

# DEVOPS

Assignment - 2

Name: Devathi, Srija  
Roll No: 20AT1A05F5  
Branch: CSE - III  
Section: C

1. Develop Jenkins pipelines with their Groovy scripts

agent any  
stages &

```
stage('Build') {  
  steps {
```

```
    sh 'mvn clean package'
```

```
  }
```

```
}
```

```
stage('Test') {
```

```
  steps {
```

```
    sh 'mvn test'
```

```
  }
```

```
}
```

```
stage('Deploy') {
```

```
  steps {
```

```
    sh 'scp target/My-app.jar  
user@server: /path/to/  
deploy'
```

```
  }
```

```
}
```

```
}
```

```
}
```

In this example, the pipeline has 3 stages

1. Build:

This stage compiles and packages code using Maven.

2. Test:

This stage runs unit testing using Maven.

3. Deploy:

This stage deploys the packaged artifact to a remote server using scp.

Discuss about Docker file creation along with docker image.

Docker is a tool allows you to create, deploy and run applications in containers. Docker images are templates for containers that contains all the necessary files, dependencies configurations to run an application.

A docker file is a script that contains instructions for building a docker image.

Basic Dockerfile:

```
# set the base image
from ubuntu:latest
# Install dependencies
RUN apt-get update && /
    apt-get install -y python3
# copy the application code
COPY app.py /app.py
# expose port 5000
EXPOSE 5000
# Define the command to run when the container
starts
CMD ["python3", "/app.py"]
```

Break Down of each line:

1. "FROM UBUNTU:latest" - sets the above base image to UBUNTU LINUX
2. "RUN apt-get update && \apt-get install -y python3" - installs the '1' directory.
3. "COPY app.py /app.py" - copies the 'app.py' file from the local directory to the '1' directory in the container.
4. "EXPOSE 5000" - expose port 5000, which is used by the application.
5. "CMD ["python3", "/app.py"]" - defines the command to run when the container starts, which is to run the app.py file using python3.

Discuss about Docker file creation along with docker image.

Docker is a tool allows you to create, deploy and run applications in containers. Docker images are templates for containers that contains all the necessary files, dependencies configurations to run an application.

A docker file is a script that contains instructions for building a docker image.

Basic Dockerfile:

```
# set the base image
from ubuntu:latest
# Install dependencies
RUN apt-get update && /
    apt-get install -y python3
# copy the application code
COPY app.py /app.py
# expose port 5000
EXPOSE 5000
# Define the command to run when the container
starts
CMD ["python3", "/app.py"]
```

Break Down of each line:

1. "FROM UBUNTU:latest" - sets the above base image to UBUNTU LINUX
2. "RUN apt-get update && apt-get install -y python3" - installs the 'python3' directory.
3. "COPY app.py /app.py" - copies the 'app.py' file from the local directory to the '/app.py' directory in the container.
4. "EXPOSE 5000" - expose port 5000, which is used by the application.
5. "CMD ["python3", "/app.py"]" - defines the command to run when the container starts, which is to run the app.py file using python3.



To create docker image from this docker file, you can use following command in the same directory as the docker file.

```
docker build -t my-app
```

This command builds the docker image using the dockerfile in the current directory and tags it with the name "my-app". Once the image is built you can run it using following command.

```
docker run -p 5000:5000 my-app
```

This command starts the container using the "my-app" image and maps port 5000 in the container to port 5000 on the host.

Verified

Sourav  
kolur