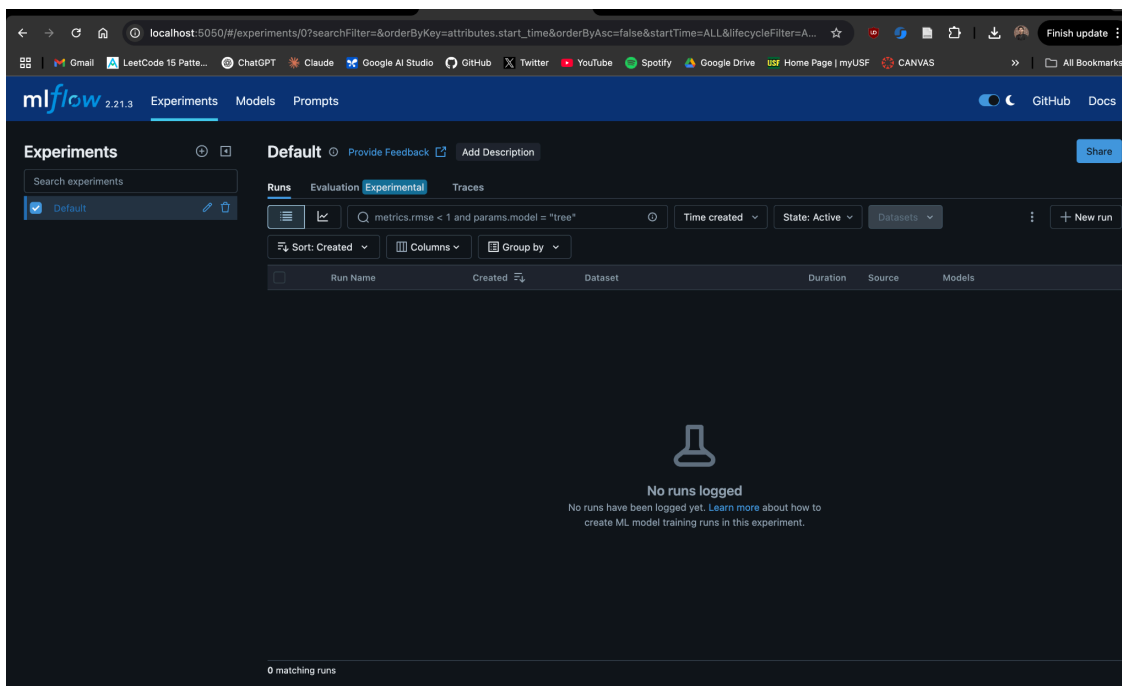


1. Dockerfile Build Output – Terminal showing successful Docker image build for MLflow.

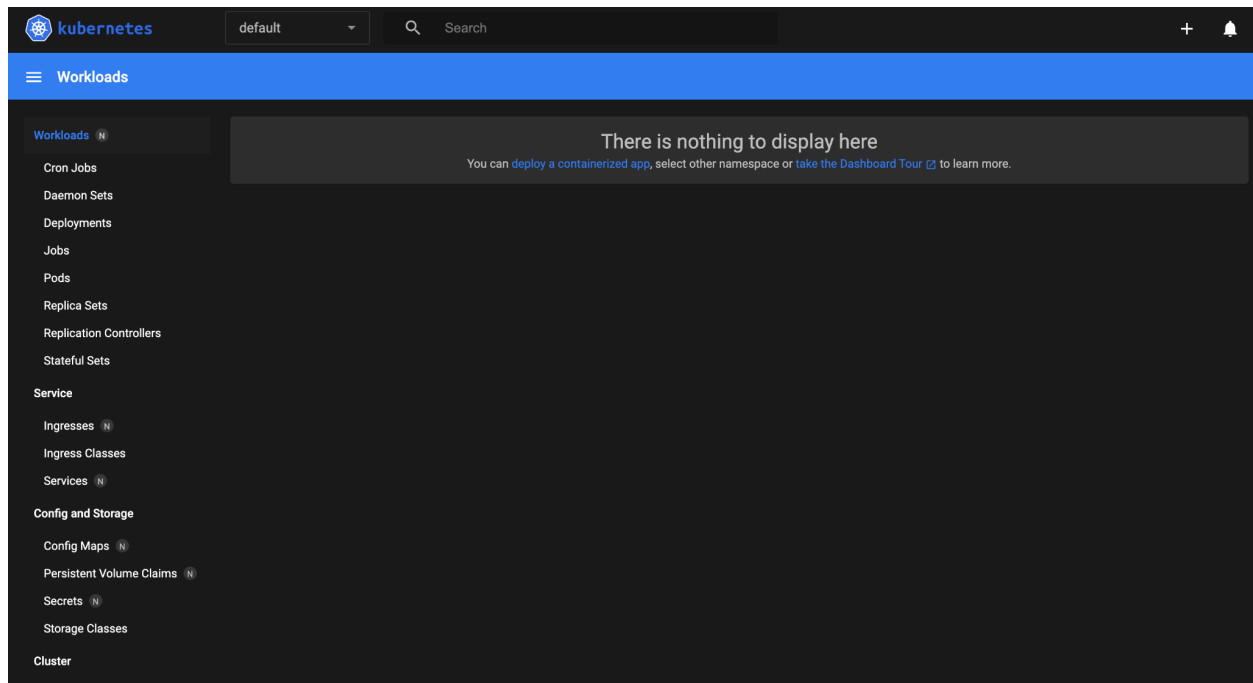
```
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 249B                                              0.0s
[+] Building 54.0s (7/8)                                                         docker:desktop-linux
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 249B                                              0.0s
[+] Building 54.1s (7/8)                                                         docker:desktop-linux
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 249B                                              0.0s
[+] Building 54.3s (7/8)                                                         docker:desktop-linux
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 249B                                              0.0s
[+] Building 54.4s (7/8)                                                         docker:desktop-linux
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 249B                                              0.0s
[+] Building 54.5s (7/8)                                                         docker:desktop-linux
=> [auth] library/python:pull token for registry-1.docker.io                     0.0s
=> [internal] load .dockerignore                                                 0.0s
[+] Building 54.6s (8/8) FINISHED                                               docker:desktop-linux
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 249B                                              0.0s
=> [internal] load metadata for docker.io/library/python:3.10-slim              3.9s
=> [auth] library/python:pull token for registry-1.docker.io                     0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                    0.0s
=> [1/3] FROM docker.io/library/python:3.10-slim@sha256:65c843653048a3ba22c8d5083a022f44aef774974f0f7f70cbf8cee4e931ac96 1.8s
=> => resolve docker.io/library/python:3.10-slim@sha256:65c843653048a3ba22c8d5083a022f44aef774974f0f7f70cbf8cee4e931ac96 0.0s
=> => sha256:38013e0b717a80c4f351a257ea288ea53b24f381309e946ee9f0dfa479ad77c6 250B / 250B 0.1s
=> => sha256:b7f8494172b43bd9ef3a8840decc959ae877d8469d52cfe51e53dbffde08ed9 15.56MB 0.9s
=> => sha256:a7d9a0ac6293889b2e134861072f9099a06d78ca983d7172d7bb8b236008c7c3 3.32MB / 28.07MB 0.6s
=> => sha256:16c9c4a8e9eef856231273efbb42a473740e8d50d74d35e6aedd04ff69fe161f 28.07MB / 28.07MB 1.2s
=> => extracting sha256:16c9c4a8e9eef856231273efbb42a473740e8d50d74d35e6aedd04ff69fe161f 0.4s
=> => extracting sha256:a7d9a0ac6293889b2e134861072f9099a06d78ca983d7172d7bb8b236008c7c3 0.0s
=> => extracting sha256:b7f8494172b43bd9ef3a8840decc959ae877d8469d52cfe51e53dbffde08ed9 0.2s
=> => extracting sha256:38013e0b717a8dc4f351a257ea288ea53b24f381309e946ee9f0dfa479ad77c6 0.0s
=> [2/3] WORKDIR /mlflow                                                         0.1s
=> [3/3] RUN pip install mlflow                                                  24.9s
=> => exporting to image                                                         23.7s
=> => exporting layers                                                         19.8s
=> => exporting manifest sha256:895081f0065fbb1e2563ff41f58a00e4f2595aa049bcb66e8e27dc0e2a0f5c48 0.0s
=> => exporting config sha256:1e64450649cd6094efe20ad2b1a007ca1874688bf785bac677add9d5a6d4fd4fc 0.0s
=> => exporting attestation manifest sha256:62d86d33f0fbf8761e6a68fca8a8d834ff11bea155fe8d4a493eb905d4c5ace1e 0.0s
=> => exporting manifest list sha256:510afe311b7502cb05a4e8ed7b077690474328d1c11e73b0e9dc8ed61b46ce7d 0.0s
=> => naming to docker.io/library/mlflow-server:latest                         0.0s
=> => unpacking to docker.io/library/mlflow-server:latest                       3.9s

What's next:
View a summary of image vulnerabilities and recommendations -> docker scout quickview
```

2. Local MLflow Server Running – Browser view of MLflow UI served from Docker at localhost:5050.



3. Minikube Dashboard (Initial View) – Overview of the Kubernetes cluster via Minikube dashboard.



4. Docker Build in Minikube – Terminal output confirming MLflow image built inside Minikube's Docker.

```
ERROR: failed to solve: failed to read dockerfile: open dockerfile: no such file or directory
21:29:45 nk13:~/gitRepos/mlops-msds603-nk/notebooks/lab4 % cd mlflow_test
21:30:01 nk13:~/gitRepos/mlops-msds603-nk/notebooks/lab4/mlflow_test % docker build -t mlflow-server .
[+] Building 30.4s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 249B
=> [internal] load metadata for docker.io/library/python:3.10-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/python:3.10-slim@sha256:65c843653048a3ba22c8d5083a022f44aef774974f0f7f70cbf8cee4e931ac96
=> => resolve docker.io/library/python:3.10-slim@sha256:65c843653048a3ba22c8d5083a022f44aef774974f0f7f70cbf8cee4e931ac96
=> => sha256:a9bc37fa702940246619c7b6f6309c8c73396a236f83d1a59fca6fe9b4f32eaa 1.75kB / 1.75kB
=> => sha256:d17a2e30e3177adc53097a3ff245f44642263157b1606e56103bfa63d162985f 5.32kB / 5.32kB
=> => sha256:16c9c4a8e9eef856231273efbb42a473740e8d50d74d35e6aedd04ff69fe161f 28.07MB / 28.07MB
=> => sha256:a7d9a0ac6293889b2e134861072f9099a06d78ca983d7172d7bb8b236008c7c3 3.33MB / 3.33MB
=> => sha256:bf8494172b43bdd9ef3a8840deecc959ae877d8469d52cfe51e53dbffde08ed9 15.56MB / 15.56MB
=> => sha256:65c843653048a3ba22c8d5083a022f44aef774974f0f7f70cbf8cee4e931ac96 9.13kB / 9.13kB
=> => sha256:38013e09717a8dc4f351a257ea288ea53b24f381309e946ee9f0dfa479a477c6 250B / 250B
=> => extracting sha256:16c9c4a8e9eef856231273efbb42a473740e8d50d74d35e6aedd04ff69fe161f
=> => extracting sha256:a7d9a0ac6293889b2e134861072f9099a06d78ca983d7172d7bb8b236008c7c3
=> => extracting sha256:bf8494172b43bdd9ef3a8840deecc959ae877d8469d52cfe51e53dbffde08ed9
=> => extracting sha256:38013e09717a8dc4f351a257ea288ea53b24f381309e946ee9f0dfa479a477c6
=> [2/3] WORKDIR /mlflow
=> [3/3] RUN pip install mlflow
=> exporting to image
=> exporting layers
=> writing image sha256:2e4012d342c735080d3d90aa7ae62f0f42aa6e8dae06af5c971687a4ba6835ac
=> naming to docker.io/library/mlflow-server
What's next:
View a summary of image vulnerabilities and recommendations -> docker scout quickview
21:31:35 nk13:~/gitRepos/mlops-msds603-nk/notebooks/lab4/mlflow_test %
```

5. Minikube Dashboard (Full View) – Dashboard showing running Kubernetes components.

The screenshot displays the Minikube Dashboard in its full view. The interface features a dark theme with a blue header bar. On the left, a sidebar lists various Kubernetes components under categories like Workloads, Service, Config and Storage, and Cluster. The main content area is titled 'Workload Status' and shows three large green circles representing the status of Deployments, Pods, and Replica Sets, each with a 'Running: 1' label. Below this, there are two tables: 'Deployments' and 'Pods'. The 'Deployments' table shows a single deployment named 'mlflow-deployment' using the 'mlflow-server' image, with 1 pod running, created 'a minute ago'. The 'Pods' table shows a single pod named 'app: mlflow'.

Name	Images	Labels	Pods	Created ↑
mlflow-deployment	mlflow-server	-	1 / 1	a minute ago

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
app: mlflow								

6. Kubernetes Deployments Tab – MLflow deployment successfully created and running.

This screenshot shows the 'Deployments' tab within the Kubernetes Dashboard. The browser's address bar indicates the URL: `127.0.0.1:50804/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/deployment?namespace=de...`. The sidebar on the left is expanded to show the 'Deployments' tab. The main content area displays a table with the details of the 'mlflow-deployment'. The table shows that the deployment is using the 'mlflow-server' image, has 1 pod running, and was created '2 minutes ago'.

Name	Images	Labels	Pods	Created ↑
mlflow-deployment	mlflow-server	-	1 / 1	2 minutes ago

7. Kubernetes Pods Tab – View of the MLflow pod running in the Minikube cluster.

The screenshot shows the Kubernetes dashboard interface. The top navigation bar includes the Kubernetes logo, a dropdown menu set to 'default', a search bar, and a 'Finish update' button. The left sidebar contains a list of workload types: Cron Jobs, Daemon Sets, Deployments, Jobs, Pods (selected), Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, Ingress Classes, Services, Config Maps, Persistent Volume Claims, Secrets, Storage Classes, and Cluster. The main content area is titled 'Pods' and contains a table with the following columns: Name, Images, Labels, Node, Status, Restarts, CPU Usage (cores), Memory Usage (bytes), and Created. A single pod is listed: 'mlflow-deployment-6854dc8c4b-mjhkw' with image 'mlflow-server', labels 'app: mlflow' and 'pod-template-hash: 6854dc8c4b', running on node 'minikube'. The status is 'Running' with 0 restarts. CPU and memory usage are indicated by dashes. The pod was created 3 minutes ago.

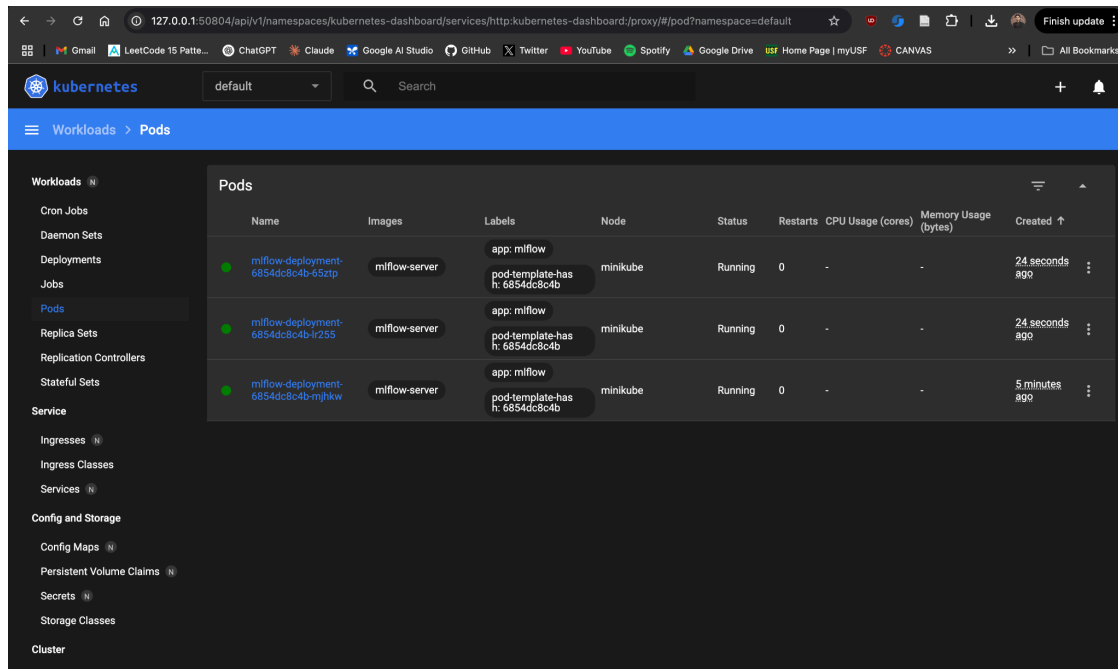
Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
mlflow-deployment-6854dc8c4b-mjhkw	mlflow-server	app: mlflow pod-template-hash: 6854dc8c4b	minikube	Running	0	-	-	3 minutes ago

8. Kubernetes Services Tab – MLflow service created with NodePort configuration.

The screenshot shows the Kubernetes dashboard interface. The top navigation bar includes the Kubernetes logo, a dropdown menu set to 'default', a search bar, and a 'Finish update' button. The left sidebar contains a list of workload types: Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service (selected), Ingresses, Ingress Classes, Services, Config Maps, Persistent Volume Claims, Secrets, Storage Classes, and Cluster. The main content area is titled 'Services' and contains a table with the following columns: Name, Labels, Type, Cluster IP, Internal Endpoints, External Endpoints, and Created. Two services are listed: 'mlflow-service' with type 'NodePort', cluster IP '10.111.204.239', and internal endpoints 'mlflow-service:5050 TCP' and 'mlflow-service:32495 TCP'; and 'kubernetes' with type 'ClusterIP', cluster IP '10.96.0.1', and internal endpoints 'kubernetes:443 TCP' and 'kubernetes:0 TCP'. Both were created 3 and 15 minutes ago respectively.

Name	Labels	Type	Cluster IP	Internal Endpoints	External Endpoints	Created
mlflow-service	-	NodePort	10.111.204.239	mlflow-service:5050 TCP mlflow-service:32495 TCP	-	3 minutes ago
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	15 minutes ago

9. Scaled MLflow Pods (Replicas) – Dashboard showing 3 MLflow pod replicas running after scaling.



10. MLflow Port Forwarded & Accessed – MLflow UI accessed via port-forwarded service at localhost:5050.

