

## CABUS, CLEMENT HAROLD MIGUEL

### IV-ACSAD

- Hello Minikube:

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

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

PS C:\WINDOWS\system32> minikube start
* minikube v1.37.0 on Microsoft Windows 11 Pro 10.0.26200.7171 Build 26200.7171
* Automatically selected the docker driver. Other choices: hyperv, ssh
* 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 ...
  > gcr.io/k8s-minikube/kicbase...: 488.52 MiB / 488.52 MiB 100.00% 42.06 M
  > preloaded-images-k8s-v18-v1...: 337.07 MiB / 337.07 MiB 100.00% 14.34 M
* Creating docker container (CPUs=2, Memory=8100MB) ...
! 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: default-storageclass, storage-provisioner

! C:\WINDOWS\system32\kubectl.exe is version 1.28.0, which may have incompatibilities with Kubernetes 1.34.0.
- Want kubectl v1.34.0? Try 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\WINDOWS\system32> kubectl cluster-info
>> kubectl get nodes
Kubernetes control plane is running at https://127.0.0.1:52001
CoreDNS is running at https://127.0.0.1:52001/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
NAME      STATUS    ROLES          AGE     VERSION
minikube  Ready     control-plane  13s    v1.34.0
PS C:\WINDOWS\system32>
```

```

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
  NAME        NAMESPACE   TARGET PORT   URL
  default     default     8080          http://192.168.49.2:31944
* Starting tunnel for service hello-node./r
  NAME        NAMESPACE   TARGET PORT   URL
  default     default     8080          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.

```

**Containers** [Give feedback](#)

Container CPU usage (i) Container memory usage (i) [Show charts](#)

9.64% / 1200% (12 CPUs available) 567.3MB / 15.21GB

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	Actions
<input type="checkbox"/>	minikube	ab59f7ec5ec5	k8s-minikube	51997:22	<a href="#">Show all ports (5)</a>

**Walkthroughs**

<http://127.0.0.1:52116>

NOW: 2025-11-16 11:17:22.819787733 +0000 UTC m=+36.863518540

- Get a Shell to a Running Container:

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER**: A sidebar with a tree view. One item is expanded, labeled "GET A SHELL TO A RUNNING CONTAINER". Inside this folder is a file named "shell-demo.yaml".
- EDITOR**: The main workspace shows the content of "shell-demo.yaml". The YAML code defines a Pod named "shell-demo" with a single container named "shared-data" running an "nginx" image. This container mounts a volume named "shared-data" at the path "/usr/share/nginx/html".
- TERMINAL**: A bottom pane showing a terminal session. The user runs `kubectl apply -f shell-demo.yaml`, creating a pod named "shell-demo". Then, they run `kubectl get pod shell-demo` to check its status, which shows it is "ContainerCreating". Finally, they run `kubectl exec -it shell-demo -- /bin/bash`, which opens a new terminal window showing a root shell prompt on the "shell-demo" pod.

```

apiVersion: v1
kind: Pod
metadata:
  name: shell-demo
spec:
  containers:
    - name: shared-data
      image: nginx
      volumeMounts:
        - name: shared-data
          mountPath: /usr/share/nginx/html
  volumes:
    - name: shared-data
      emptyDir: {}

```

TERMINAL OUTPUT:

- PS D:\chummy\4th year files\Elec 3\Get a Shell to a Running Container> `kubectl apply -f shell-demo.yaml`
- pod/shell-demo created
- PS D:\chummy\4th year files\Elec 3\Get a Shell to a Running Container> `kubectl get pod shell-demo`

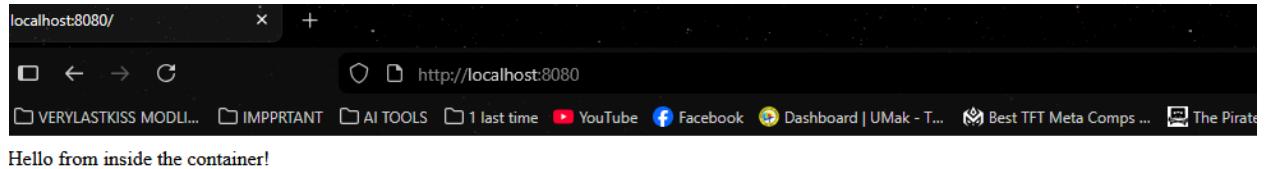
NAME	READY	STATUS	RESTARTS	AGE
shell-demo	0/1	ContainerCreating	0	8s

- PS D:\chummy\4th year files\Elec 3\Get a Shell to a Running Container> `kubectl exec -it shell-demo -- /bin/bash`

```
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

```



- Deploying Wordpress and MySQL with persistent volumes:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl create secret generic mysql-pass --from-literal=password='YourPasswordHere123'
secret/mysql-pass created
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl get secrets
NAME      TYPE    DATA   AGE
mysql-pass  Opaque  1      3s
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes>

```

! mysql-pvc.yaml X

```

! mysql-pvc.yaml
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: mysql-pv-claim
5    labels:
6      app: wordpress
7  spec:
8    accessModes:
9      - ReadWriteOnce
10   resources:
11     requests:
12       storage: 20Gi

```

```

● >> kubectl get pvc
persistentvolumeclaim/mysql-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>          0s
○ PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes>

```

```
! mysql-pvc.yaml X ! wordpress-pvc.yaml X
! wordpress-pvc.yaml
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: wp-pv-claim
5    labels:
6      app: wordpress
7  spec:
8    accessModes:
9      - ReadWriteOnce
10   resources:
11     requests:
12       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  VOLUME ATTRIBUTESCLASS  AGE
mysql-pv-claim  Bound   pvc-d135ea87-d514-48d3-9788-6bce2fc563ac  20Gi      RWO          standard      <unset>           26s
wp-pv-claim    Bound   pvc-3fd05adc-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 X

```
! mysql-deployment.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: wordpress-mysql
5    labels:
6      app: wordpress
7  spec:
8    ports:
9      - port: 3306
10   selector:
11     app: wordpress
12     tier: mysql
13   clusterIP: None
14 ---
15 apiVersion: apps/v1
16 kind: Deployment
17 metadata:
18   name: wordpress-mysql
19   labels:
20     app: wordpress
21 spec:
22   selector:
23     matchLabels:
24       app: wordpress
25       tier: mysql
26   strategy:
27     type: Recreate
28   template:
29     metadata:
30       labels:
31         app: wordpress
32         tier: mysql
33     spec:
34       containers:
35         - image: mysql:8.0
36           name: mysql
37           env:
38             - name: MYSQL_ROOT_PASSWORD
```

```
mysql-pvc.yaml      wordpress-pvc.yaml      mysql-deployment.yaml X
mysql-deployment.yaml
35   - image: mysql:8.0
36     name: mysql
37     env:
38       - name: MYSQL_ROOT_PASSWORD
39         valueFrom:
40           secretKeyRef:
41             name: mysql-pass
42             key: password
43       - name: MYSQL_DATABASE
44         value: wordpress
45       - name: MYSQL_USER
46         value: wordpress
47       - name: MYSQL_PASSWORD
48         valueFrom:
49           secretKeyRef:
50             name: mysql-pass
51             key: password
52     ports:
53       - containerPort: 3306
54         name: mysql
55     volumeMounts:
56       - name: mysql-persistent-storage
57         mountPath: /var/lib/mysql
58     volumes:
59       - name: mysql-persistent-storage
60         persistentVolumeClaim:
61           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-my-sql 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-my-sql-5ccb49cfb-vv19b  0/1     ContainerCreating   0          13s
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl logs deployment/wordpress-my-sql
Error from server (BadRequest): container "mysql" in pod "wordpress-my-sql-5ccb49cfb-vv19b" 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-my-sql-5ccb49cfb-vv19b  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-my-sql-5ccb49cfb-vv19b  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-my-sql-5ccb49cfb-vv19b  1/1     Running   0          29s
PS D:\chummy\4th year files\Elec 3\Deploying Wordpress and MySQL with persistent volumes> kubectl logs deployment/wordpress-my-sql
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-010068] [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.
2025-11-16T11:31:51.51.508725Z 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 X
<a href="#">! mysql-pvc.yaml</a>	<a href="#">! wordpress-pvc.yaml</a>	<a href="#">! mysql-deployment.yaml</a>	<a href="#">! wordpress-deployment.yaml X</a>
<pre> ! wordpress-deployment.yaml 1  apiVersion: v1 2  kind: Service 3  metadata: 4    name: wordpress 5    labels: 6      app: wordpress 7  spec: 8    ports: 9      - port: 80 10   selector: 11     app: wordpress 12     tier: frontend 13   type: LoadBalancer 14 --- 15  apiVersion: apps/v1 16  kind: Deployment 17  metadata: 18    name: wordpress 19    labels: 20      app: wordpress 21  spec: 22    selector: 23      matchLabels: 24        app: wordpress 25        tier: frontend 26    strategy: 27      type: Recreate 28    template: 29      metadata: 30        labels: 31          app: wordpress 32          tier: frontend 33      spec: 34        containers: 35          - image: wordpress:6.2.1-apache 36            name: wordpress 37            env: 38              - name: WORDPRESS_DB_HOST </pre>			

```

! mysql-pvc.yaml X ! wordpress-pvc.yaml ! mysql-deployment.yaml ! wordpress-deployment.yaml X
! wordpress-deployment.yaml
30   labels:
31     app: wordpress
32     tier: frontend
33   spec:
34     containers:
35       - image: wordpress:6.2.1-apache
36         name: wordpress
37         env:
38           - name: WORDPRESS_DB_HOST
39             value: wordpress-mysql
40           - name: WORDPRESS_DB_PASSWORD
41             valueFrom:
42               secretKeyRef:
43                 name: mysql-pass
44                 key: password
45           - name: WORDPRESS_DB_USER
46             value: wordpress
47         ports:
48           - containerPort: 80
49             name: wordpress
50         volumeMounts:
51           - name: wordpress-persistent-storage
52             mountPath: /var/www/html
53         volumes:
54           - name: wordpress-persistent-storage
55             persistentVolumeClaim:
56               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	VOLUME ATTRIBUTES	CLASS	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

VERYLASTKISS MODUL... IMPRTANT AI TOOLS 1 last time YouTube Facebook Dashboard | UMAK - ... Best TFT Meta Comps... The Pirate Bay - The g... New Tab EZTV - TV Torrents On... GAMESTATUS Thesis Drive DEVLOCKE - Go...



## 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	<input type="text" value="My K8s Blog"/>
Username	<input type="text" value="admin"/>
Password	<input type="password" value="*****"/> <input checked="" type="checkbox"/> Show <span style="background-color: #2e3436; color: white; padding: 2px 5px;">Strong</span>
Your Email	<input type="text" value="chummycabus@gmail.com"/>
Search engine visibility	<input type="checkbox"/> Discourage search engines from indexing this site <small>It is up to search engines to honor this request.</small>

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

**Double-check your email address before continuing.**

**Install WordPress.**

Dashboard - My K8s Blog — WordPress

http://127.0.0.1:53011/wp-admin/

VERYLASTKISS MODUL... IMPRTANT AI TOOLS 1 last time YouTube Facebook Dashboard | UMAK - ... Best TFT Meta Comps... The Pirate Bay - The g... New Tab EZTV - TV Torrents On... GAMESTATUS Thesis Drive DEVLOCKE - Google D... BATHALA - Google D... Howdy, admin

Screen Options Help

Dashboard

WordPress 6.8.3 is available! Please update now.

## 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.

[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.

[View more about recommended PHP](#)

**Quick Draft**

Title:

Content:

**Save Draft**

Drag boxes here

Drag boxes here

7:35 PM  
11/16/2025