






-  _____
-  Jenkins  _____



Continuous Integration, CI



1.

--	--	--	--	--	--	--	--	--

- [illegible]

2. CI/CD ☐

- **Jenkins**
- **GitHub Actions** GitHub YAML
- **GitLab CI/CD** GitLab
- **CircleCI**
- **Travis CI** CI













3.

--	--	--	--	--	--	--

- **Maven/Gradle** Java
- **npm/Yarn** JavaScript
- **pip/Poetry** Python
- **Docker**

4.

--	--	--	--

-      
-    
-  

5.

--	--	--	--	--

- **Kubernetes** k8s ██████████
- **Terraform** ████████ IaC █
- **Ansible/Chef** ████████



CI Pipeline

```
graph LR
    A[Source Code] --> B[Build]
    B --> C[Test]
    C --> D[Package]
    D --> E[Deploy]
    E --> F[Monitor]
    F --> G{Feedback}
    G --> A
    G --> H[Rollback]
    G --> I[Cleanup]
    H --> J[Rebuild]
    J --> B
    I --> K[Log]
    K --> L[Report]
```



1. [] [] Git Hook[]
2. [] [] Docker[]
3. [] []
 - []
 - []
 - E2E[]
4. [] [] Docker[] JAR[] Nexus[] Harbor[]
5. [] [] []



1. GitHub Actions Pipeline

```
# .github/workflows/ci.yml
name: CI Pipeline
on: [push]
jobs:
  build-and-test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
```

- run: npm install
- run: npm run build
- run: npm test

deploy:

needs: build-and-test

runs-on: ubuntu-latest








steps:

- run: kubectl apply -f k8s/deployment.yaml

2 Jenkins + Docker

```
// Jenkinsfile
pipeline {
  agent { docker 'maven:3.8.6' }
  stages {
    stage('Build') {
      steps {
        sh 'mvn clean package'
      }
    }
    stage('Test') {
      steps {
        sh 'mvn test'
      }
    }
    stage('Deploy') {
      steps {
        sh 'docker build -t myapp .'
        sh 'docker push myrepo/myapp'
      }
    }
  }
}
```



1.   5 
2.  
3.  

4. ☐ ☐ ☐ ☐ ☐ ☐ ELK ☐ ☐ ☐ ☐ ☐ Prometheus ☐ ☐ ☐ ☐ ☐
5. ☐ ☐ ☐ ☐ ☐ ☐ Cl ☐ ☐ ☐ Trivy ☐ ☐ ☐ ☐ ☐ ☐ ☐ OWASP ZAP ☐ ☐ ☐ ☐ ☐ ☐



- **CI** API
- **CI** Fastlane iOS/Android Firebase Test Lab
- **Serverless** AWS CodePipeline + Lambda



📦 Jenkins 📦📦📦

📦📦📦 **Jenkins** 📦📦 📦📦📦📦📦 📦📦📦📦📦📦📦 📦 📦📦📦📦 📦📦
Linux/Windows 📦📦

1. 📦 Jenkins

Linux (Ubuntu/CentOS)

```
# 📦 Java (Jenkins 📦 )
sudo apt update && sudo apt install openjdk-11-jdk -y # Ubuntu
sudo yum install java-11-openjdk -y # CentOS

# 📦 Jenkins 📦📦📦
curl -fsSL https://pkg.jenkins.io/debian/jenkins.io.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc >
/dev/null
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian binary/" | sudo tee
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt update && sudo apt install jenkins -y

# 📦 Jenkins
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

Windows

- 1. 📦 Jenkins Windows Installer 📦📦
 - 2. 📦📦📦📦📦
-

2. 📦📦📦

- 1. 📦 **Jenkins**
📦
 📦


```

    steps {
        git 'https://github.com/your-repo.git'
    }
}
stage('Build') {
    steps {
        sh 'mvn clean package'
    }
}
stage('Test') {
    steps {
        sh 'mvn test'
    }
}
stage('Deploy') {
    steps {
        sh 'docker build -t myapp .'
        sh 'docker push myrepo/myapp'
    }
}
}
}

```

5. Docker Kubernetes

Docker

-  Jenkins  Docker  Jenkins  docker 

```

sudo usermod -aG docker jenkins
sudo systemctl restart jenkins

```

-  Pipeline  Docker 

```

stage('Docker Build') {
    steps {
        script {
            docker.build("myapp:${env.BUILD_ID}").push()
        }
    }
}

```






Kubernetes

1. ☐ **Kubernetes CLI (kubectl)** ☐
2. ☐ Pipeline ☐

```
stage('Deploy to k8s') {
    steps {
        sh 'kubectl apply -f k8s/deployment.yaml'
```

6.

GitHub Webhook




1.  > Settings > Webhooks 
 - Payload URL: `http://<Jenkins-IP>:8080/github-webhook/`
 - Content type: `application/json`
2.  **"GitHub hook trigger for GITScm polling"** 

Jenkinsfile

1.
2. Jenkins **"Pipeline script from SCM"**

7.

--	--	--	--	--

-                                                                              

8. Jenkins

- Jenkins 安装
 - sudo chown -R jenkins:jenkins /var/lib/jenkins
- Jenkins 配置
 - Manage Jenkins > Plugin Manager > Advanced Update Site
- Jenkins 启动
 - sudo systemctl edit jenkins
 - # Environment="JENKINS_PORT=8081"
 - sudo systemctl restart jenkins

部署

- Jenkins 部署 Docker Jenkins
- Jenkins 部署 Pipeline Jenkins
- Jenkins 部署 Docker/K8s GitHub Webhook Jenkins

部署 Jenkins 到 Kubernetes 集群