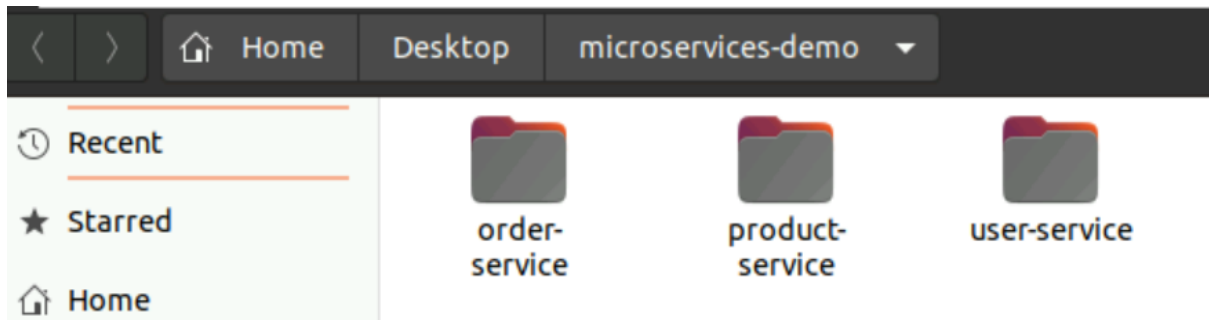
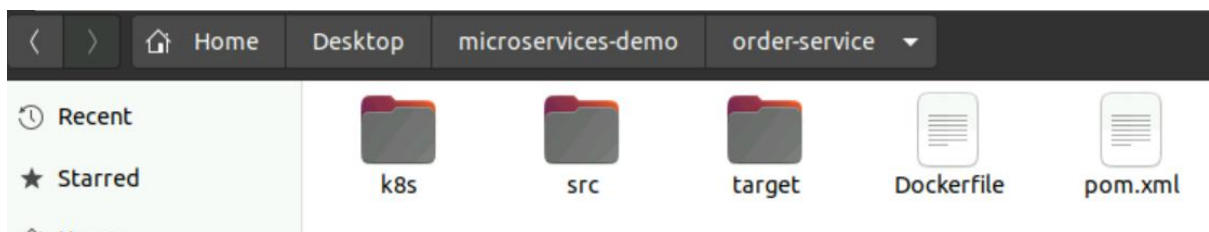


ABDUL SAMIM MONDAL Q1 SOLUTION

CREATE A JAVA MICROSERVICES



These files should be there inside each service



```
master@master-vm:~/Desktop/microservices-demo$ tree
.
├── Jenkinsfile
├── Jenkinsfile.save
├── order-service
│   ├── Dockerfile
│   ├── k8s
│   │   ├── deployment.yaml
│   │   └── service.yaml
│   ├── pom.xml
│   └── src
│       ├── main
│       │   ├── java
│       │   │   └── com
│       │   │       └── example
│       │   │           └── demo
│       │   │               ├── DemoApplication.java
│       │   │               └── HelloController.java
│       │   └── resources
│       │       └── application.properties
│       └── target
│           ├── classes
│           │   ├── application.properties
│           │   └── com
│           │       └── example
│           │           └── demo
│           │               ├── DemoApplication.class
│           │               └── HelloController.class
│           ├── demo-0.0.1-SNAPSHOT.jar
│           ├── demo-0.0.1-SNAPSHOT.jar.original
│           ├── generated-sources
│           │   └── annotations
│           ├── maven-archiver
│           │   └── pom.properties
│           ├── maven-status
│           └── maven-compiler-plugin
└──
```

ABDUL SAMIM MONDAL Q1 SOLUTION



Build the services

```
[INFO] -----
master@master-vm:~/Desktop/microservices-demo/product-service$ cd ..
master@master-vm:~/Desktop/microservices-demo$ ls
order-service  product-service  user-service
master@master-vm:~/Desktop/microservices-demo$ cd order-service/
master@master-vm:~/Desktop/microservices-demo/order-service$ mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:demo >-----
[INFO] Building demo 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ demo ---
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:resources (default-resources) @ demo ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ demo ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 2 source files to /home/master/Desktop/microservices-demo/order-service/target/classes
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @ demo ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] skip non existing resourceDirectory /home/master/Desktop/microservices-demo/order-service/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:testCompile (default-testCompile) @ demo ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ demo ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ demo ---
```

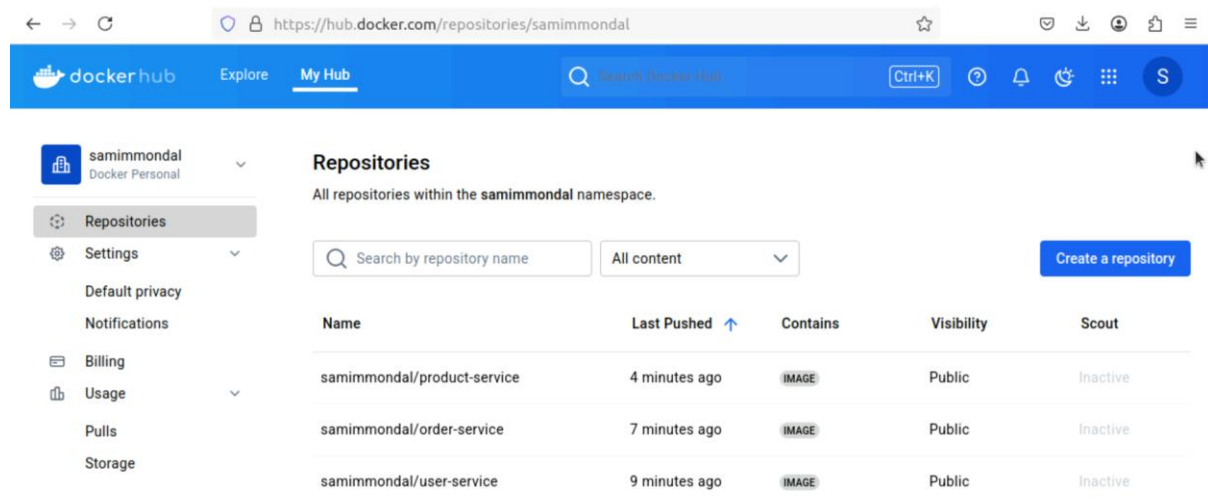
ABDUL SAMIM MONDAL Q1 SOLUTION

Build and push the images onto dockerhub

```
master@master-vm:~/Desktop/microservices-demo/order-service$ docker build -t samimmondal/order-service:1.0 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 17.72MB
Step 1/4 : FROM openjdk:11-jre-slim
--> 764a04af3eff
Step 2/4 : COPY target/demo-0.0.1-SNAPSHOT.jar /app.jar
--> b7c24a9e77af
Step 3/4 : EXPOSE 8080
--> Running in 2c3bc9c02b1f
--> Removed intermediate container 2c3bc9c02b1f
--> d3d01715ec95
Step 4/4 : ENTRYPOINT ["java", "-jar", "/app.jar"]
--> Running in d90224cacfa2
--> Removed intermediate container d90224cacfa2
--> 731da3dd7cf0
Successfully built 731da3dd7cf0
Successfully tagged samimmondal/order-service:1.0
master@master-vm:~/Desktop/microservices-demo/order-service$ docker push samimmondal/ordre-service:1.0
The push refers to repository [docker.io/samimmondal/ordre-service]
An image does not exist locally with the tag: samimmondal/ordre-service
master@master-vm:~/Desktop/microservices-demo/order-service$ docker push samimmondal/order-service:1.0
The push refers to repository [docker.io/samimmondal/order-service]
ae384afa8578: Pushing [=====] 9.241MB/17.69MB
d7802b8508af: Mounted from samimmondal/user-service
e3abdc2e9252: Mounted from samimmondal/user-service
fa8578: Pushed
eafe6e032dbd: Mounted from samimmondal/user-service
92a4e8a3140f: Mounted from samimmondal/user-service
1.0: digest: sha256:40ed0a364a99358ede319a25502b46024893b7a76b22979d8327137d3a7158c8 size: 1371
```

Dockerhub



The screenshot shows the Docker Hub profile for 'samimmondal'. The left sidebar contains navigation links: Repositories, Settings, Default privacy, Notifications, Billing, Usage, Pulls, and Storage. The main area is titled 'Repositories' and shows a list of repositories within the 'samimmondal' namespace. The list includes 'product-service', 'order-service', and 'user-service', all with 'Public' visibility and 'Inactive' status. A 'Create a repository' button is visible in the top right of the repository list.

Name	Last Pushed	Contains	Visibility	Scout
samimmondal/product-service	4 minutes ago	IMAGE	Public	Inactive
samimmondal/order-service	7 minutes ago	IMAGE	Public	Inactive
samimmondal/user-service	9 minutes ago	IMAGE	Public	Inactive

Deploy each service to Kubernetes

```
master@master-vm:~/Desktop/microservices-demo$ cd user-service/k8s
master@master-vm:~/Desktop/microservices-demo/user-service/k8s$ sed -i 's|IMAGE_TAG|1.0|g' deployment.yaml
master@master-vm:~/Desktop/microservices-demo/user-service/k8s$ kubectl apply -f deployment.yaml
deployment.apps/user-service created
master@master-vm:~/Desktop/microservices-demo/user-service/k8s$ kubectl apply -f service.yaml
service/user-service created
master@master-vm:~/Desktop/microservices-demo/user-service/k8s$ cd ../../
```

ABDUL SAMIM MONDAL Q1 SOLUTION

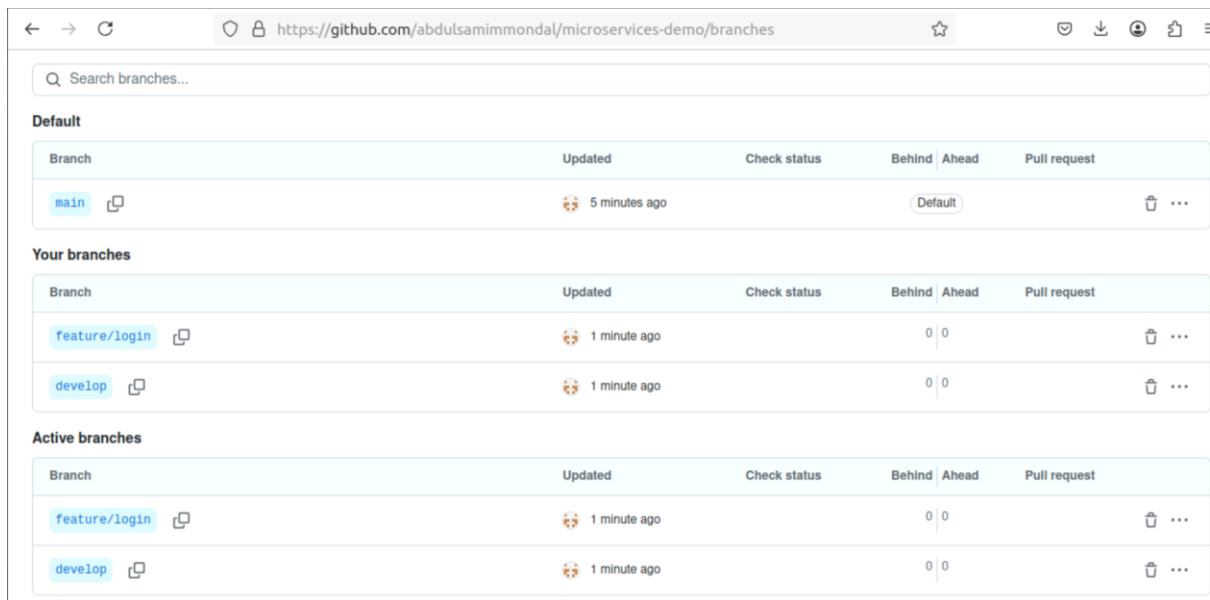
Check if pods are running

```
master@master-vm:~/Desktop/microservices-demo$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
order-service-cfbf54ccb-rxg7k      1/1     Running   0           70s
product-service-c96b5f5c7-lz6lk    1/1     Running   0           70s
user-service-6b6d7f6567-4v2v8      1/1     Running   0           69s
```

Check services are exposing ports correctly

```
master@master-vm:~/Desktop/microservices-demo$ kubectl get svc
NAME            TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)          AGE
kubernetes      ClusterIP   10.96.0.1     <none>       443/TCP          47h
nginx-service   NodePort    10.105.157.108 <none>       80:32105/TCP     39h
order-service   ClusterIP   10.107.11.171 <none>       80/TCP           16m
product-service ClusterIP   10.98.191.7    <none>       80/TCP           17m
user-service    ClusterIP   10.111.231.205 <none>       80/TCP           19m
```

Github repo



The screenshot shows the GitHub repository page for 'microservices-demo' by 'abdulsamimmondal'. The URL is 'https://github.com/abdulsamimmondal/microservices-demo/branches'. The page displays a list of branches under three sections: 'Default', 'Your branches', and 'Active branches'. Each section contains a table with columns for Branch, Updated, Check status, Behind, Ahead, and Pull request.

Branch	Updated	Check status	Behind	Ahead	Pull request
main	5 minutes ago				Default

Branch	Updated	Check status	Behind	Ahead	Pull request
feature/login	1 minute ago		0	0	
develop	1 minute ago		0	0	

Branch	Updated	Check status	Behind	Ahead	Pull request
feature/login	1 minute ago		0	0	
develop	1 minute ago		0	0	

ABDUL SAMIM MONDAL Q1 SOLUTION

The screenshot shows the GitHub repository page for 'microservices-demo' by user 'abdulsamimmondal'. The repository is public and has 3 branches and 0 tags. The main branch is selected. The commit history shows an initial commit of 'microservices demo' by 'abdulsamimmondal' 11 minutes ago. The commit message is 'Initial commit of microservices demo'. The commit hash is 'c176e13'. The commit is linked to 1 commit. Below the commit history, there is a table of files:

File	Commit	Time
order-service	Initial commit of microservices demo	11 minutes ago
product-service	Initial commit of microservices demo	11 minutes ago
user-service	Initial commit of microservices demo	11 minutes ago

Webhook is up and running successfully

The screenshot shows the GitHub repository settings page for 'microservices-demo' by user 'abdulsamimmondal'. The 'Settings' tab is selected. The 'Webhooks' section is active, showing a list of webhooks. A single webhook is listed with the URL 'https://d1ca-103-182-177-42.ngrok...' and the event type '(push)'. The status is 'Last delivery was successful'. The 'Add webhook' button is visible. The 'Webhooks' section is titled 'Webhooks' and includes a description: 'Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).' The 'Add webhook' button is located at the top right of the section. The 'Edit' and 'Delete' buttons are located at the bottom right of the webhook entry.

ABDUL SAMIM MONDAL Q1 SOLUTION

Using ngrok so github can reach local Jenkins

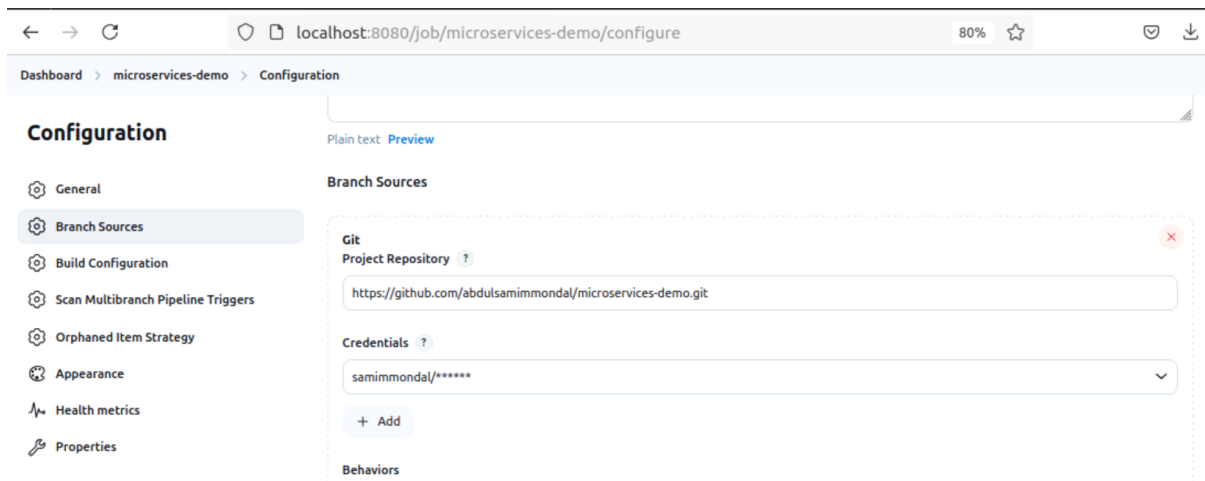
```
ngrok
♥ ngrok? We're hiring https://ngrok.com/careers

Session Status      online
Account             ABDUL SAMIM MONDAL (Plan: Free)
Version             3.22.0
Region              India (in)
Latency              96ms
Web Interface        http://127.0.0.1:4040
Forwarding            https://d1ca-103-182-177-42.ngrok-free.app -> http://localhost:8080

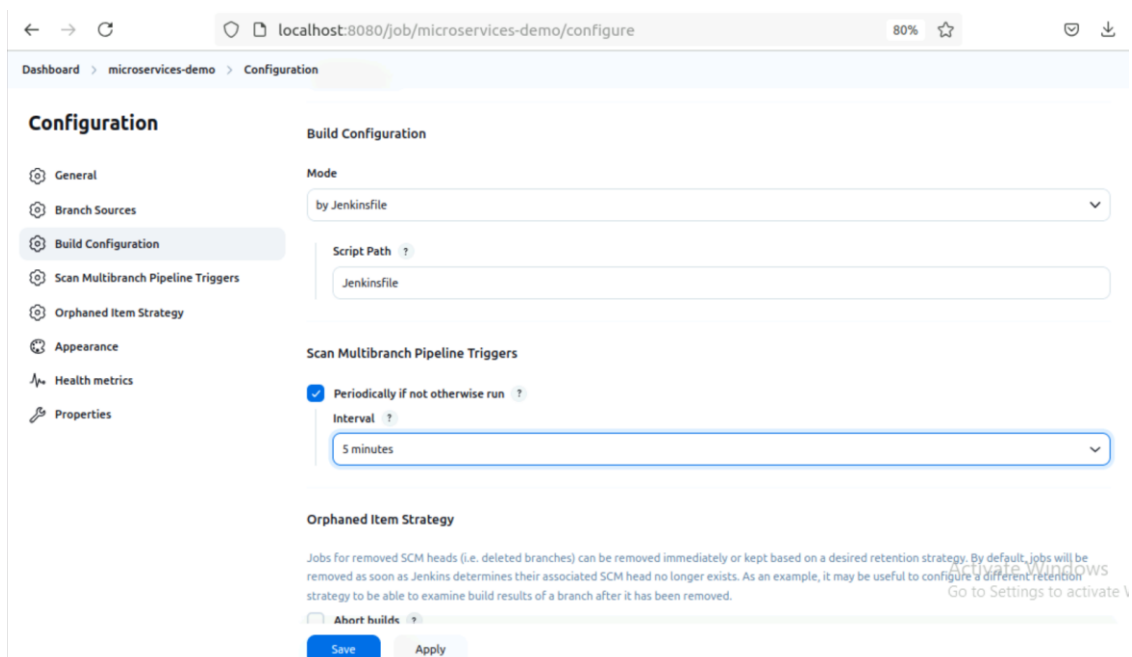
Connections
  ttl    opn    rt1    rt5    p50    p90
   1      0     0.00   0.00   31.27  31.27

HTTP Requests
-----
15:41:54.614 IST POST /github-webhook/      200 OK
```

Jenkins pipeline configuration



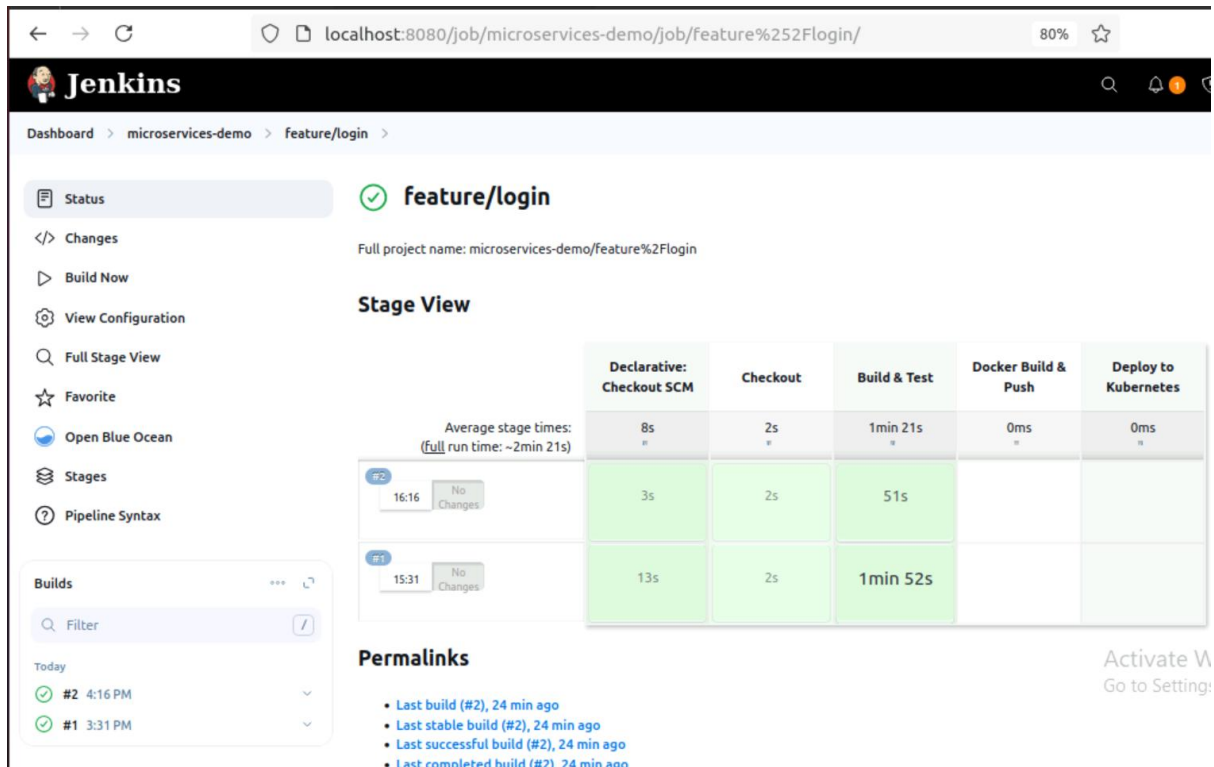
In case webhooks does not work choose this option-



ABDUL SAMIM MONDAL Q1 SOLUTION

Jenkins output

Feature branch



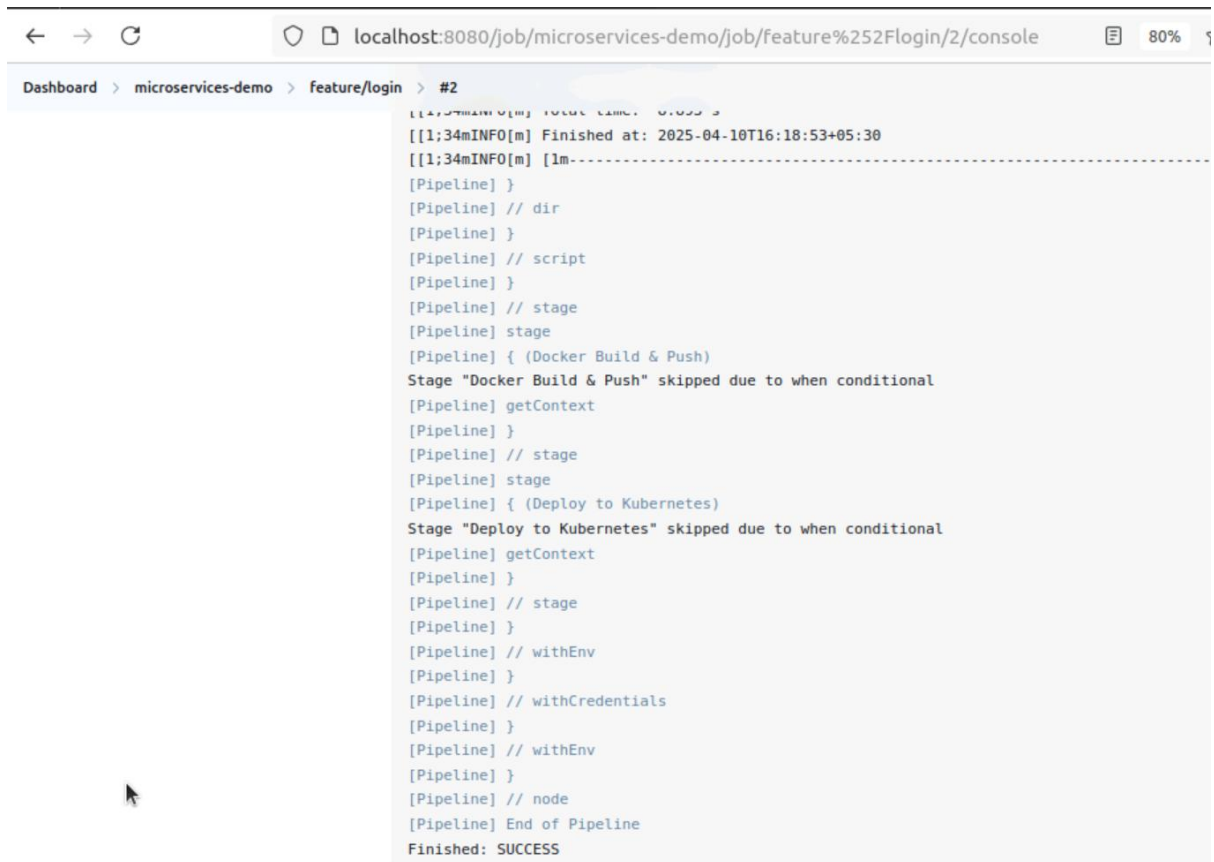
The screenshot shows the Jenkins web interface for a job named 'feature/login'. The left sidebar contains navigation links: Status, Changes, Build Now, View Configuration, Full Stage View, Favorite, Open Blue Ocean, Stages, and Pipeline Syntax. The main area displays the 'feature/login' job status as successful. Below this, the 'Stage View' table shows the following data:

	Declarative: Checkout SCM	Checkout	Build & Test	Docker Build & Push	Deploy to Kubernetes
Average stage times: (full run time: ~2min 21s)	8s	2s	1min 21s	0ms	0ms
#2 (16:16, No Changes)	3s	2s	51s		
#1 (15:31, No Changes)	13s	2s	1min 52s		

Below the table, the 'Permalinks' section lists:

- Last build (#2), 24 min ago
- Last stable build (#2), 24 min ago
- Last successful build (#2), 24 min ago
- Last completed build (#2), 24 min ago

The 'Builds' section on the left shows two builds: #2 at 4:16 PM and #1 at 3:31 PM, both with green status icons.

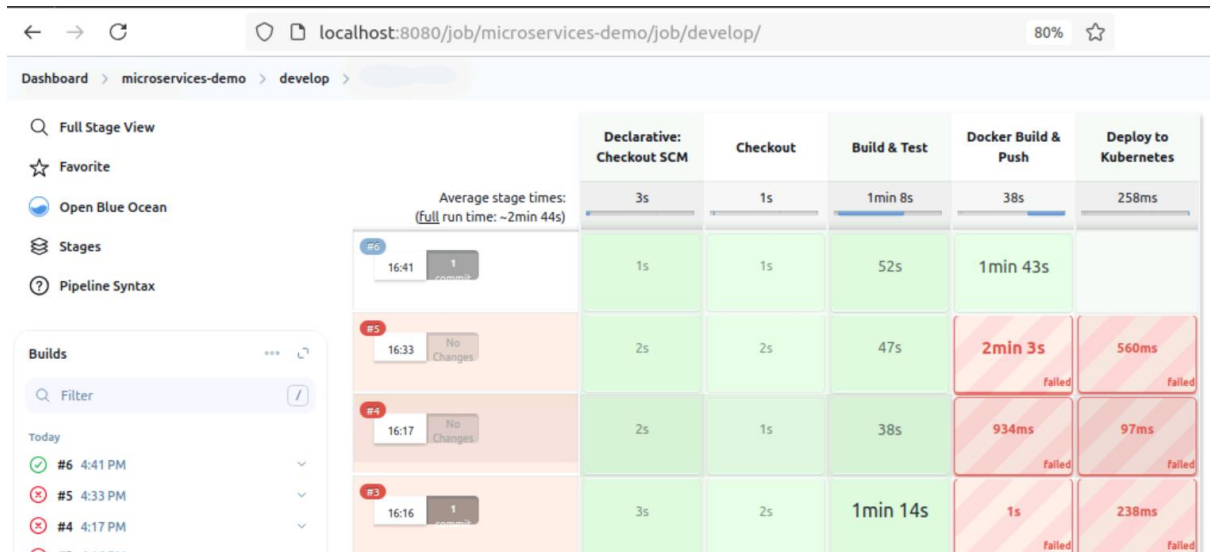


The screenshot shows the Jenkins console output for build #2 of the 'feature/login' job. The output is as follows:

```
[[1;34mINFO[m] Finished at: 2025-04-10T16:18:53+05:30
[[1;34mINFO[m] [1m-----
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Docker Build & Push)
Stage "Docker Build & Push" skipped due to when conditional
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy to Kubernetes)
Stage "Deploy to Kubernetes" skipped due to when conditional
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

ABDUL SAMIM MONDAL Q1 SOLUTION

Develop branch

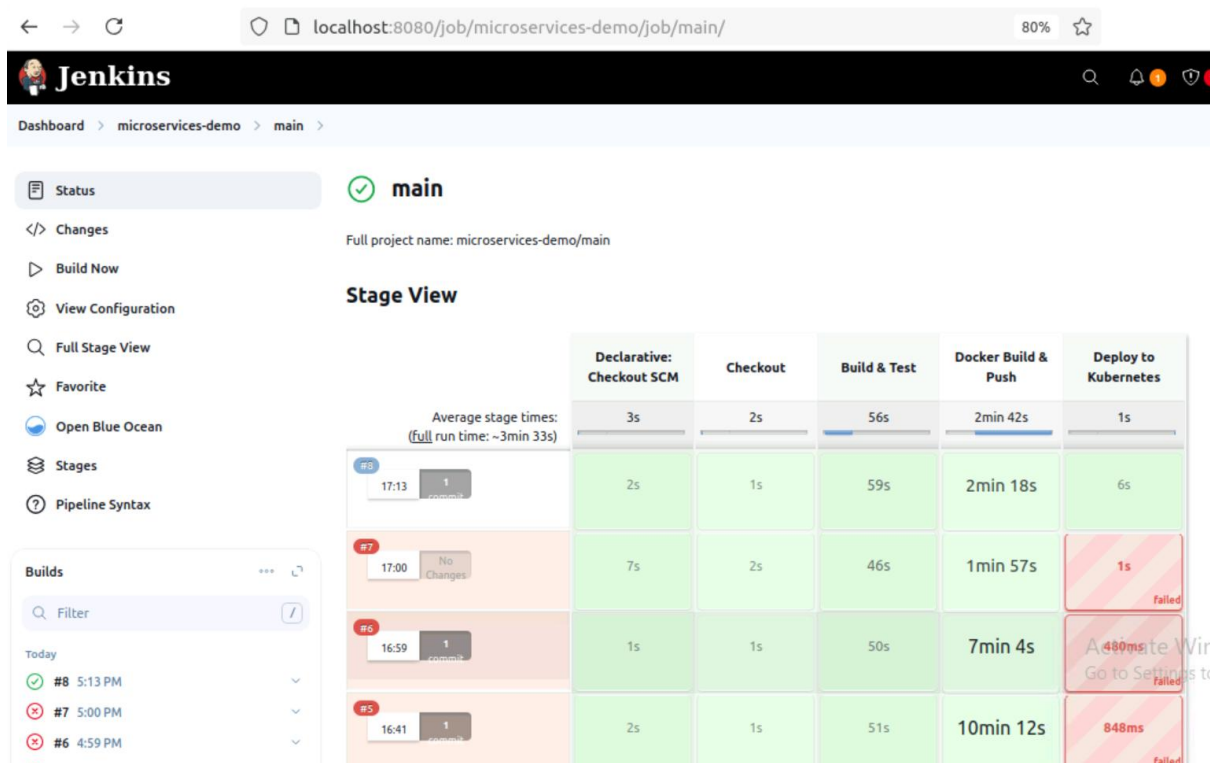


```
Dashboard > microservices-demo > develop > #6
92a4e8a3140f: Preparing
92a4e8a3140f: Waiting
eafe6e032dbd: Layer already exists
d7802b8508af: Layer already exists
e3abdc2e9252: Layer already exists
92a4e8a3140f: Layer already exists
38edad74299d: Pushed
1.0: digest: sha256:758b9159595ab3c2144ccc4f47e6c298143902b0ca44bbf686ecbab8d5f8e77d size: 1371
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy to Kubernetes)
Stage "Deploy to Kubernetes" skipped due to when conditional
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

AS SOON AS COMMITS ARE MADE THE BUILD IN JENKINS STARTS AUTOMATICALLY!

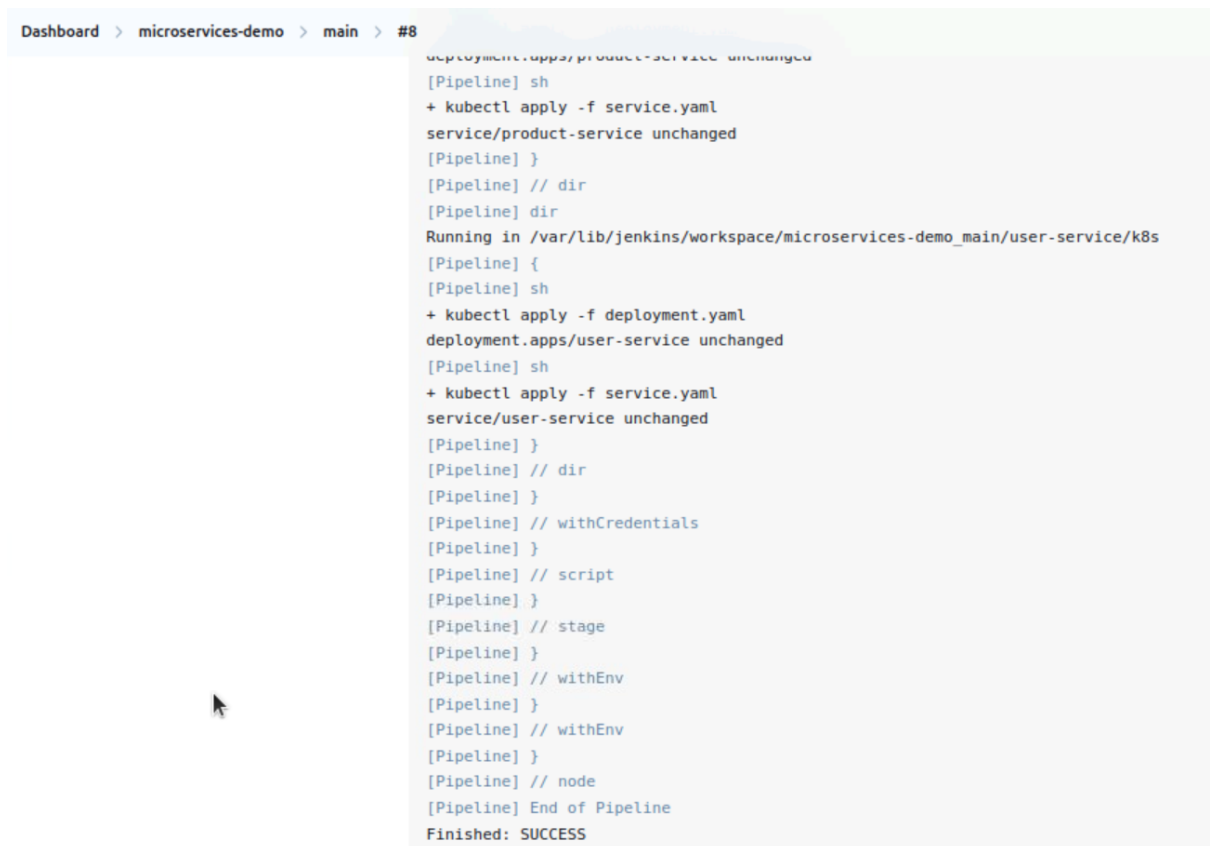
ABDUL SAMIM MONDAL Q1 SOLUTION

Main branch



The screenshot shows the Jenkins web interface for a project named 'microservices-demo' on the 'main' branch. The left sidebar contains navigation links: Status, Changes, Build Now, View Configuration, Full Stage View, Favorite, Open Blue Ocean, Stages, and Pipeline Syntax. The main area displays the 'main' branch status with a green checkmark and a full project name. Below this is the 'Stage View' table, which shows the duration of each stage for the last four builds. The stages are: Declarative: Checkout SCM, Checkout, Build & Test, Docker Build & Push, and Deploy to Kubernetes. The table indicates that the last two builds (#7 and #8) failed in the 'Deploy to Kubernetes' stage.

	Declarative: Checkout SCM	Checkout	Build & Test	Docker Build & Push	Deploy to Kubernetes
Average stage times: (full run time: ~3min 33s)	3s	2s	56s	2min 42s	1s
#8 17:13 1 commit	2s	1s	59s	2min 18s	6s
#7 17:00 No Changes	7s	2s	46s	1min 57s	1s failed
#6 16:59 1 commit	1s	1s	50s	7min 4s	480ms failed
#5 16:41 1 commit	2s	1s	51s	10min 12s	848ms failed



The screenshot shows the Jenkins pipeline log for build #8. The log displays the execution of a pipeline with stages for 'Checkout SCM', 'Checkout', 'Build & Test', and 'Deploy to Kubernetes'. The pipeline is successful, as indicated by the 'Finished: SUCCESS' message at the bottom.

```
deployment.apps/product-service unchanged
[Pipeline] sh
+ kubectl apply -f service.yaml
service/product-service unchanged
[Pipeline] }
[Pipeline] // dir
[Pipeline] dir
Running in /var/lib/jenkins/workspace/microservices-demo_main/user-service/k8s
[Pipeline] {
[Pipeline] sh
+ kubectl apply -f deployment.yaml
deployment.apps/user-service unchanged
[Pipeline] sh
+ kubectl apply -f service.yaml
service/user-service unchanged
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```