Hands-on: Jenkins Pipeline with Sample Code

🔧 **Pipeline Flow: Build → Unit Test → Code Lint → Artifact**

✅ Prerequisites:

- Jenkins with:
  - Pipeline plugin
  - Git plugin
  - Jenkins agent with Python 3 installed
- Bitbucket repo access
- Bitbucket personal access token added to Jenkins credentials

---

📁 Sample Project Structure

```
sample-pipeline-app/
├── Jenkinsfile
├── requirements.txt
├── app.py
├── tests/
│   └── test_app.py

```

🧾 Sample Files

`app.py`

```
def add(a, b):
    return a + b

if __name__ == "__main__":
    print("Sum is:", add(2, 3))

```

`tests/test_app.py`

```
from app import add

def test_add():
    assert add(2, 3) == 5

```

`requirements.txt`

```
pytest
flake8
```

```

```

`Jenkinsfile`

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Installing dependencies...'
                sh 'pip install -r requirements.txt'
            }
        }

        stage('Unit Test') {
            steps {
                echo 'Running unit tests...'
                sh 'pytest tests/'
            }
        }

        stage('Lint') {
            steps {
                echo 'Checking code style...'
                sh 'flake8 app.py'
            }
        }

        stage('Package') {
            steps {
                echo 'Creating artifact...'
                sh 'zip -r app.zip app.py tests/'
                archiveArtifacts artifacts: 'app.zip', fingerprint: true
            }
        }
    }
}

```

🧪 Jenkins Job Setup

Step 1: Create Repository

- Push the above files to a new Bitbucket Enterprise repo: `sample-pipeline-app`.

Step 2: Create Jenkins Pipeline Job

1. Jenkins Dashboard → **New Item**
2. Name: `sample-pipeline-job`
3. Type: **Pipeline** → Click OK
4. Under **Pipeline Definition**, choose:
    - **Pipeline script from SCM**
    - SCM: Git

    - Repository URL: Your Bitbucket repo
    - Credentials: Add Bitbucket token
    - Script Path: `Jenkinsfile`
5. Save

Step 3: Run the Pipeline

- Go to the job page → Click **Build Now**
- Observe console logs for:
    - Python packages installed
    - Unit test output from `pytest`
    - Style check output from `flake8`
    - Zip file `app.zip` created and archived

✅ Outcome:

✔️ Complete Jenkins Pipeline with code setup
✔️ Unit test executed via `pytest`
✔️ Code linting via `flake8`
✔️ Build artifact archived