  selector:
    app: wordpress
    tier: frontend
  type: LoadBalancer
---
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
```

mysql-pvc.yaml | wordpress-pvc.yaml | mysql-deployment.yaml | wordpress-deployment.yaml

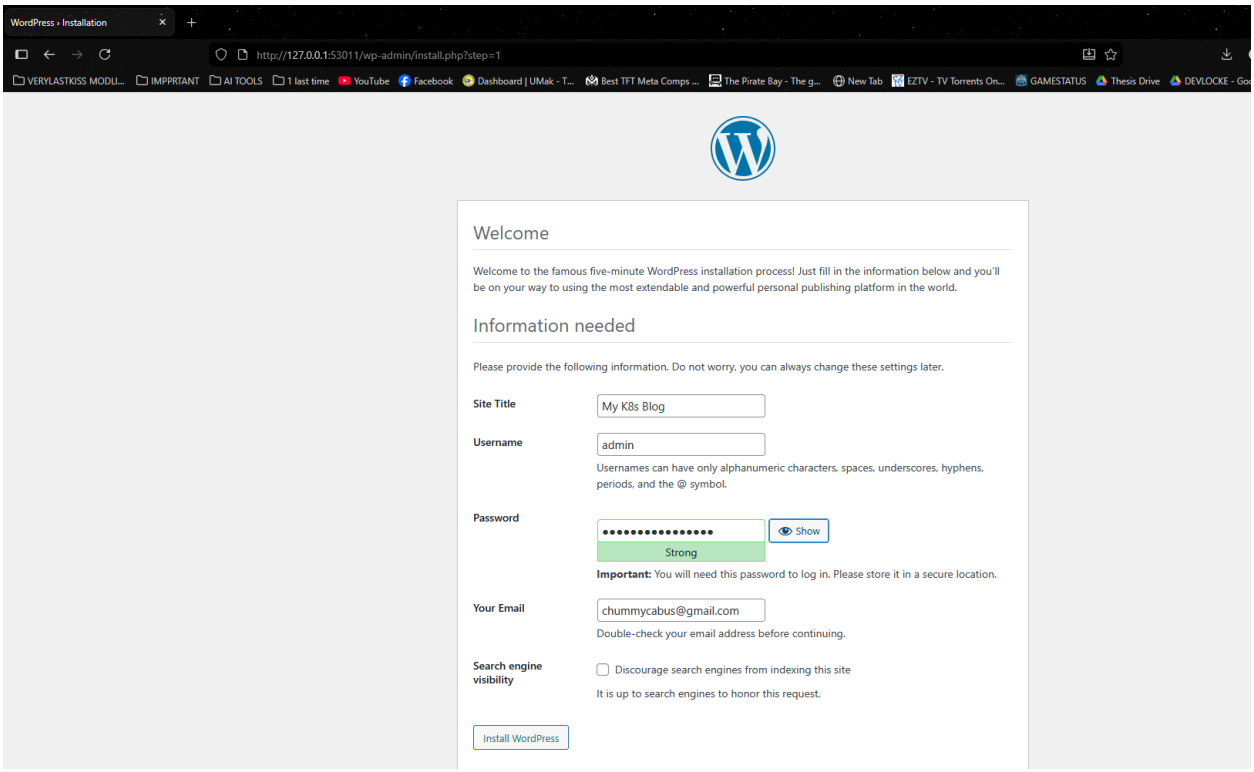
wordpress-deployment.yaml

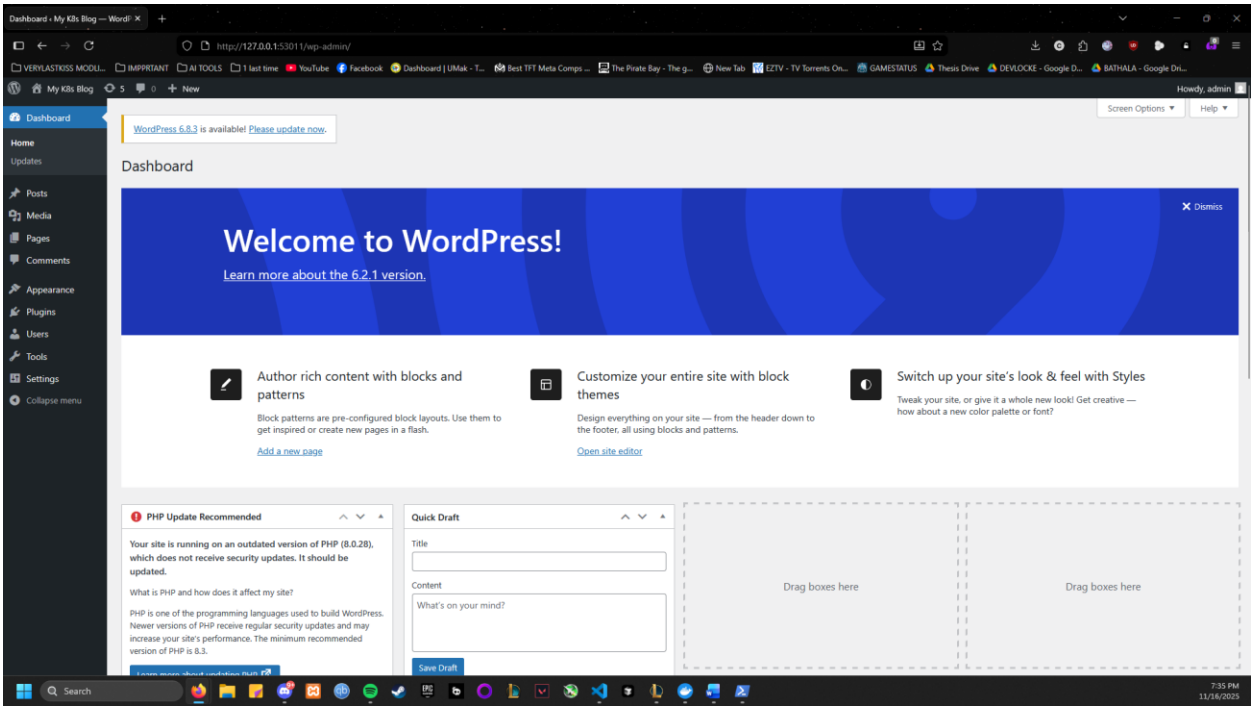
```yaml
    labels:
      app: wordpress
      tier: frontend
  spec:
    containers:
    - image: wordpress:6.2.1-apache
      name: wordpress
      env:
      - name: WORDPRESS_DB_HOST
        value: wordpress-mysql
      - name: WORDPRESS_DB_PASSWORD
        valueFrom:
          secretKeyRef:
            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
```

```
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl apply -f wordpress-deployment.yaml
service/wordpress created
deployment.apps/wordpress created
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes>
```

```
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl get pods
>> kubectl get services
>> kubectl get pvc
NAME                              READY   STATUS              RESTARTS   AGE
shell-demo                        1/1     Running             0          11m
two-containers                    2/2     Running             0          6m29s
wordpress-68859958fd-2pszj        0/1     ContainerCreating   0          10s
wordpress-mysql-5ccb49cfb-vvl9b   1/1     Running             0          88s
NAME              TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes        ClusterIP      10.96.0.1       <none>        443/TCP        16m
wordpress         LoadBalancer   10.96.148.31    <pending>     80:31399/TCP   10s
wordpress-mysql   ClusterIP      None            <none>        3306/TCP       88s
NAME             STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mysql-pv-claim   Bound    pvc-d135ea87-d514-48d3-9788-6bce2fc563ac   20Gi       RWO            standard       <unset>                 3m
wp-pv-claim      Bound    pvc-3fd054dc-7056-49f1-ae32-ca6d40dd89f6   20Gi       RWO            standard       <unset>                 2m34s
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes>
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes>
```

```
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> minikube service wordpress --url
http://127.0.0.1:53011
  ! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```


WordPress › Installation
http://127.0.0.1:53011/wp-admin/install.php?step=1
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
My K8s Blog
Username
admin
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password
Show
Strong
Important: You will need this password to log in. Please store it in a secure location.
Your Email
chummycabus@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.
Install WordPress


Dashboard ‹ My K8s Blog — WordP
http://127.0.0.1:53011/wp-admin/
My K8s Blog
New
Howdy, admin
Screen Options
Help
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
WordPress 6.8.3 is available! Please update now.
Dashboard
Welcome to WordPress!
Learn more about the 6.2.1 version.
Dismiss
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.
Add a new page
Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
Open site editor
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?
Save Draft
Drag boxes here
Drag boxes here
Search
7:35 PM
11/16/2025