

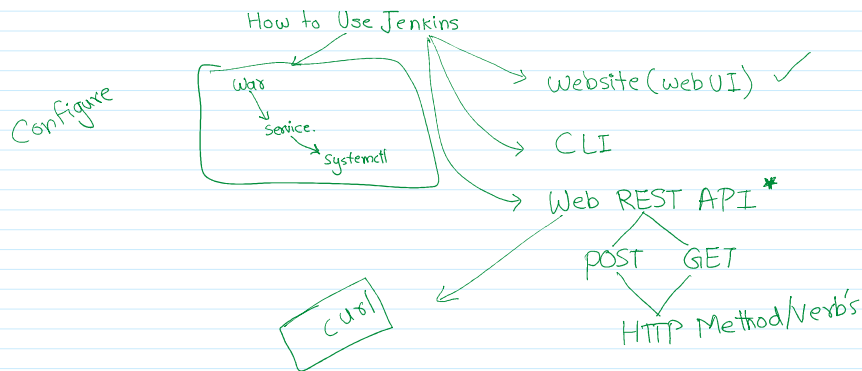
58.20Jun.Jenkins Integrations(GitJavaMvnDocker)

Tuesday, June 20, 2023 7:08 PM

- A. Jenkins Integrations with:
 - a. Github
 - b. Java
 - c. Maven
 - d. Docker
 - e. Ansible
- B. Minikube (AWS/GCloud Shell/PWK)
- C. K8S CRUD App Deployments

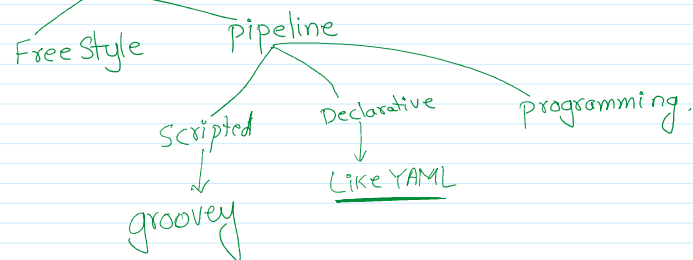
CI Tool

- Jenkins (80%)
 - Open Source.
 - Distribution
 - Java → Jenkins

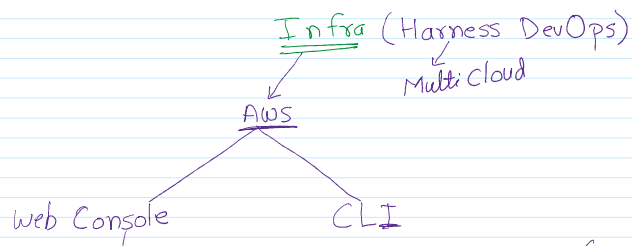
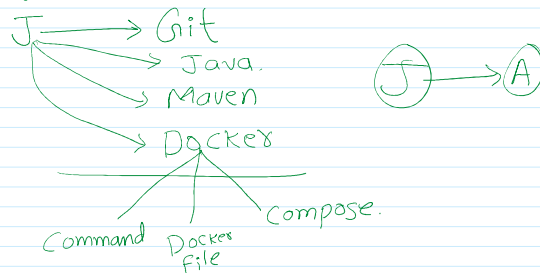


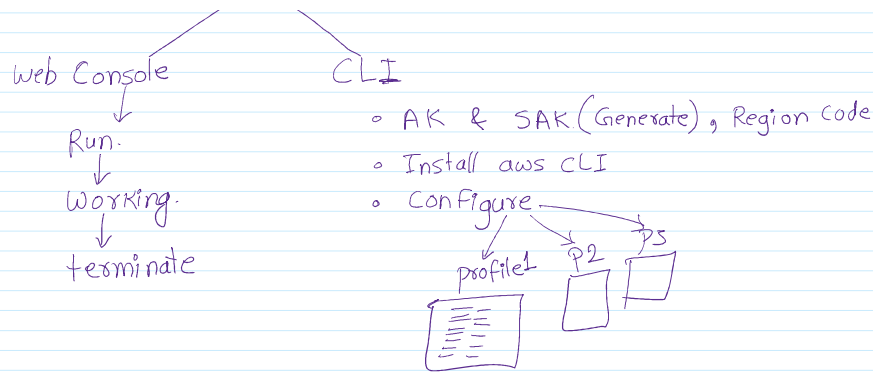
1] way to Deployment's

2] Working with Jenkins Job



3] Integrations





```
$ msixexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
```

```
$ aws --version
```

```
$ aws configure --profile p1
AWS Access Key ID [ ]: *****IZ7V
AWS Secret Access Key [ ]: *****TNBm]
Default region name [ ]: ap-south-1
Default output format [ ]: json
```

```
$ aws s3 ls --profile p1
```

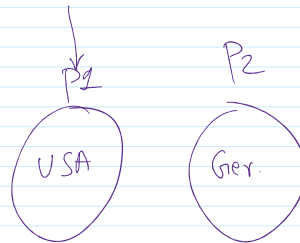
```
$ aws ^
s3 ls ^
--profile mir
```

For Unix(Linux/Mac) EC2 Instance:

```
$ aws ec2 run-instances \
--image-id ami-0f5ee92e2d63afc18 \
--count 1 \
--instance-type t2.micro \
--key-name mujahed \
--security-group-ids sg-04fb9b54c6f43819a \
--subnet-id subnet-0f8384b366fe16fe9 \
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=JenkinsInt}]' \
--profile mir
```

For Windows EC2 Instance:

```
$ aws ec2 run-instances ^
--image-id ami-0f5ee92e2d63afc18 ^
--count 1 ^
--instance-type t2.micro ^
--key-name mujahed ^
--security-group-ids sg-04fb9b54c6f43819a ^
--subnet-id subnet-0f8384b366fe16fe9 ^
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=JenkinsInt}]' ^
--profile mir
```



INSTALL JENKINS IN UBUNTU

```
#!/bin/bash
```

```
# Update packages
sudo apt update
```

```
#Check Version of Java
apt-cache search openjdk
```

```
# Install Java
sudo apt install -y openjdk-11-jdk
```

```
# Install Jenkins dependencies
sudo apt install -y git
sudo apt install -y maven
```

```
# Install Jenkins
# sudo wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

```
# sudo echo "deb https://pkg.jenkins.io/debian-stable binary/" >> /etc/apt/sources.list.d/jenkins.list
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
```

```
# Check Jenkins Version
$ apt-cache madison jenkins
```

```
sudo apt install -y jenkins #Jenkins 2.410
sudo apt install -y jenkins=2.410
```

```
# Start Jenkins and enable it to start on boot
```

```
sudo systemctl status jenkins
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

```
pipeline {
  agent any

  stages {
    stage('Greeting') {
      steps {
        echo 'Welcome'
      }
    }
    stage('Check Java') {
      steps {
        sh 'java -version'
        sh 'javac -version'
      }
    }
  }
}
```

```
pipeline {
  agent any

  stages {
    stage('Greeting') {
      steps {
        echo 'Welcome'
      }
    }
    stage('Check Java') {
      steps {
        sh 'java -version'
        sh 'javac -version'
      }
    }
    stage('Check Git') {
      steps {
        sh 'git --version'
      }
    }
  }
}
```

Installing Maven

```
$ apt update && apt install -y maven
$ mvn --version
```

```
pipeline {
  agent any

  stages {
    stage('Greeting') {
      steps {
        echo 'Welcome'
      }
    }
    stage('Check Java') {
      steps {
        sh 'java -version'
        sh 'javac -version'
      }
    }
    stage('Check Git') {
      steps {
        sh 'git --version'
      }
    }
    stage('Check Maven') {
      steps {
        sh 'mvn --version'
      }
    }
  }
}
```

Installing Docker

```
$ sudo apt update
$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
$ apt-cache policy docker-ce
$ apt install -y docker-ce
$ docker info
```

Install Jenkins Plugins

Docker

Docker Commons

Install Jenkins Plugins

Docker
Docker Commons
Docker Pipeline
Docker Compose Build Step

```
usermod -aG docker jenkins
usermod -aG root jenkins
chmod 777 /var/run/docker.sock
```

```
$ cat > Dockerfile
FROM alpine
CMD ["echo", "Welcome to DevOps"]
```

\$ docker images

\$ docker build -t hello . #Read Dockerfile and Create Custom Image based on Dockerfile

\$ docker run --rm hello # Run Container from Custom Image

```
pipeline {
  agent any

  stages {
    stage('Write File') {
      steps {
        script {
          def date = new Date()
          def data = "First Line\nSecond Line\n" + date
          writeFile(file: 'welcome.txt', text: data)
          sh "ls -l"
        }
      }
    }
  }
}
```

Create Dockerfile using Jenkins Pipeline

```
pipeline {
  agent any

  stages {
    stage('Create Dockerfile') {
      steps {
        script {
          def cmd = "CMD ['echo', 'Welcome from NubeEra!']"
          def data = "FROM alpine\n" + cmd + "\n"
          writeFile(file: 'Dockerfile', text: data)
          sh "ls -l"
        }
      }
    }
  }
}
```

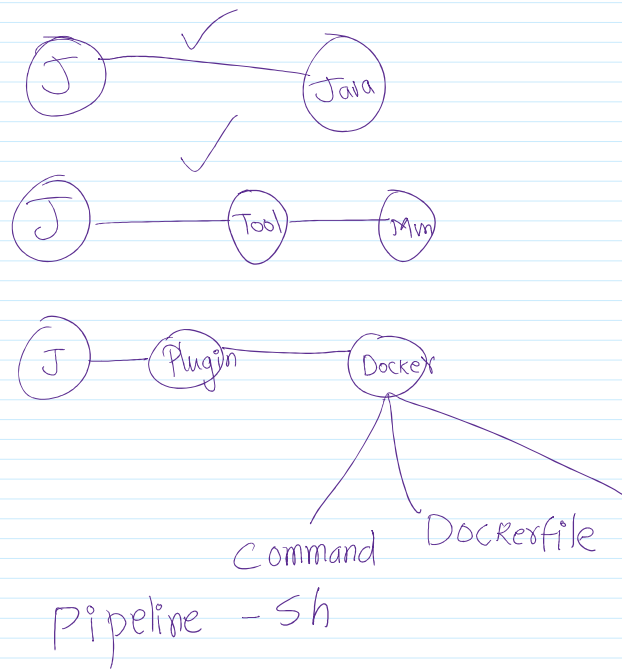
Docker file, Docker Build and Run

```
pipeline {
  agent any

  stages {
    //Create Dockerfile
    stage('Create Dockerfile') {
      steps {
        script {
          def cmd = "CMD ['echo', 'Welcome from NubeEra!']"
          def data = "FROM alpine\n" + cmd + "\n"
          writeFile(file: 'Dockerfile', text: data)
          sh "ls -l"
        }
      }
    }
    //Docker Build and Run
    stage('docker build and run') {
      steps {
        sh "docker build -t hello ."
        sh "docker run --rm hello"
      }
    }
  }
}
```

free -m

df -H



Terminate EC2 Instance using CLI:
i-0c52e0afac5d0258e