

Cloud Computing: Project 6: Deliverables

Building an E-commerce Microservices Application on Cloud using Docker,
Kubernetes, Jenkins and Git

PES2UG21CS461	SAKSHAM ALOK
PES2UG21CS463	SAMANTHA SUSAN JOHN
PES2UG21CS915	HIMANSHU NANDA
PES2UG21CS917	KUSHAAGRA SHRIVASTAVA

Table of Contents

WEEK 1	1
SCREENSHOTS	1
GITHUB REPOSITORY LINK	2
WEEK 2	2
DOCKER	3
CONTAINERS	3
AUTH-LOGIN	4
PRODUCT-ID	5
PRODUCT-USER	6
USER	7
KUBERNETES	8
ORCHESTRATION	8
CODE	9
WEEK 3	11
OVERVIEW	11
DASHBOARD	11
BUILD HISTORY	11
FILES	12
AUTH	12
PRODUCTS-ID	13
PRODUCTS-USER	14
USER	15

Week 1

Microservices Architecture Design and Development

Screenshots

Login

Enter your information to login to your account

Email

Password

Please fill in this field.

Login

Do not have an account? [Sign up](#)

Sign Up

Enter your information to create an account

Name

Email


Password

Phone Number

Sign Up

Already have an account? [Login](#)

Used Electronics

 Saksham Alok [Upload](#) [Logout](#)

Filters

Categories

SoCs

Processors

Sensors

Displays

Memory

Connectivity

Other


Prices

Under Rs.500


Rs.500 - Rs.1000

Rs.1000 - Rs.5000


Clear



Arduino Uno
6 months old
₹ 400



Sensor Kit
7 months old
₹ 700



Grove - 4-Digit Display
11 months old
₹ 550



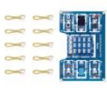

Cloud Computing Project 6 Deliverables

461-463-915-917

Used Electronics

Saksham Alok [Upload](#) [Logout](#)

Listed Products:

 <p>Raspberry Pi 4 1 year old ₹ 900 Sold</p>	 <p>Arduino Uno 6 months old ₹ 400</p>	 <p>Sensor Kit 7 months old ₹ 700</p>	 <p>Grove - 4-Digit Display 11 months old ₹ 550</p>
---	---	--	--

Bought Products:



Used Electronics

Saksham Alok [Upload](#) [Logout](#)

Upload Product

Product Name

Product Description

Age of the Product

Image Link

Category

SoCs

Price

[Upload Product](#)

Used Electronics

Saksham Alok [Upload](#) [Logout](#)

Search Results for: Grove

 <p>Grove - 4-Digit Display 11 months old ₹ 550</p>
--

GitHub Repository Link

<https://github.com/KushShriv/Monolithic-to-Microservices-CC.git>

Week 2
Containerization and Orchestration

Docker
Containers

TAG

user

Last pushed 10 minutes ago by [himanshunanda22](#)

docker pull himanshunanda22/project-6:user

Copy

Digest	OS/ARCH	Last pull	Compressed Size ⓘ
0d9575390b9f	linux/amd64	---	675.47 MB

TAG

products-user

Last pushed 19 minutes ago by [himanshunanda22](#)

docker pull himanshunanda22/project-6:products-user

Copy

Digest	OS/ARCH	Last pull	Compressed Size ⓘ
5c086b978599	linux/amd64	---	675.39 MB

TAG

auth-login

Last pushed 44 minutes ago by [himanshunanda22](#)

docker pull himanshunanda22/project-6:auth-login

Copy

Digest	OS/ARCH	Last pull	Compressed Size ⓘ
f422a08eab1c	linux/amd64	---	674.81 MB

TAG

products-id

Last pushed an hour ago by [himanshunanda22](#)

docker pull himanshunanda22/project-6:products-id

Copy

Digest	OS/ARCH	Last pull	Compressed Size ⓘ
38dff88e29a9	linux/amd64	---	674.87 MB

Cloud Computing Project 6 Deliverables 461-463-915-917

Auth-login

```
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker build -t auth-login:latest -f app/api/auth/login/Dockerfile .
[+] Building 0.0s (0/0) docker:default
[+] Building 28.6s (12/12) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 624B
=> [internal] load metadata for docker.io/library/node:20
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 5.27kB
=> [1/7] FROM docker.io/library/node:20@sha256:844b41cf784f66d7920fd673f7af54ca7b81e289985edc6cd864e7d05e0d133c
=> CACHED [2/7] WORKDIR /app
=> CACHED [3/7] COPY package*.json ./
=> CACHED [4/7] RUN npm install
=> [5/7] COPY . .
=> [6/7] RUN npx prisma generate
=> [7/7] RUN npm run build
=> exporting to image
=> => exporting layers
=> => writing image sha256:7ef3cad56257d67031ef4e6e3e963401b9c102357b749a897f40513a9f3a7dfd
=> => naming to docker.io/library/auth-login:latest

View build details: docker-desktop://dashboard/build/default/default/6tyrnjgdpfpmcxa5yho2u259

• What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
○ PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> 
```

```
• PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker push himanshunanda22/project-6:auth-login
The push refers to repository [docker.io/himanshunanda22/project-6]
9223bcf8d8b4: Layer already exists
40afb7066e00: Layer already exists
b1f8280ea5b9: Layer already exists
44fa5c5c0bba: Layer already exists
14c2694f88e3: Layer already exists
72702b05020d: Layer already exists
123194e8ca81: Layer already exists
a28e61f75fa0: Layer already exists
41c2d1f0a1d3: Layer already exists
3e81cc85b636: Layer already exists
893507f6057f: Layer already exists
2353f7120e0e: Layer already exists
51a9318e6edf: Layer already exists
c5bb35826823: Layer already exists
auth-login: digest: sha256:f422a08eab1c84b6899d5bf7bf6aea2d8a23ad0912c1edach0e3e2da3515e43e size: 3265
○ PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> 
```

```
# Use an appropriate base image for Node.js applications
FROM node:20

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application code to the working directory
COPY . .

# Run prisma generate
RUN npx prisma generate

# Build the Next.js application
RUN npm run build

# Expose the port the app runs on (assuming it's 3000)
EXPOSE 3000

# Command to run the application
CMD ["npm", "start"]
```

Cloud Computing Project 6 Deliverables

461-463-915-917

Product-id

```
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker build -t products-id:latest -f app/api/products/[i
d]\Dockerfile .
[+] Building 0.0s (0/0) docker:default
[+] Building 31.5s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 800B
=> [internal] load metadata for docker.io/library/node:20
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/7] FROM docker.io/library/node:20@sha256:844b41cf784f66d7920fd673f7af54ca7b81e289985edc6cd864e7d05e0d133c
=> [internal] load build context
=> => transferring context: 389.08kB
=> CACHED [2/7] WORKDIR /app
=> CACHED [3/7] COPY package*.json ./
=> CACHED [4/7] RUN npm install
=> [5/7] COPY . .
=> [6/7] RUN npx prisma generate
=> [7/7] RUN npm run build
=> exporting to image
=> => exporting layers
=> => writing image sha256:abcad492d9e2f10f00de3b8c75f0f575528cfdc5246d1b26887f04ade3a2cab0
=> => naming to docker.io/library/products-id:latest

View build details: docker-desktop://dashboard/build/default/default/c6nn4y0rhc4x1a22ky42x7ekd

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2>
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker tag products-id himanshunanda22/project-6:products
-id
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker push himanshunanda22/project-6:products-id
The push refers to repository [docker.io/himanshunanda22/project-6]
ccf2a298688f: Pushed
52b87d3e6685: Pushed
0536088158a5: Pushed
44fa5c5c9bba: Layer already exists
14c2694f88e3: Layer already exists
72702b05020d: Layer already exists
123194e8ca81: Layer already exists
a28e61f75fa0: Layer already exists
41c2d1f0a1d3: Layer already exists
3e81cc85b636: Layer already exists
893507f6057f: Layer already exists
2353f7120e0e: Layer already exists
51a0318e6edf: Layer already exists
c5bb35826823: Layer already exists
products-id: digest: sha256:38dff88e29a9e82f143f0b6318df288906eeca14be5d861417e4f76a1142853b size: 3266
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2>
```

```
# Use an appropriate base image for Node.js applications
FROM node:20

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application code to the working directory
COPY . .

# Run prisma generate (if you are using Prisma in this microservice)
# If you are not using Prisma, you can remove this line
RUN npx prisma generate

# Build the application (if there is a build step, adjust the command accordingly)
RUN npm run build

# Expose the port the app runs on (assuming it's 3000, customize as needed)
EXPOSE 3000

# Command to run the application
CMD ["npm", "start"]
```

Cloud Computing Project 6 Deliverables 461-463-915-917

Product-user

```
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker build -t products-user:latest -f app/api/products-user/dockerfile .
[+] Building 0.0s (0/0) docker:default
[+] Building 29.8s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 800B
=> [internal] load metadata for docker.io/library/node:20
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/7] FROM docker.io/library/node:20@sha256:844b41cf784f66d7920fd673f7af54ca7b81e289985edc6cd864e7d05e0d133c
=> [internal] load build context
=> => transferring context: 903.80kB
=> CACHED [2/7] WORKDIR /app
=> CACHED [3/7] COPY package*.json ./
=> CACHED [4/7] RUN npm install
=> [5/7] COPY . .
=> [6/7] RUN npx prisma generate
=> [7/7] RUN npm run build
=> exporting to image
=> => exporting layers
=> => writing image sha256:a23127238b2929d56c2105c8e7d6bc240d2f1e0e39c18600a87a72068150a6b9
=> => naming to docker.io/library/products-user:latest

View build details: docker-desktop://dashboard/build/default/default/xy88x32vdt9rohek837phzi

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2>
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker tag products-user himanshunanda22/project-6:products-user
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker push himanshunanda22/project-6:products-user
The push refers to repository [docker.io/himanshunanda22/project-6]
b03e73944ffc: Pushed
471ea366c789: Pushed
426cfa323b08: Pushed
44fa5c5c9bba: Layer already exists
14c2694f88e3: Layer already exists
72702b05020d: Layer already exists
123194e8ca81: Layer already exists
a28e61f75fa0: Pushed
41c2d1f0a1d3: Pushed
3e81cc85b636: Layer already exists
893507f6057f: Layer already exists
2353f7120e0e: Layer already exists
51a9318e6edf: Layer already exists
c5bb35826823: Layer already exists
products-user: digest: sha256:5c086b97859927b87e7436c7024f46c15150e3421f914c72e0eae28b5bd93ec size: 3266
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2>
```

```
# Use an appropriate base image for Node.js applications
FROM node:20

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application code to the working directory
COPY . .

# Run prisma generate (if you are using Prisma in this microservice)
# If you are not using Prisma, you can remove this line
RUN npx prisma generate

# Build the application (if there is a build step, adjust the command accordingly)
RUN npm run build

# Expose the port the app runs on (assuming it's 3000, customize as needed)
EXPOSE 3000

# Command to run the application
CMD ["npm", "start"]
```

Cloud Computing Project 6 Deliverables

461-463-915-917

User

```
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker build -t user:latest -f app/api/User/[id]/Dockerfile .
[+] Building 36.1s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 800B
=> [internal] load metadata for docker.io/library/node:20
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/7] FROM docker.io/library/node:20@sha256:844b41cf784f66d7920fd673f7af54ca7b81e289985edc6cd864e7d05e0d133c
=> [internal] load build context
=> => transferring context: 701.53kB
=> CACHED [2/7] WORKDIR /app
=> CACHED [3/7] COPY package*.json ./
=> CACHED [4/7] RUN npm install
=> [5/7] COPY . .
=> [6/7] RUN npx prisma generate
=> [7/7] RUN npm run build
=> exporting to image
=> => writing image sha256:cdb5637ba5ace1f6b0ea014605435fe979b4f082e7bc8c76f134caf212d2c18a
=> => naming to docker.io/library/user:latest

View build details: docker-desktop://dashboard/build/default/default/nskjiqwk4e0m68cs7q1ph8zb

Each is a summary of image vulnerabilities and recommendations -> docker desktop quickstart
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker tag user himanshunanda22/project-6:user
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> docker push himanshunanda22/project-6:user
The push refers to repository [docker.io/himanshunanda22/project-6]
c1fda07a4cc6: Pushed
2ba7517200a3: Pushed
b31b21c8b3b1: Pushed
44fa5c5c9bba: Layer already exists
14c2694f88e3: Layer already exists
72702b05020d: Layer already exists
123194e8ca81: Layer already exists
a28e61f75fa0: Layer already exists
41c2d1f0a1d3: Layer already exists
3e81cc85b636: Layer already exists
893507f6057f: Layer already exists
2353f7120e0e: Layer already exists
51a9318e6edf: Layer already exists
c5bb35826823: Layer already exists
user: digest: sha256:0d9575390b9f5f0102882e9870fc992048e3ab2806e0a5adf2cf55d60ab138fc size: 3266
```

```
# Use an appropriate base image for Node.js applications
FROM node:20

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application code to the working directory
COPY . .

# Run prisma generate (if you are using Prisma in this microservice)
# If you are not using Prisma, you can remove this line
RUN npx prisma generate

# Build the application (if there is a build step, adjust the command accordingly)
RUN npm run build

# Expose the port the app runs on (assuming it's 3000, customize as needed)
EXPOSE 3000

# Command to run the application
CMD ["npm", "start"]
```


Cloud Computing Project 6 Deliverables

461-463-915-917

Kubernetes

Orchestration

```
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl apply -f kubernetes/auth/login/auth-login-deployment.yaml
deployment.apps/auth-login-deployment created
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl apply -f kubernetes/auth/login/auth-login-service.yaml
service/auth-login-service created
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
auth-login-deployment-7d44fdb7d9-4hkzf  1/1     Running   0           30s
auth-login-deployment-7d44fdb7d9-gh55s  1/1     Running   0           30s
auth-login-deployment-7d44fdb7d9-klrw2  1/1     Running   0           30s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get deployments
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
auth-login-deployment              3/3     3             3           34s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get services
NAME                                TYPE     CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
auth-login-service                 ClusterIP  10.104.82.233    <none>         80/TCP      32s
kubernetes                         ClusterIP  10.96.0.1         <none>         443/TCP     68d
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2>

PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl apply -f kubernetes/products/[id]/products-id-deployment.yaml
deployment.apps/products-id-deployment created
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl apply -f kubernetes/products/[id]/products-id-service.yaml
service/products-id-service created
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
auth-login-deployment-7d44fdb7d9-4hkzf  1/1     Running   0           14m
auth-login-deployment-7d44fdb7d9-gh55s  1/1     Running   0           14m
auth-login-deployment-7d44fdb7d9-klrw2  1/1     Running   0           14m
products-id-deployment-6f96dcb47-j2x9c  1/1     Running   0           11s
products-id-deployment-6f96dcb47-mxmkc  1/1     Running   0           11s
products-id-deployment-6f96dcb47-t4lp6  1/1     Running   0           11s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get deployments
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
auth-login-deployment              3/3     3             3           14m
products-id-deployment              3/3     3             3           16s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get services
NAME                                TYPE     CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
auth-login-service                 ClusterIP  10.104.82.233    <none>         80/TCP      14m
kubernetes                         ClusterIP  10.96.0.1         <none>         443/TCP     68d
products-id-service                 ClusterIP  10.100.243.55    <none>         80/TCP      14s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2>

PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl apply -f kubernetes/products/user/products-user-deployment.yaml
deployment.apps/products-user-deployment configured
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl apply -f kubernetes/products/user/products-user-service.yaml
service/products-user-service unchanged
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
auth-login-deployment-7d44fdb7d9-4hkzf  1/1     Running   0           38m
auth-login-deployment-7d44fdb7d9-gh55s  1/1     Running   0           38m
auth-login-deployment-7d44fdb7d9-klrw2  1/1     Running   0           38m
products-id-deployment-6f96dcb47-j2x9c  1/1     Running   0           24m
products-id-deployment-6f96dcb47-mxmkc  1/1     Running   0           24m
products-id-deployment-6f96dcb47-t4lp6  1/1     Running   0           24m
products-user-deployment-5c7ddb8f8f-cn45k  1/1     Running   0           8s
products-user-deployment-5c7ddb8f8f-jzg1p  1/1     Running   0           9s
products-user-deployment-5c7ddb8f8f-vjnhf  1/1     Running   0           10s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get services
NAME                                TYPE     CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
auth-login-service                 ClusterIP  10.104.82.233    <none>         80/TCP      38m
kubernetes                         ClusterIP  10.96.0.1         <none>         443/TCP     68d
products-id-service                 ClusterIP  10.100.243.55    <none>         80/TCP      24m
products-user-service               ClusterIP  10.108.30.127    <none>         80/TCP      2m
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get deployments
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
auth-login-deployment              3/3     3             3           38m
products-id-deployment              3/3     3             3           24m
products-user-deployment            3/3     3             3           2m7s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2>

PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl apply -f kubernetes/User/User-deployment.yaml
deployment.apps/user-deployment unchanged
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl apply -f kubernetes/User/User-service.yaml
service/user-service unchanged
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
auth-login-deployment-7d44fdb7d9-4hkzf  1/1     Running   0           47m
auth-login-deployment-7d44fdb7d9-gh55s  1/1     Running   0           47m
auth-login-deployment-7d44fdb7d9-klrw2  1/1     Running   0           47m
products-id-deployment-6f96dcb47-j2x9c  1/1     Running   0           33m
products-id-deployment-6f96dcb47-mxmkc  1/1     Running   0           33m
products-id-deployment-6f96dcb47-t4lp6  1/1     Running   0           33m
products-user-deployment-5c7ddb8f8f-cn45k  1/1     Running   0           9m35s
products-user-deployment-5c7ddb8f8f-jzg1p  1/1     Running   0           9m36s
products-user-deployment-5c7ddb8f8f-vjnhf  1/1     Running   0           9m37s
user-deployment-84866f86d8-fcpkk  1/1     Running   0           76s
user-deployment-84866f86d8-jhmf8  1/1     Running   0           77s
user-deployment-84866f86d8-zwpj7  1/1     Running   0           77s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get deployments
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
auth-login-deployment              3/3     3             3           47m
products-id-deployment              3/3     3             3           33m
products-user-deployment            3/3     3             3           11m
user-deployment                    3/3     3             3           80s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2> kubectl get services
NAME                                TYPE     CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
auth-login-service                 ClusterIP  10.104.82.233    <none>         80/TCP      47m
kubernetes                         ClusterIP  10.96.0.1         <none>         443/TCP     68d
products-id-service                 ClusterIP  10.100.243.55    <none>         80/TCP      33m
products-user-service               ClusterIP  10.108.30.127    <none>         80/TCP      11m
user-service                       ClusterIP  10.103.250.21     <none>         80/TCP      94s
PS C:\Users\himan\OneDrive\Desktop\Sem -VI Lab\CC\Project\PES2UG21CS461_463_915_917_Project-6\Week -2>
```

Code

User

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: user-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: user
  template:
    metadata:
      labels:
        app: user
    spec:
      containers:
        - name: user
          image: kushshriv/project-6:user
          ports:
            - containerPort: 3000
```

```
apiVersion: v1
kind: Service
metadata:
  name: user-service
spec:
  selector:
    app: user
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
```

Auth-login

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: auth-login-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: auth-login
  template:
    metadata:
      labels:
        app: auth-login
    spec:
      containers:
        - name: auth-login
          image: kushshriv/project-6:auth-login
          ports:
            - containerPort: 3000
```

```
apiVersion: v1
kind: Service
metadata:
  name: auth-login-service
spec:
  selector:
    app: auth-login
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
```

Products-id

```
kind: Deployment
metadata:
  name: products-id-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: products-id
  template:
    metadata:
      labels:
        app: products-id
    spec:
      containers:
        - name: products-id
          image: kushshriv/project-6:products-id
          ports:
            - containerPort: 3000
```

```
apiVersion: v1
kind: Service
metadata:
  name: products-id-service
spec:
  selector:
    app: products-id
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
```

Products-User

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: products-user-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: products-user
  template:
    metadata:
      labels:
        app: products-user
    spec:
      containers:
        - name: products-user
          image: kushshriv/project-6:products-user
          ports:
            - containerPort: 3000
```

```
apiVersion: v1
kind: Service
metadata:
  name: products-user-service
spec:
  selector:
    app: products-user
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
```

Cloud Computing Project 6 Deliverables

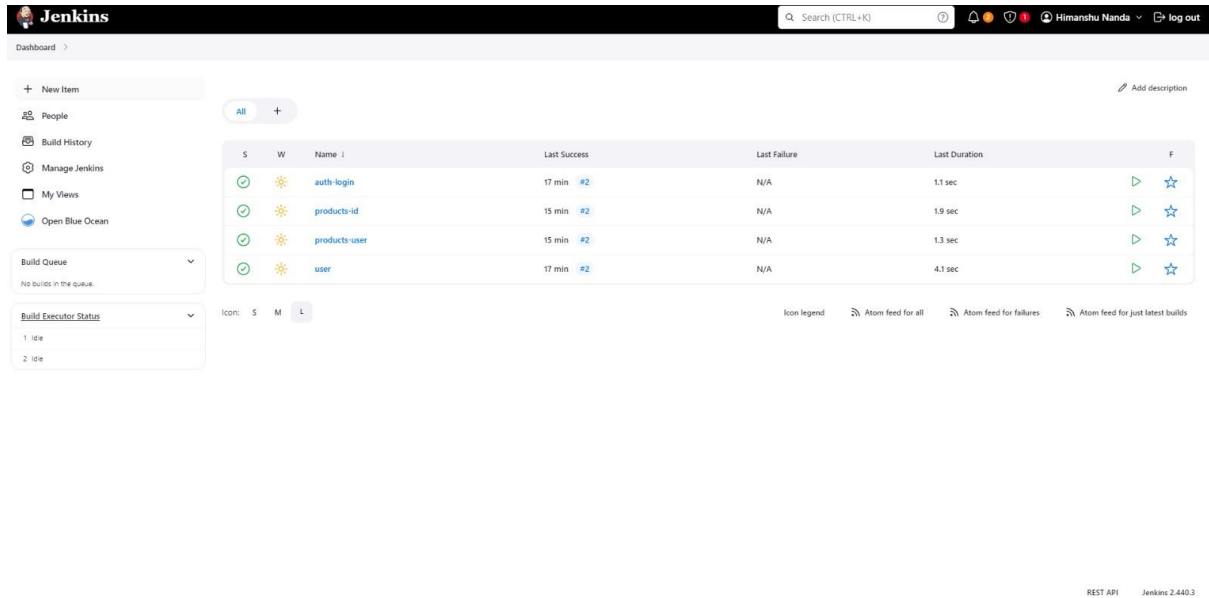
461-463-915-917

Week 3

Continuous Integration and Deployment

Overview

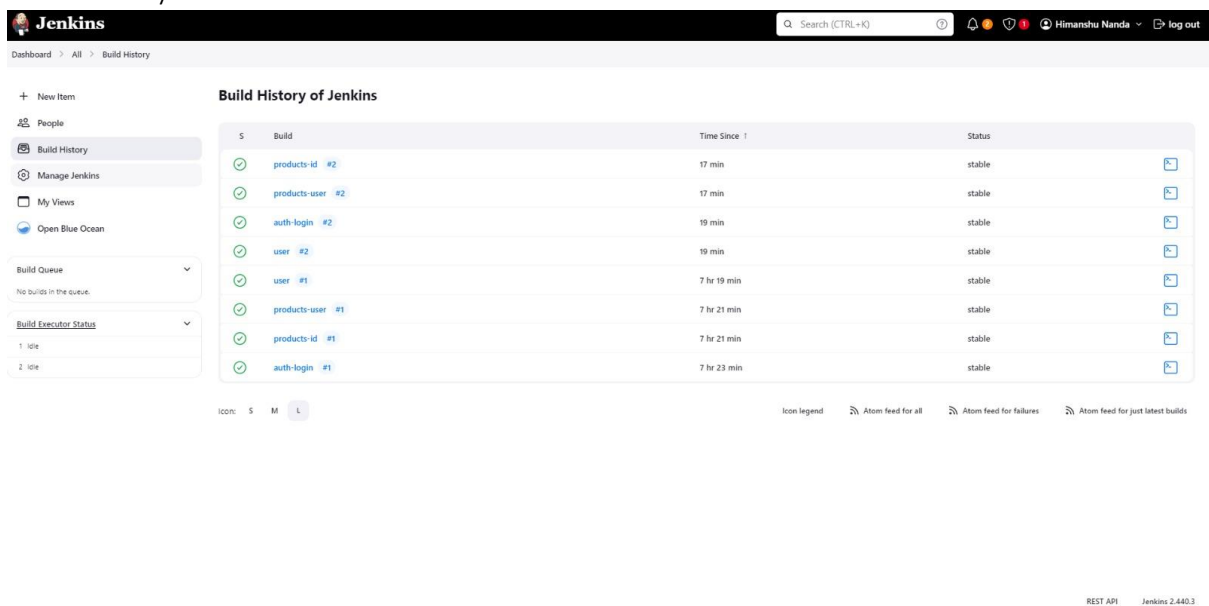
Dashboard



The Jenkins Dashboard shows the overall status of the CI/CD pipeline. It includes a sidebar with navigation links like 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'Open Blue Ocean'. The main area displays a table of recent builds with columns for status, name, last success, last failure, and last duration. Below the table, there are sections for 'Build Queue' and 'Build Executor Status'.

S	W	Name	Last Success	Last Failure	Last Duration	F
✓	☀	auth-login	17 min #2	N/A	1.1 sec	▶ ☆
✓	☀	products-id	15 min #2	N/A	1.9 sec	▶ ☆
✓	☀	products-user	15 min #2	N/A	1.3 sec	▶ ☆
✓	☀	user	17 min #2	N/A	4.1 sec	▶ ☆

Build History



The Jenkins Build History page provides a detailed view of the build history. It includes a sidebar with navigation links like 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'Open Blue Ocean'. The main area displays a table of build history with columns for status, build name, time since, and status. Below the table, there are sections for 'Build Queue' and 'Build Executor Status'.

S	Build	Time Since	Status
✓	products-id #2	17 min	stable
✓	products-user #2	17 min	stable
✓	auth-login #2	19 min	stable
✓	user #2	19 min	stable
✓	user #1	7 hr 19 min	stable
✓	products-user #1	7 hr 21 min	stable
✓	products-id #1	7 hr 21 min	stable
✓	auth-login #1	7 hr 23 min	stable

Cloud Computing Project 6 Deliverables

461-463-915-917

Files

Auth

The screenshot shows the Jenkins web interface. At the top, there's a navigation bar with the Jenkins logo, a search bar, and user information (Himanshu Nanda). Below the navigation bar, the breadcrumb trail reads 'Dashboard > auth-login > Git Polling Log'. On the left sidebar, there's a menu with options like Status, Changes, Workspace, Build Now, Configure, Delete Project, Git Polling Log (which is selected), GitHub, Favorite, Open Blue Ocean, and Rename. The main content area is titled 'Git Polling Log' and displays a log of Git operations. The log starts with 'Started on Apr 22, 2024, 4:51:00 PM' and shows the process of polling for changes on the 'main' branch. It includes details about the Git repository URL, the last built revision, and the steps taken to fetch and poll for changes. The log concludes with 'Done. Took 1.6 sec: No changes'. Below the log, there's a 'Build History' section showing a list of builds with their status and timestamps.

Jenkins 2.440.3

```
pipeline {
  agent any

  stages {
    stage('Checkout') {
      steps {
        checkout scm
      }
    }

    stage('Build') {
      steps {
        sh 'docker build -t auth-login:latest -f app/api/auth/login/Dockerfile .'
      }
    }

    stage('Test') {
      steps {
        // Add commands to run tests for the `auth` microservice
        sh 'npm run test'
      }
    }

    stage('Deploy') {
      steps {
        sh 'kubectl apply -f kubernetes/auth/auth-deployment.yaml'
        sh 'kubectl apply -f kubernetes/auth/auth-service.yaml'
      }
    }
  }
}
```

Cloud Computing Project 6 Deliverables

461-463-915-917

Products-id

Jenkins Search (CTRL+K) Himanshu Nanda log out

Dashboards > products-id > Git Polling Log

Git Polling Log

Started on Apr 22, 2024, 5:04:00 PM
Polling SCM changes on built-in
Using strategy: Default
[poll] Last Built Revision: Revision e2347232937dc48b7019384e40999e63f398105 (origin/main)
Selected git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/products-id/.git # timeout=10
Fetching changes from the remote Git repositories
> git config remote.origin.url https://github.com/HimanshuNanda22/PES2022CS461_463_915_917_Project-6.git # timeout=10
Fetching upstream changes from https://github.com/HimanshuNanda22/PES2022CS461_463_915_917_Project-6.git
> git --version # timeout=10
> git --version # 'git version 2.39.2'
> git fetch --tags --force --progress -- https://github.com/HimanshuNanda22/PES2022CS461_463_915_917_Project-6.git --refs/heads/*:refs/remotes/origin/* # timeout=10
Polling for changes in
Seen branch in repository origin/main
Seen 1 remote branch
> git show-ref --tags -d # timeout=10
Done. Took 1.1 sec
No changes

Build History trend
Filter... /
55
Apr 22, 2024, 4:40 PM
21
Apr 22, 2024, 5:44 AM
Alarm feed for all Alarm feed for failures

Jenkins 2.440.3

```
pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                checkout scm
            }
        }

        stage('Build') {
            steps {
                sh 'docker build -t products-id:latest -f app/api/products/[id]/Dockerfile .'
            }
        }

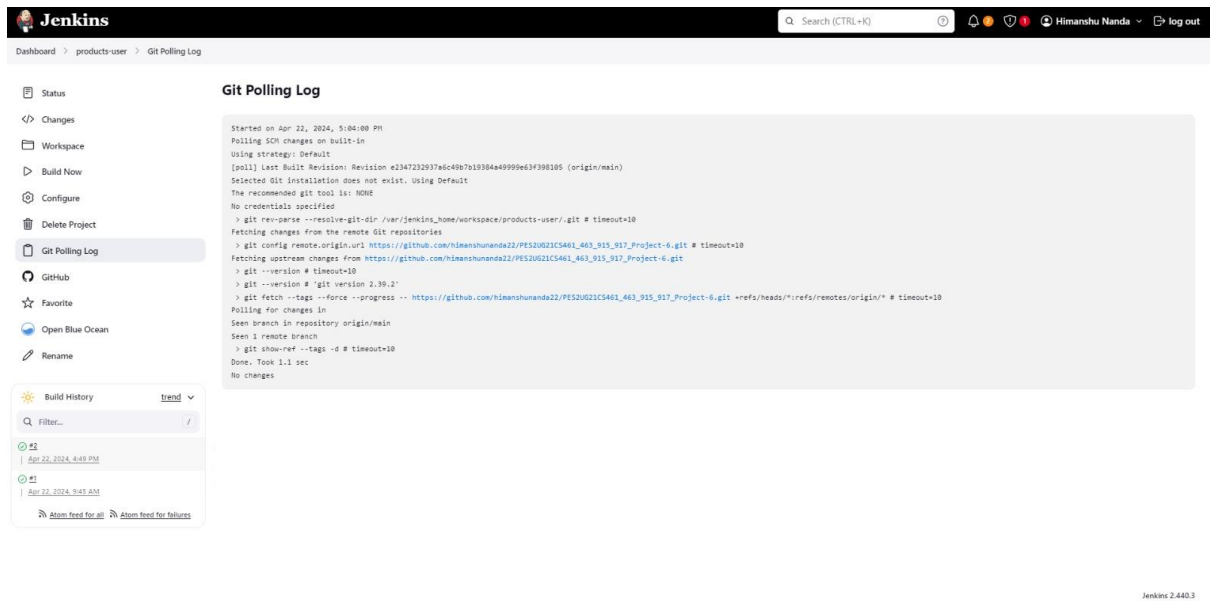
        stage('Test') {
            steps {
                // Add commands to run tests for the `products` microservice
                sh 'npm run test'
            }
        }

        stage('Deploy') {
            steps {
                sh 'kubectl apply -f kubernetes/products/[id]/products-id-deployment.yaml'
                sh 'kubectl apply -f kubernetes/products/[id]/products-id-service.yaml'
            }
        }
    }
}
```

Cloud Computing Project 6 Deliverables

461-463-915-917

Products-user



Jenkins Search (CTRL+K) Himanshu Nanda log out

Dashboard > products-user > Git Polling Log

Git Polling Log

Started on Apr 22, 2024, 5:04:00 PM
Polling SCM changes on built-in
Using strategy: Default
[poll] Last Built Revision: Revision e13472129376c40b7b19304e49999e63f398105 (origin/main)
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/products-user/.git # timeout=10
fetching changes from the remote Git repositories
> git config remote.origin.url https://github.com/Himanshunanda22/PES20221C5461_463_915_917_Project-6.git # timeout=10
fetching upstream changes from https://github.com/Himanshunanda22/PES20221C5461_463_915_917_Project-6.git
> git --version # timeout=10
> git fetch --tags --force --progress -- https://github.com/Himanshunanda22/PES20221C5461_463_915_917_Project-6.git --refs/heads/*:refs/remotes/origin/* # timeout=10
Polling for changes in
Seen branch in repository origin/main
Seen 1 remote branch
> git show-ref --tags -d # timeout=10
Done. Took 1.1 sec
No changes

Build History trend

Filter...

22
| Apr 22, 2024, 4:49 PM
21
| Apr 22, 2024, 9:45 AM

Atom feed for all Atom feed for failures

Jenkins 2.440.3

```
pipeline {
  agent any

  stages {
    stage('Checkout') {
      steps {
        checkout scm
      }
    }

    stage('Build') {
      steps {
        sh 'docker build -t products-user:latest -f app/api/products/user/Dockerfile .'
      }
    }

    stage('Test') {
      steps {
        // Add commands to run tests for the `products` microservice
        sh 'npm run test'
      }
    }

    stage('Deploy') {
      steps {
        sh 'kubectl apply -f kubernetes/products/user/products-user-deployment.yaml'
        sh 'kubectl apply -f kubernetes/products/user/products-user-service.yaml'
      }
    }
  }
}
```

Cloud Computing Project 6 Deliverables

461-463-915-917

User

The screenshot shows the Jenkins web interface. At the top, the Jenkins logo and navigation bar are visible. The main content area is titled "Git Polling Log". On the left sidebar, there are various navigation links: Status, Changes, Workspace, Build Now, Configure, Delete Project, Git Polling Log (selected), GitHub, Favorite, Open Blue Ocean, and Rename. Below these links is a "Build History" section showing a list of builds with their status and timestamps. The main log area displays the output of the Git polling process, including the start time, the repository URL, and the commands executed to fetch and checkout the code. The log shows that the repository was successfully updated with the latest changes from the remote.

```
Started on Apr 22, 2024, 5:01:00 PM
Polling SCM changes on built-in
Using strategy: Default
[pool1] Last Built Revision: Revision e23472329376c4907b19384a40999e03f98105 (origin/main)
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/user/.git # timeout=10
Fetching changes from the remote Git repositories
> git config remote.origin.url https://github.com/HimanshuNanda22/PE320221C5463_463_915_917_Project-6.git # timeout=10
Fetching upstream changes from https://github.com/HimanshuNanda22/PE320221C5463_463_915_917_Project-6.git
> git --version # timeout=10
> git fetch --tags --force --progress -- https://github.com/HimanshuNanda22/PE320221C5463_463_915_917_Project-6.git +refs/heads/*:refs/remotes/origin/* # timeout=10
Polling for changes in
Seen branch in repository origin/main
Seen 1 remote branch
> git show-ref --tags -d # timeout=10
Done. Took 3.5 sec
No changes
```

```
pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                checkout scm
            }
        }

        stage('Build') {
            steps {
                sh 'docker build -t user:latest -f app/api/User/[id]/Dockerfile .'
            }
        }

        stage('Test') {
            steps {
                // Add commands to run tests for the `user` microservice
                sh 'npm run test'
            }
        }

        stage('Deploy') {
            steps {
                sh 'kubectl apply -f kubernetes/User/User-deployment.yaml'
                sh 'kubectl apply -f kubernetes/User/User-service.yaml'
            }
        }
    }
}
```