DevOps CI/CD Pipeline — Interview Questions, Answers & Examples

1■■ Jenkins Pipeline Stages

■ Explain the key stages in your Jenkins CI/CD pipeline.

My Jenkins pipeline automates the full CI/CD flow for a Flask web app using Docker and Ansible. Stages include Checkout, Build, Test, Push, and Deploy.

```
pipeline {
    agent any
    stages {
        stage('Build') { steps { sh 'docker build -t demo-web-application:latest .' } }
        stage('Test') { steps { sh 'docker run --rm demo-web-application:latest pytest -v
        stage('Push') { steps {
            withDockerRegistry([credentialsId: 'dockerhub-creds', url: '']) {
                sh 'docker tag demo-web-application:latest anil9182/demo-web-application:
                sh 'docker push anil9182/demo-web-application:latest'
            } } }
        stage('Deploy') { steps {
            sh 'ansible-playbook ansible/site.yml -i ansible/inventory -e target_env=sit'
        } }
    }
}
```

2■■ Docker Image Build & Test

■ How do you ensure your Docker image is tested before deployment?

I test the image using pytest inside the container to ensure consistency between build and runtime environments.

```
docker build -t demo-web-application:latest .
docker run --rm demo-web-application:latest pytest -v
```

3■■ Docker Push Failure

■ Why might 'unauthorized: access token has insufficient scopes' occur when pushing Docker images?

The Jenkins credentials or Docker Hub token lacked push permissions. I generated a new token with write access and added it to Jenkins credentials.

```
withDockerRegistry([credentialsId: 'dockerhub-creds', url: 'https://index.docker.io/v1/']
    sh 'docker push anil9182/demo-web-application:latest'
}
```

4■■ Python Dependency Error (PEP 668)

■ What is 'Externally managed environment' and how do you fix it?

Debian-based Python images restrict system-wide pip installs. Use a virtual environment or override protection with:

```
RUN pip install --no-cache-dir --break-system-packages -r requirements.txt
```

5■■ Pytest Not Found

■ Why did your container fail with 'exec: pytest: not found'?

Pytest was missing in the image. Add it to requirements.txt:

```
Flask==3.0.0
pytest==7.4.4
```

6■■ Ansible Inventory Not Found

■ Why did Ansible show 'Unable to parse inventory as an inventory source'?

Inventory file was missing or invalid. Correct fix:

```
[sit]
192.168.56.10 ansible_user=ubuntu ansible_ssh_private_key_file=~/.ssh/id_rsa
```

7■■ Dynamic Environment Deployment

■ How do you deploy to SIT, UAT, and PROD using the same playbook?

Use Jenkins parameters and pass environment dynamically to Ansible.

```
parameters {
    choice(name: 'TARGET_ENV', choices: ['sit', 'uat', 'prod'], description: 'Select envi
}
sh "ansible-playbook ansible/site.yml -i ansible/inventory -e target_env=${params.TARGET_
```

8■■ Ansible Role-Based Deployment

■ Why do you use roles in Ansible?

Roles make playbooks modular and reusable. Example structure:

```
ansible/
■■■ site.yml
■■■ roles/
    ■■■ deploy/
        ■■■ tasks/main.yml
```

9■■ Jenkinsfile Groovy Error

■ What causes 'No such property: install for class: groovy.lang.Binding'?

Shell commands must be wrapped with sh keyword.

```
# Wrong:
install requirements.txt

# Correct:
sh 'pip install -r requirements.txt'
```

■ Troubleshooting Jenkins + Ansible

■ How do you debug Ansible from Jenkins?

Run commands manually to isolate issue and verify SSH access.

```
ansible-playbook ansible/site.yml -i ansible/inventory -e target_env=sit -vvvv
ansible all -m ping -i ansible/inventory
```

■ CI/CD Best Practices Summary

- Always build Docker images via Jenkins agents, not manually.
- Run Pytest inside the same Docker container used for deployment.
- Tag Docker images with build numbers or Git commit hashes.
- Use Ansible roles for modular deployment structure.
- Secure Docker and SSH credentials inside Jenkins credentials store.
- Maintain version control for Jenkinsfile, Dockerfile, and Ansible playbooks.