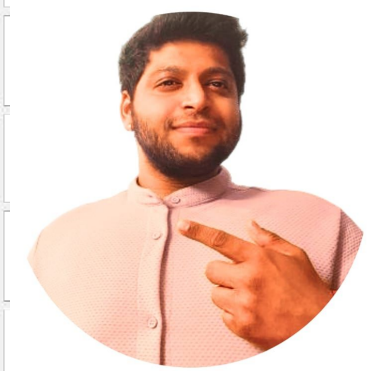
WORLD OF

Jenkins

About me – Karan

- I'm Karan Gupta
- I'm 5x AWS certified, 2x Azure, CKA certified
- Have worked with startups, Mid-Size and MNCs as a Devops Engineer.
- Published over 10+ research papers

- In free time, I love travel to new destination
- Workout enthusiast



Course Prerequisites

Karan Gupta

- **Level**
  - All (Freshers, Experienced, Professional)

- **Hardware/System Requirement**
  - Any OS will be fine
  - 256 MB of RAM, although more than 2 GB is recommended
  - 10 GB of drive space (for Jenkins and your Docker image)

- **Practice as much as possible.**
- **Study and learn at you pace.**

Karan Gupta

Contents

Karan Gupta

Contents

Karan Gupta

Bonus

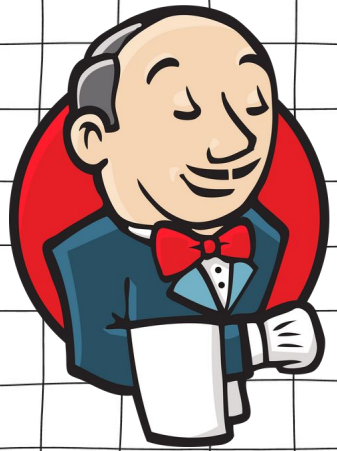
01 Freestyle Project

02 Docker Deployment

03 Kubernetes Deployment

Karan Gupta

Real-Time



Jenkins

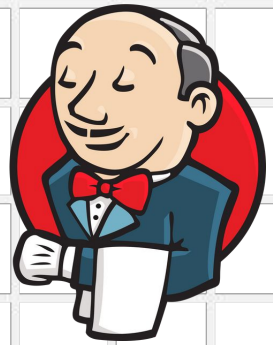
Industry based project

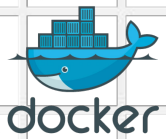
Major Project

Karan Gupta

Tools

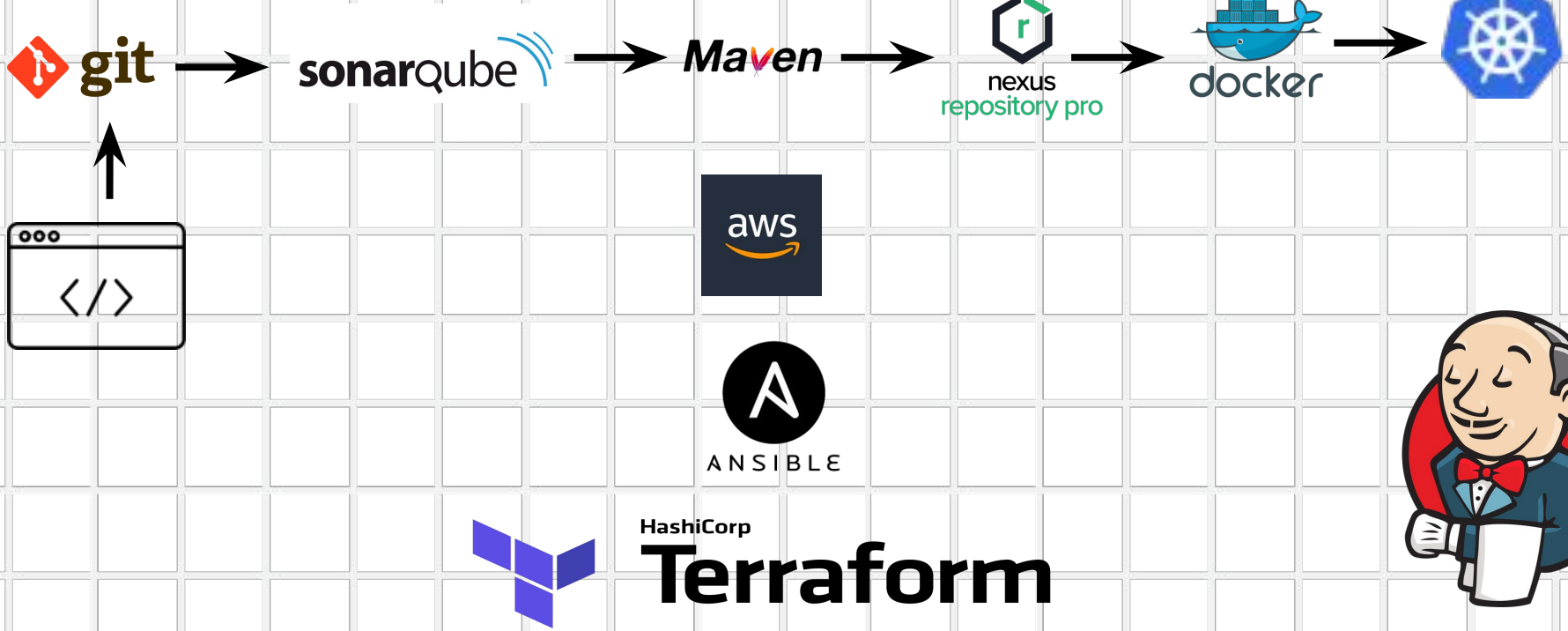
- Git + GitHub
- AWS
- Jenkins
- Maven
- Docker
- Sonarqube
- Nexus
- Kubernetes (EKS)
- Terraform
- Ansible





Architecture

Karan Gupta



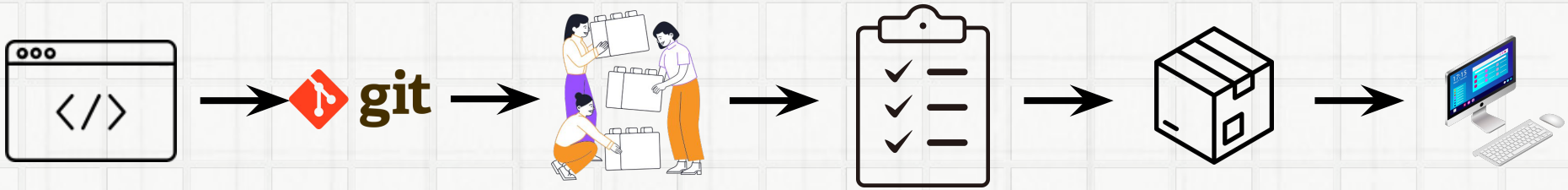
Introduction

- Jenkins is an open-source automation server that enables developers to build, test, and deploy code continuously.
- One of the most widely used automation tools for CI and CD.

Automation

Karan Gupta

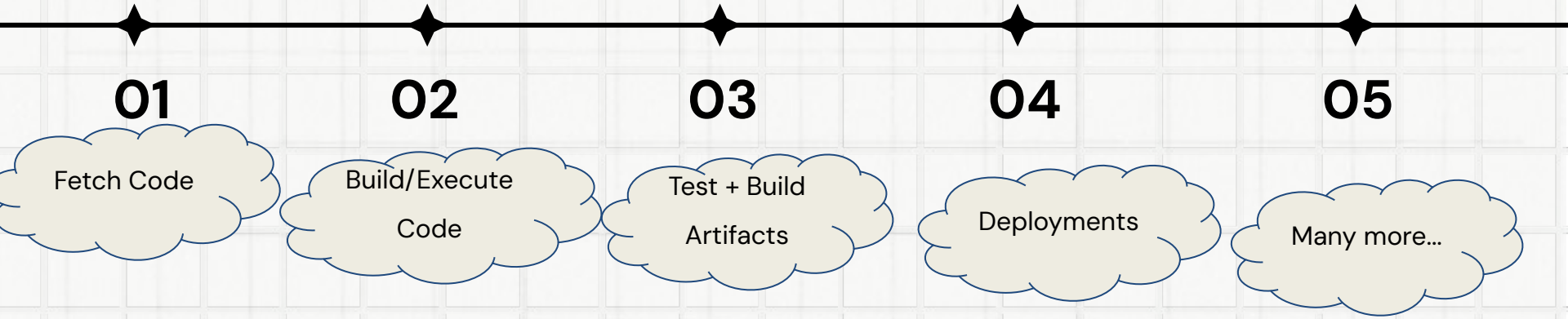
- Process of using tools and technologies to perform tasks automatically, without manual intervention



Jenkins process

Karan Gupta

- Jenkins plays a crucial role in DevOps by automating continuous integration and continuous delivery (CI/CD) processes



CI/CD

Continuous Integration

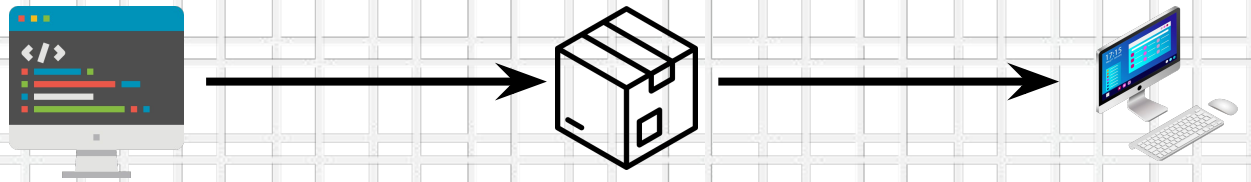
Process to get code and package it; then sent to CD for further processing

Continuous Delivery

Process in which package is deployed on system after MANUAL trigger.

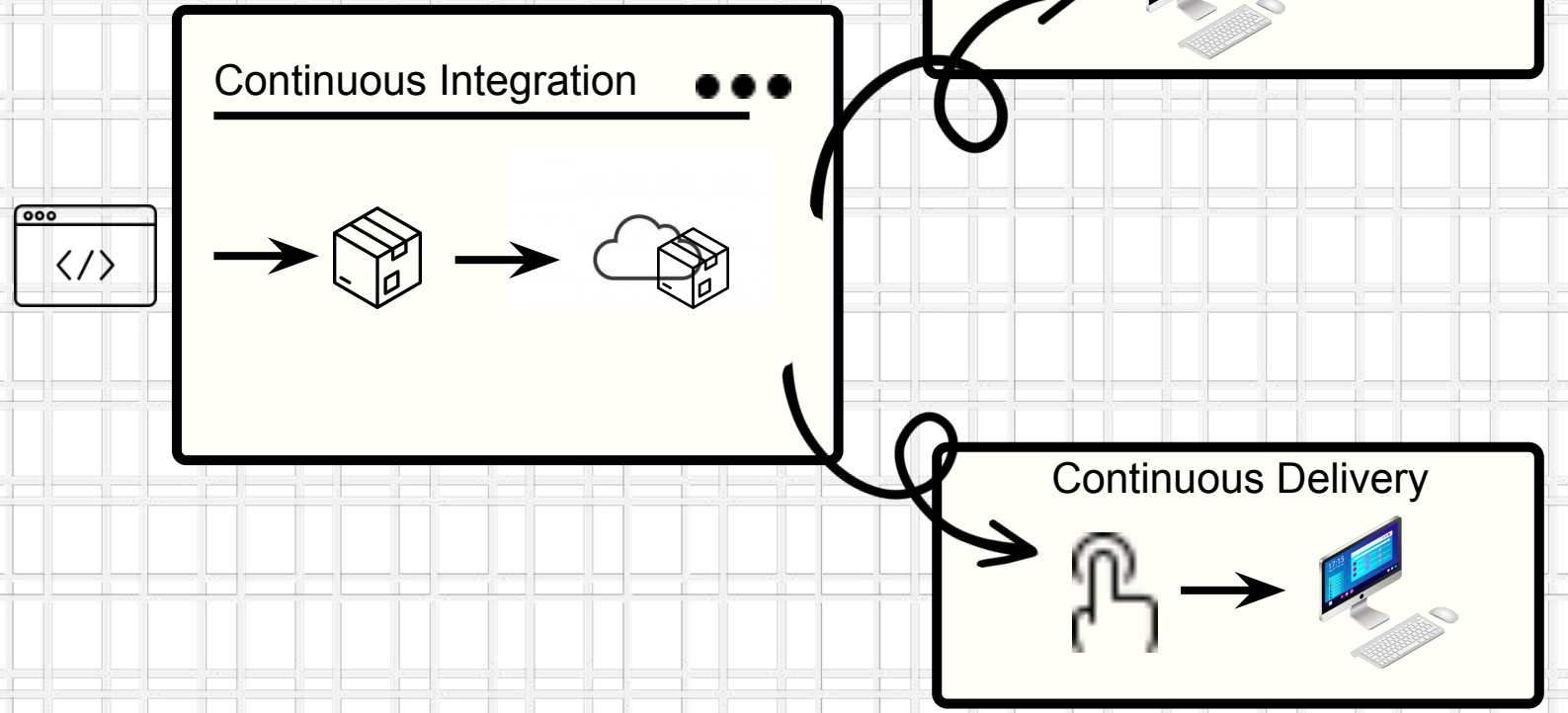
Continuous Deployment

Process in which package is deployed on system AUTOMATICALLY.



Karan Gupta

Karan Gupta


Continuous Deployment
Continuous Integration
Continuous Delivery

Need

Challenges Jenkins
Solves / Benefits

Karan Gupta

Slower and Manual Builds		Automated and faster builds
Inefficient releases and Inconsistent environment		Efficient and Faster Releases
Non-scalable And Limited Visibility		Scalability and Extensibility

WHY JENKINS

Karan Gupta

BENEFITS

- Free / Enterprise Version
- Open-Source
- Plugins
- Easy and Portable

ALTERNATIVES

- Github Actions
- Gitlab CI/CD
- Codebuild
- TravisCI

GitHub Actions

GitLab

Karan Gupta

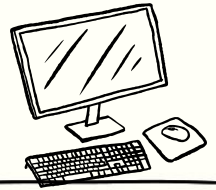
Installation

Dedicated Server

On any cloud AWS, AZURE, GCP or any other

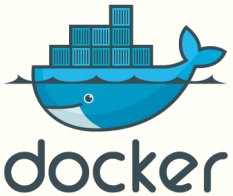
Locally

Can be installed on any OS locally.

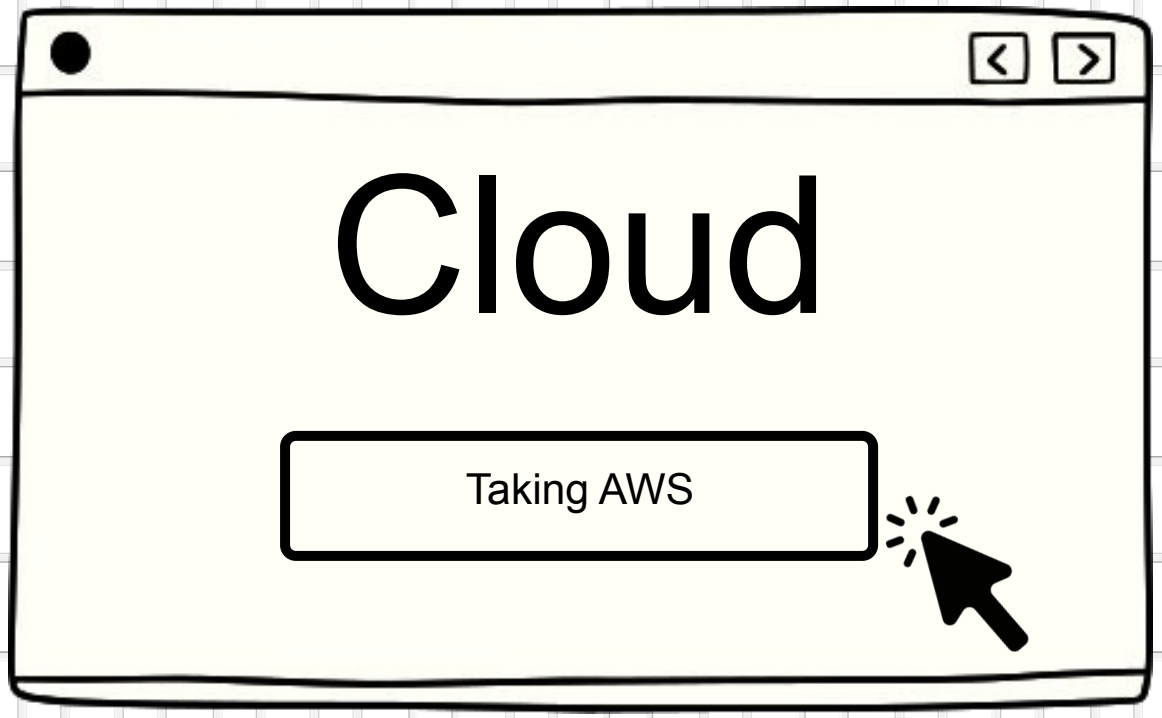


Docker

Jenkins as Containers



Karan Gupta



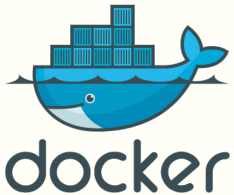
Karan Gupta

Local

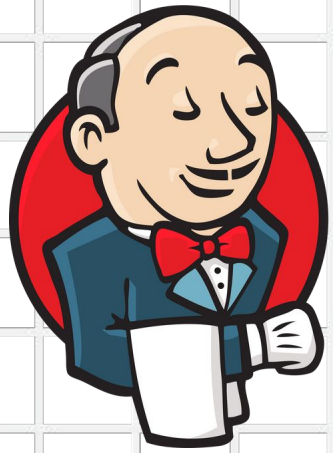
Taking Mac

Karan Gupta

Docker



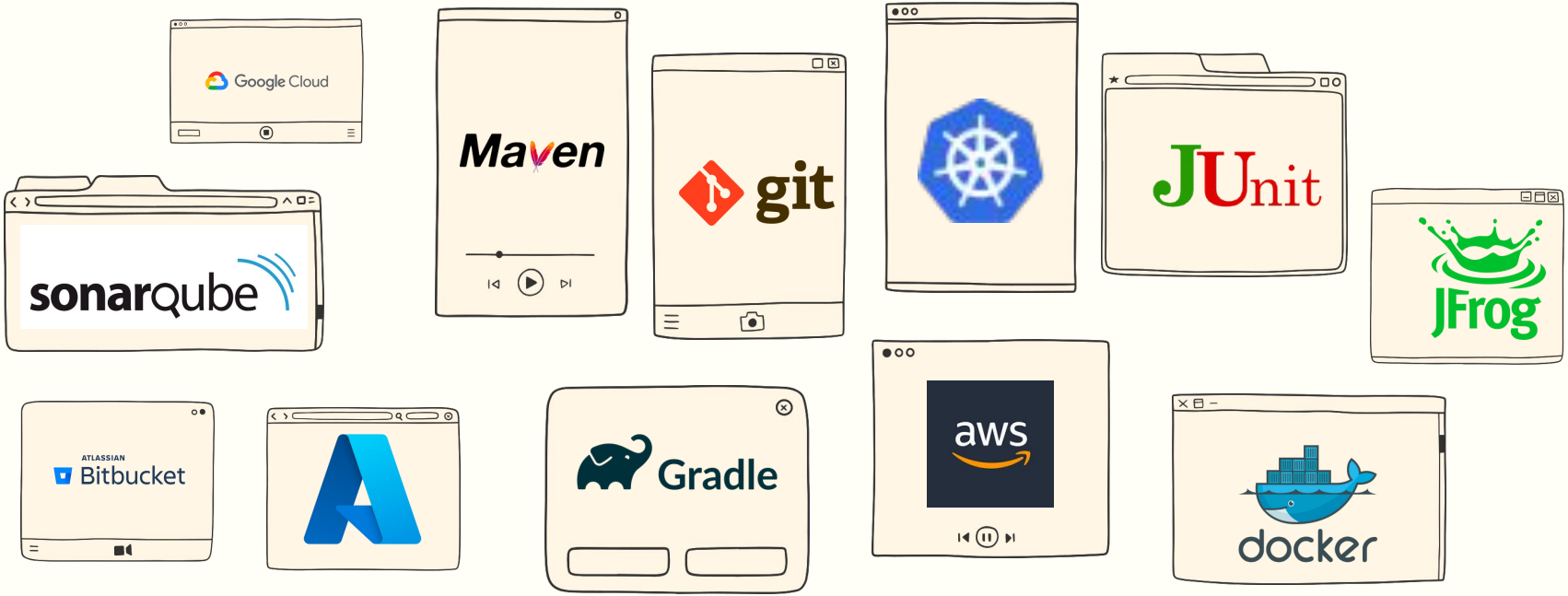
Karan Gupta



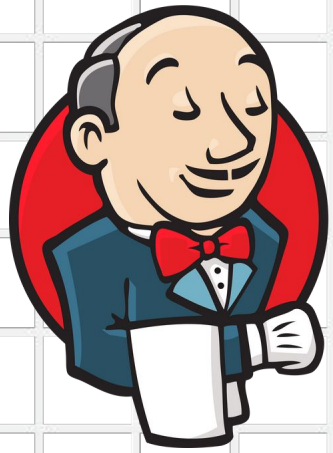
Overview of UI

Plugins

Karan Gupta



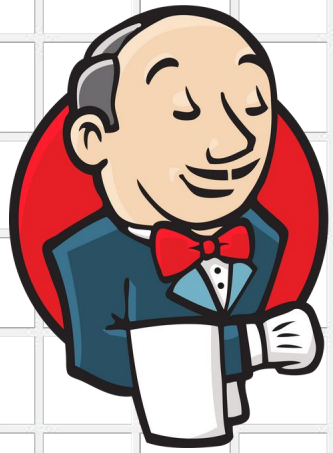
Karan Gupta



Plugins

Installation and Updation

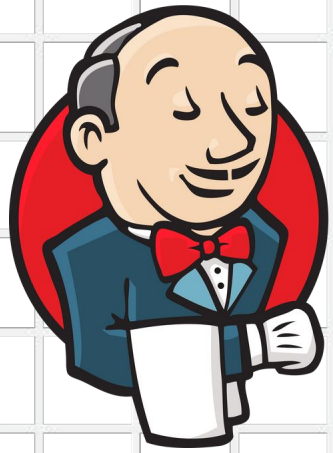
Karan Gupta



Plugins

Installation Directly on Server

Karan Gupta



HOME_DIRECTORY

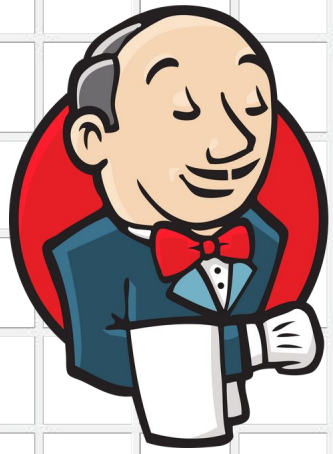
/var/lib/jenkins

Folder Structure

Karan Gupta

- **config.xml:** global configurations
- **credentials:** Holds sensitive information
- **plugins:** Downloaded plugin files
- **jobs:** Contains subfolders for each defined job:
  - **config.xml:** Specific configuration of the job
  - **builds:** Stores the build history
  - **workspace**
- **nodes:** Configuration files for nodes

Karan Gupta

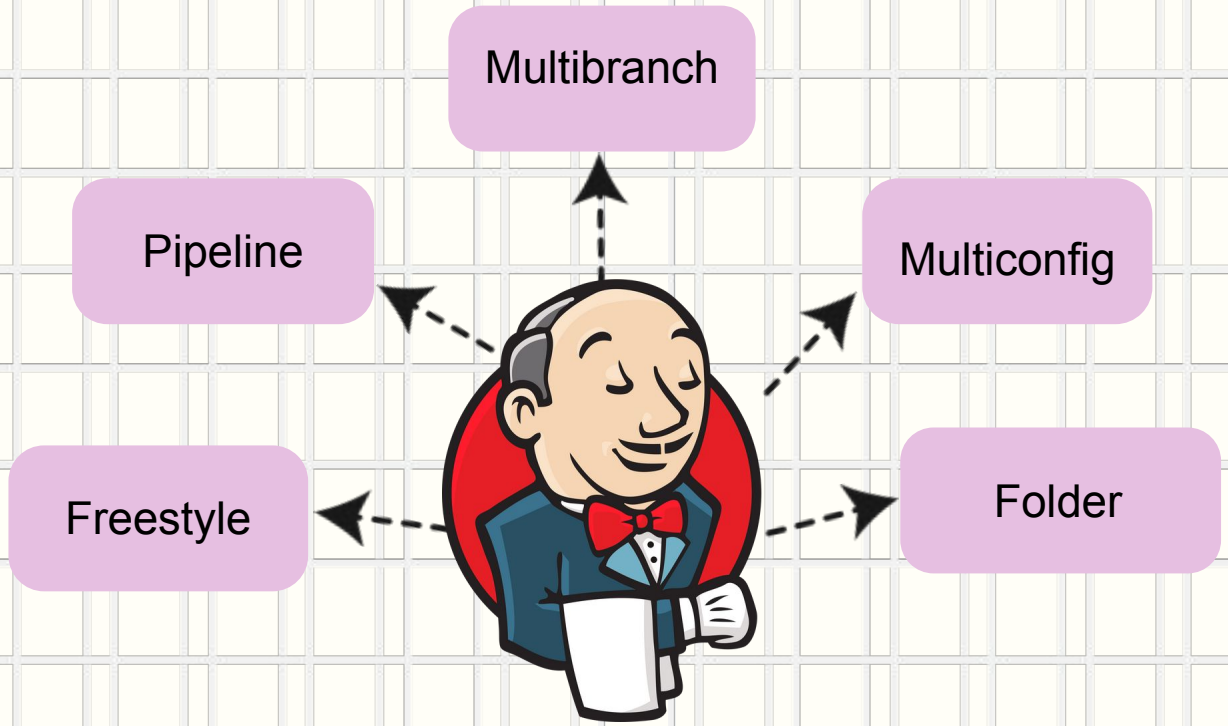


Jenkins

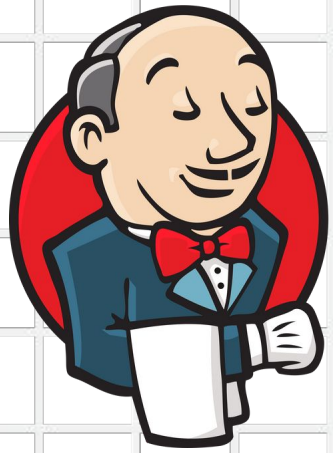
Credentials

JOBS

Karan Gupta

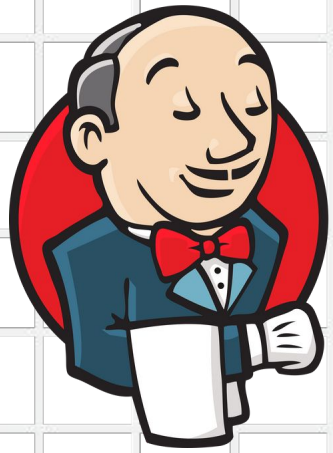


Karan Gupta



Freestyle Job

Karan Gupta



GIT Refresher

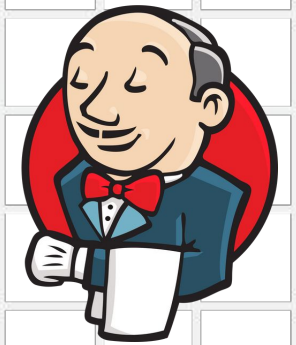
GIT

Karan Gupta

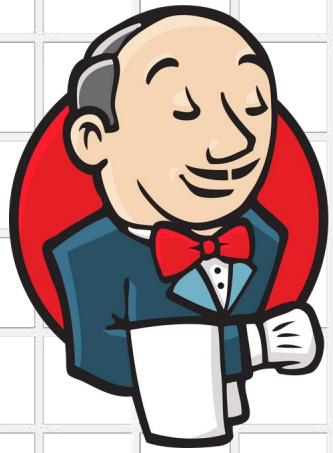
- Git is a **distributed version control system (DVCS)** used primarily for **tracking changes in code** and coordinating collaboration among developers.

Commands

- git init
- git clone
- git add
- git commit -m “message”
- git push



Karan Gupta



Maven Refresher

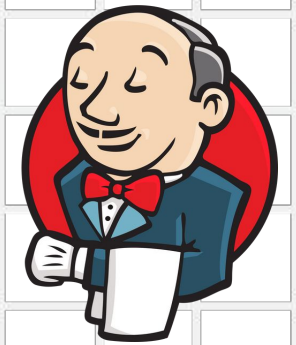
Maven Tool

Karan Gupta

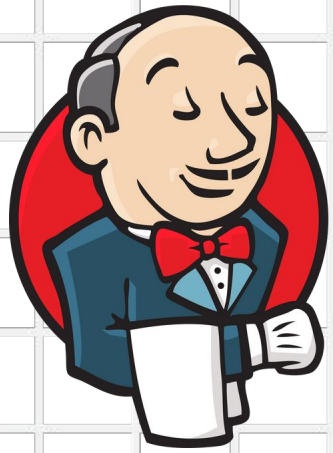
- **Maven** is a free and open-source project management tool specifically designed for **Java-based projects**.

Commands

- mvn clean
- mvn compile
- mvn test
- mvn package



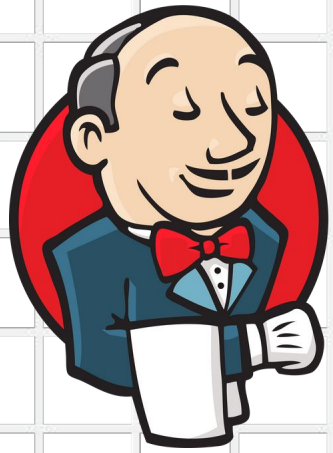
Karan Gupta



Freestyle

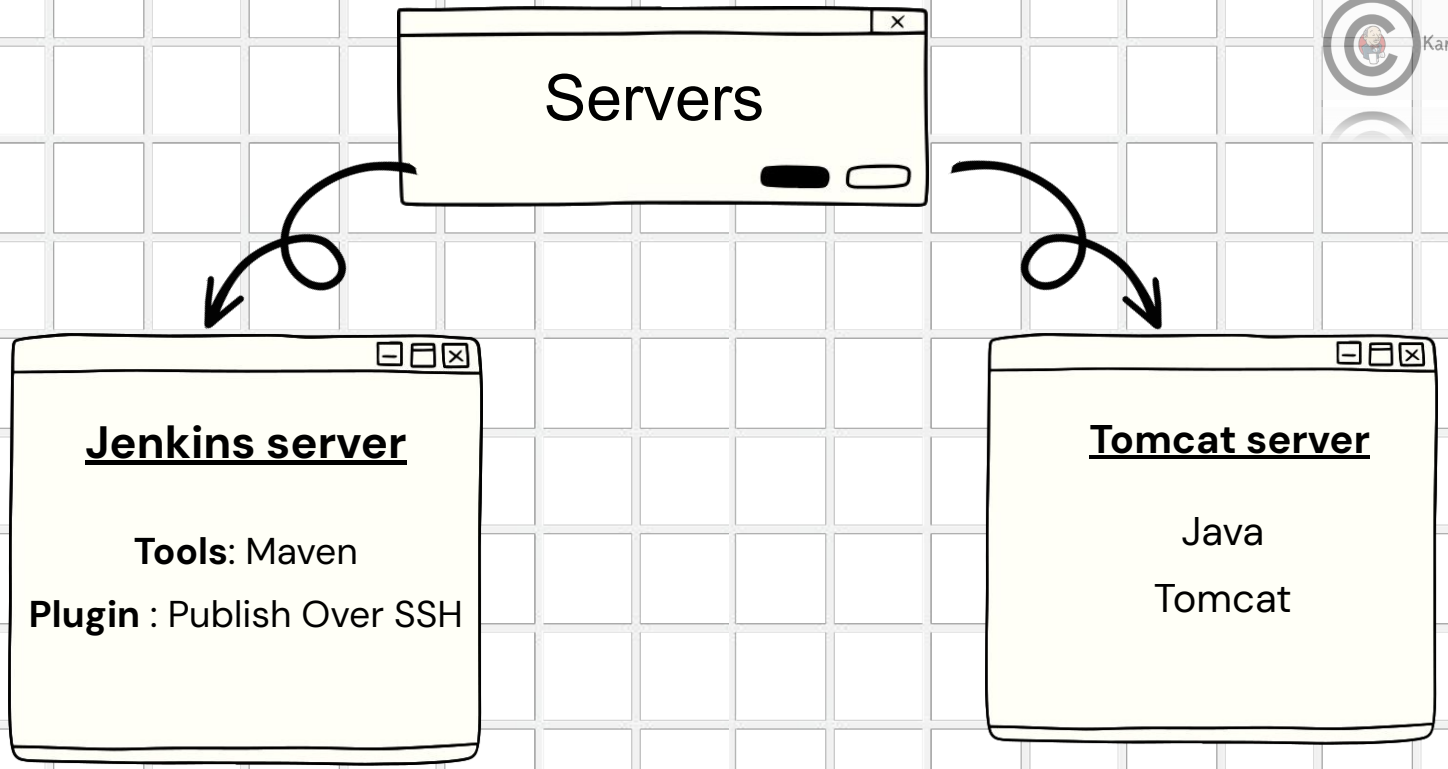
Using Maven

Karan Gupta



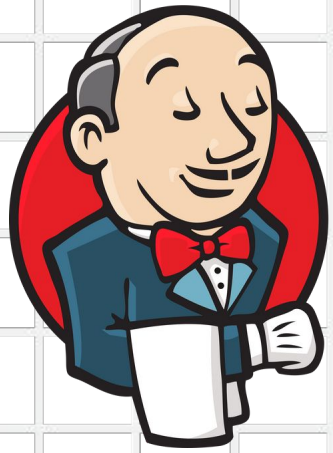
Freestyle

Project


Servers
Jenkins server
Tools: Maven
Plugin : Publish Over SSH
Tomcat server
Java
Tomcat

Karan Gupta

Karan Gupta



Github Webhooks

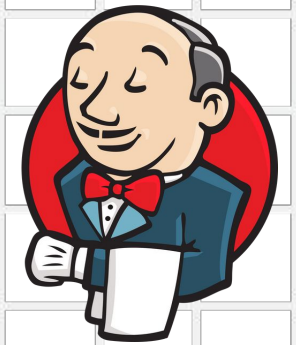
GitHub Webhook

Karan Gupta

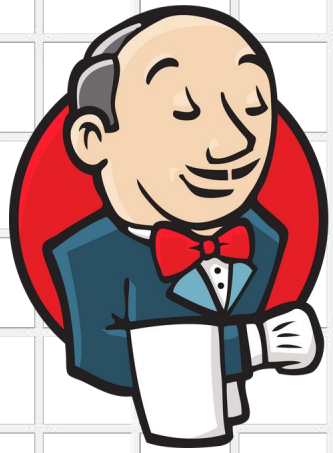
- Event-driven notification system from GitHub that automatically triggers actions in your Jenkins server when specific events occur in your repository.

Benefits

- Reduced Latency
- Automated Triggering
- Traceability and Visibility
- Secured



Karan Gupta



Pipeline

Job

Jenkinsfile

Karan Gupta

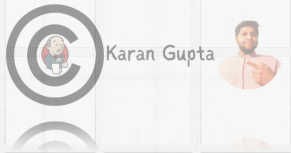
```
node {
    stage('Checkout') {
        echo 'this is checkout stage'
    }
    stage('Build') {
        echo 'this is build stage'
    }
}
```

Imperative

```
pipeline{
    agent any
    stages{
        stage('build stage'){
            steps{
                echo "this is build stage"
            }
        }
        stage('deploy stage'){
            steps{
                echo "this is deploy stage"
            }
        }
    }
}
```

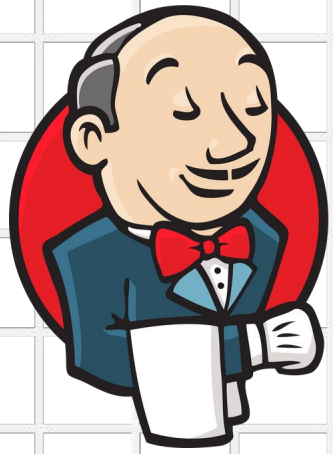
Declarative

Declarative Jenkinsfile




```
pipeline{
    agent any
    stages{
        stage('build stage'){
            steps{
                echo "this is build stage"
            }
        }
        stage('deploy stage'){
            steps{
                echo "this is deploy stage"
            }
        }
    }
}
```

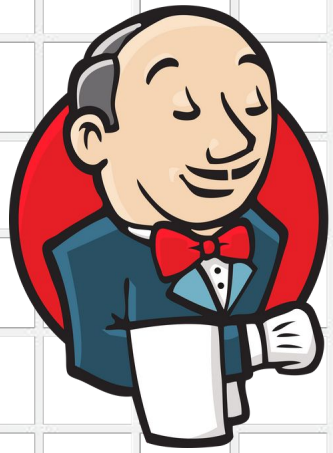
Karan Gupta



Jenkinsfile

1 - Basic

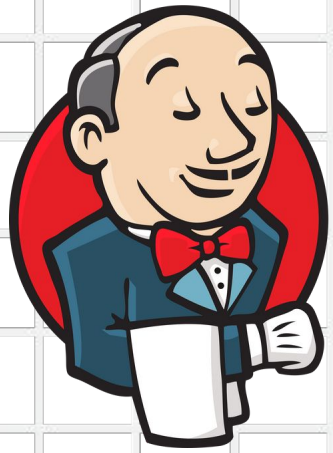
Karan Gupta



Jenkinsfile

2 - Using Options

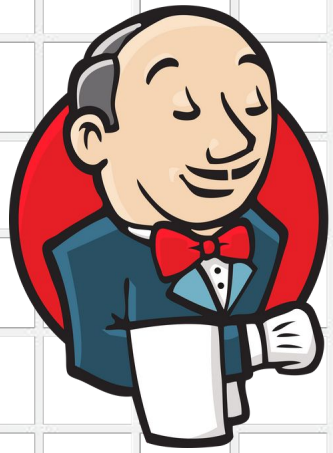
Karan Gupta



Jenkinsfile

3- Checkout Stage

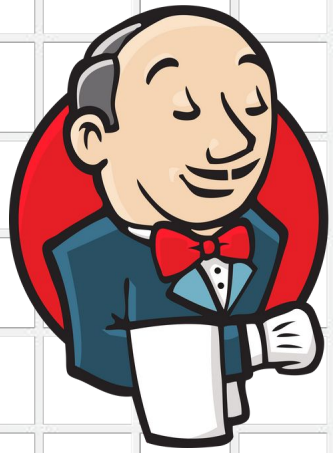
Karan Gupta



Jenkinsfile

4 - Using Tools

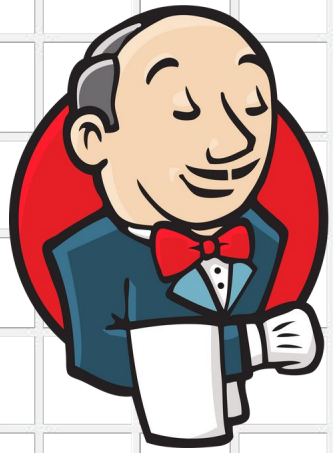
Karan Gupta



Jenkinsfile

5 - Env. Variables

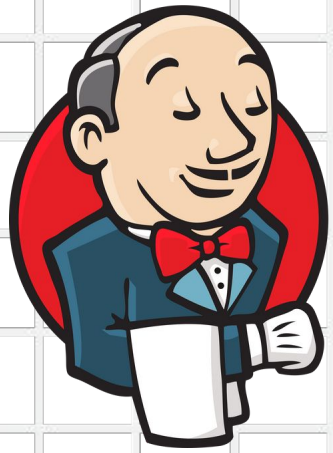
Karan Gupta



Jenkinsfile

6 - Parameterized

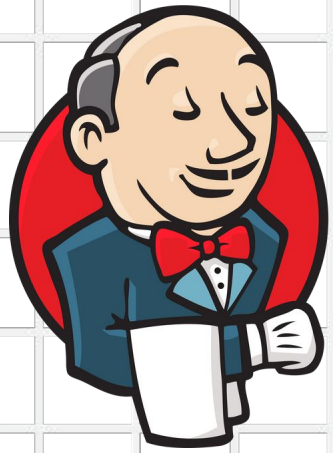
Karan Gupta



Jenkinsfile

7 - Upstream & Downstream Jobs

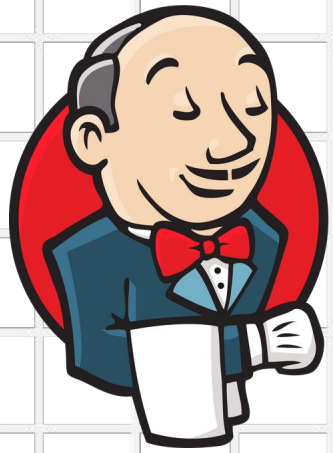
Karan Gupta



Jenkinsfile

8 - Post Actions

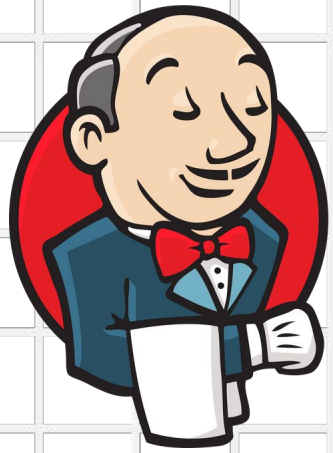
Karan Gupta



Jenkinsfile

9 - Built-In Variables

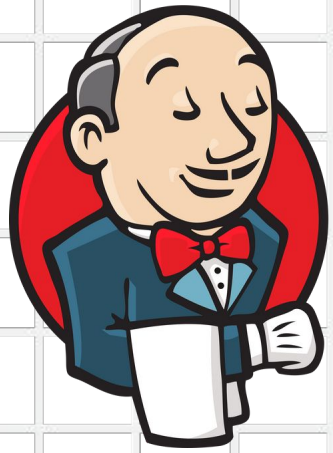
Karan Gupta



Jenkinsfile

10 - Using Script

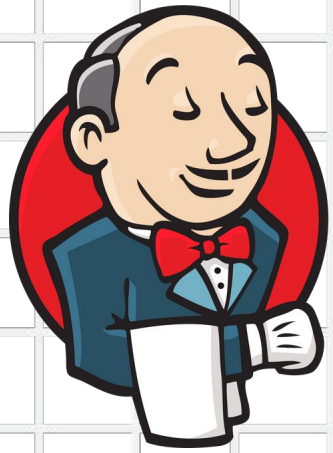
Karan Gupta



Jenkinsfile

11 - Script from file

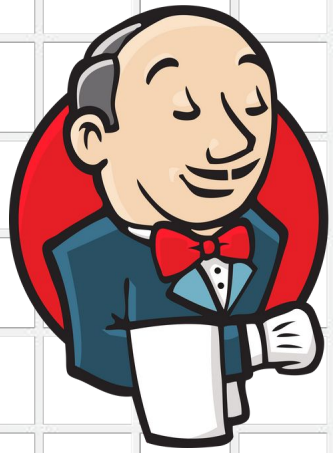
Karan Gupta



Jenkinsfile

12 - Applying Loops

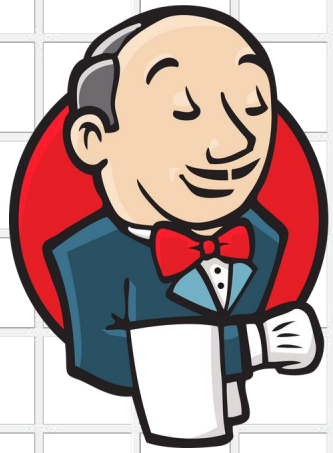
Karan Gupta



Jenkinsfile

13 - Loops + Script

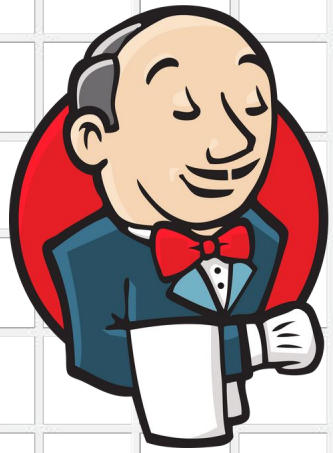
Karan Gupta



Jenkinsfile

14 - Conditions

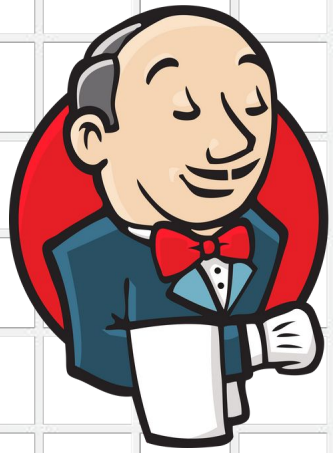
Karan Gupta



Jenkinsfile

15- Functions

Karan Gupta



Docker Refresher

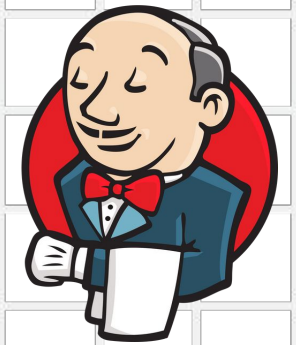
Docker

Karan Gupta

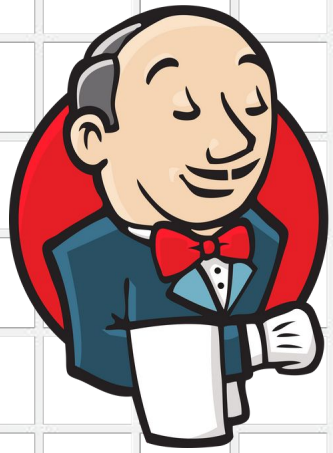
- Docker is an open-source platform for **developing, shipping, and running applications** in a standardized way using **containers**.

Commands

- docker ps
- docker run [image] [command]
- docker stop [container id/name]
- docker rm [container id /name]
- docker images
- docker rmi [image id/name]



Karan Gupta



Docker

Server Setup

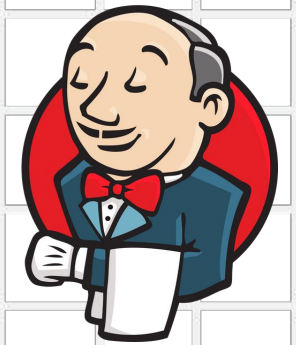
Docker Hub

Karan Gupta

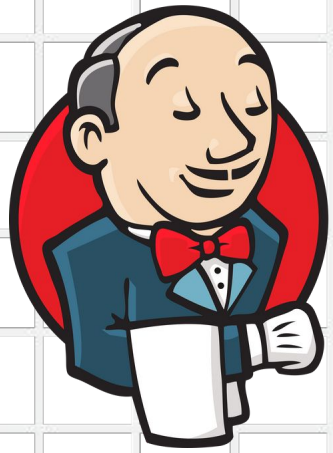
- **Docker Hub** is a **cloud-based registry service** owned by Docker, Inc. Docker Hub serves as a **central repository** for storing and sharing **Docker images**.

Commands

- docker login
- docker pull [image]
- docker push [image]
- docker search [term]



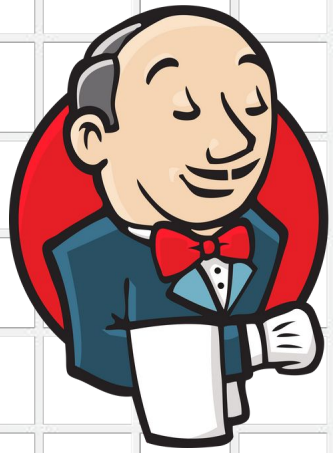
Karan Gupta



Jenkinsfile

16 - Docker

Karan Gupta



Jenkinsfile

17 - Docker + Dockerhub

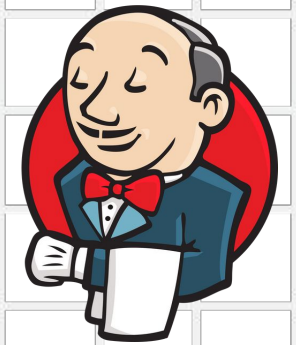
Karan Gupta

PROJECT

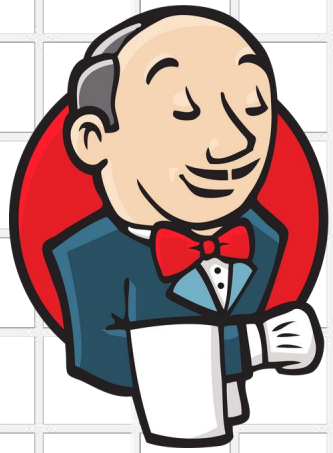
Docker Deployment

Project
Docker Deployment

Karan Gupta



Karan Gupta

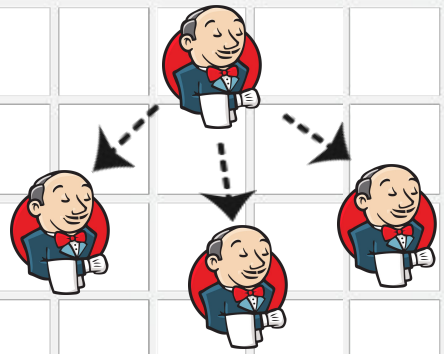


Agents

Agents Benefits

Scalability

Flexibility

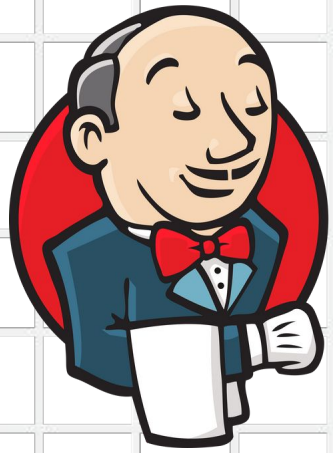


Maintenance

Security

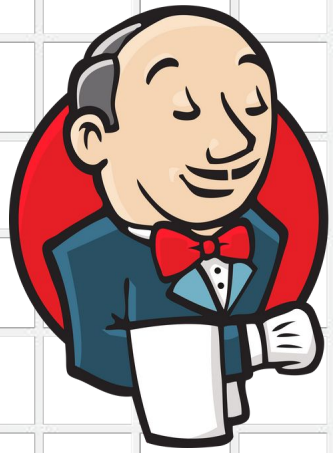
Karan Gupta

Karan Gupta



Adding Agent

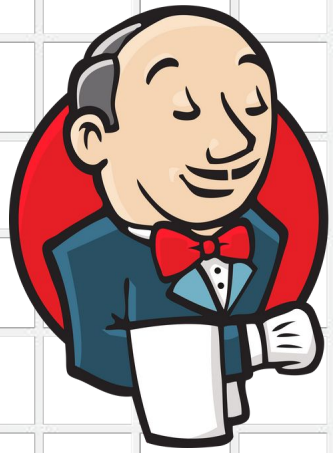
Karan Gupta



Jenkinsfile

18 - Agent Usage

Karan Gupta



Post Action

Email Notification

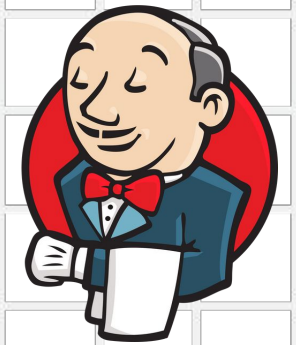
Email Notification

Karan Gupta

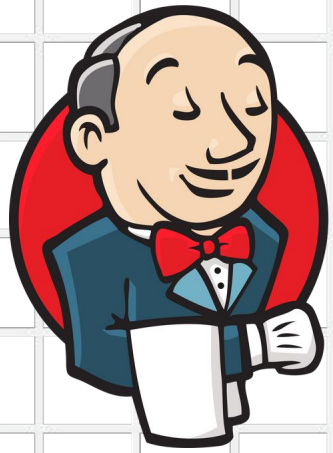
- E-mail notification allows you to receive automatic emails about various events and activities within your builds and pipelines.

Gmail Sample

- **Smtp server** - smtp.gmail.com
- **Suffix** - @gmail.com
- **Port**: 465
- Username and App Passwords
- SSL selected
- Recipient address



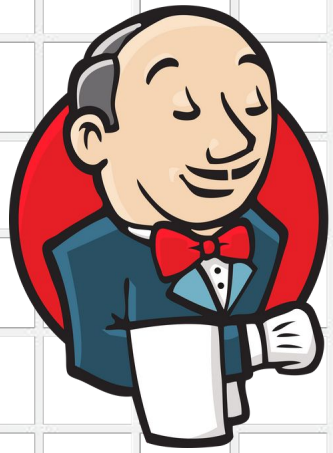
Karan Gupta



Jenkinsfile

19 - Email

Karan Gupta



Multi-Branch Job

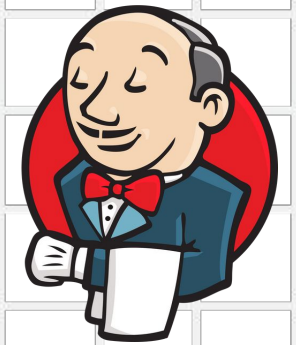
Multi Branch Job

Karan Gupta

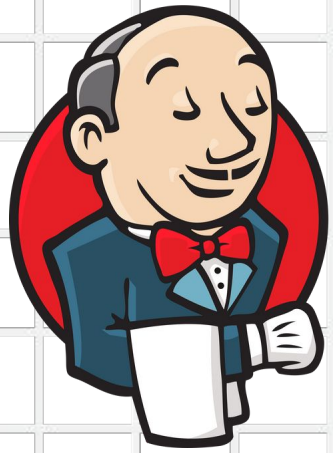
- A multi-branch job in Jenkins is a powerful feature that allows you to manage and automate builds for multiple branches or repositories from a single configuration.

Benefits

- Reduce duplication
- Centralized management
- Branch-specific configurations
- Individual build histories



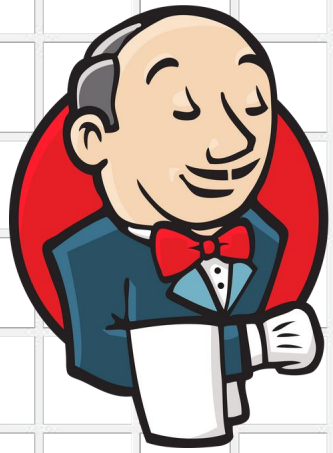
Karan Gupta



Jenkinsfile

20 - Multibranch

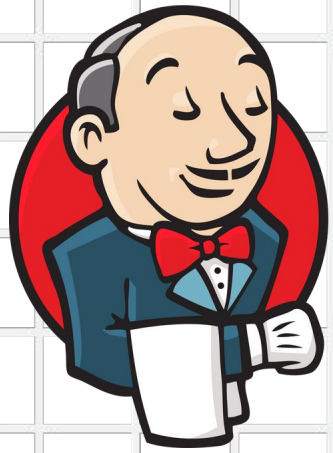
Karan Gupta



Jenkinsfile

21 - AWS

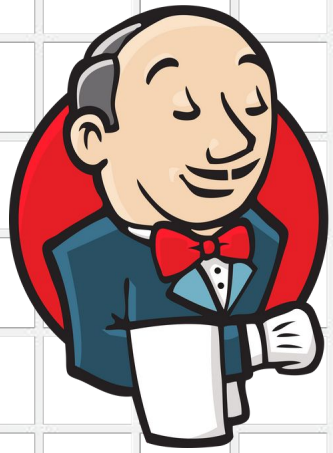
Karan Gupta



Jenkinsfile

21 (a) - AWS

Karan Gupta



Jenkins

AWS

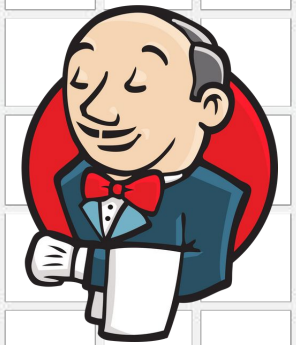
AWS with Jenkins

Karan Gupta

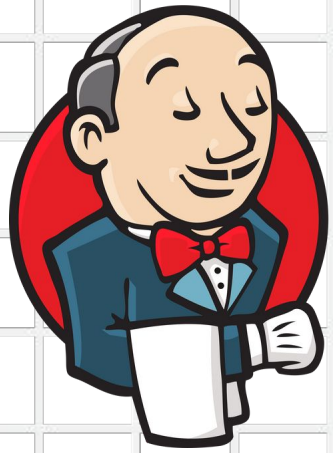
- Integrating **Jenkins** with **AWS** offers several advantages for building and deploying applications on the AWS Cloud.

Few Industry tasks

- Artifact management
- Creating custom AMI
- Taking snapshots
- Codepipeline
- Creating Infrastructure



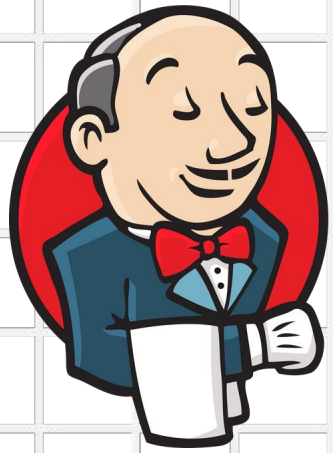
Karan Gupta



Jenkinsfile

21 (b) - AWS

Karan Gupta



Kubernetes Refresher

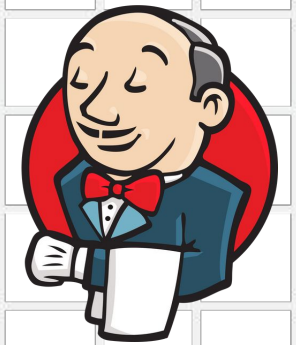
Kubernetes

Karan Gupta

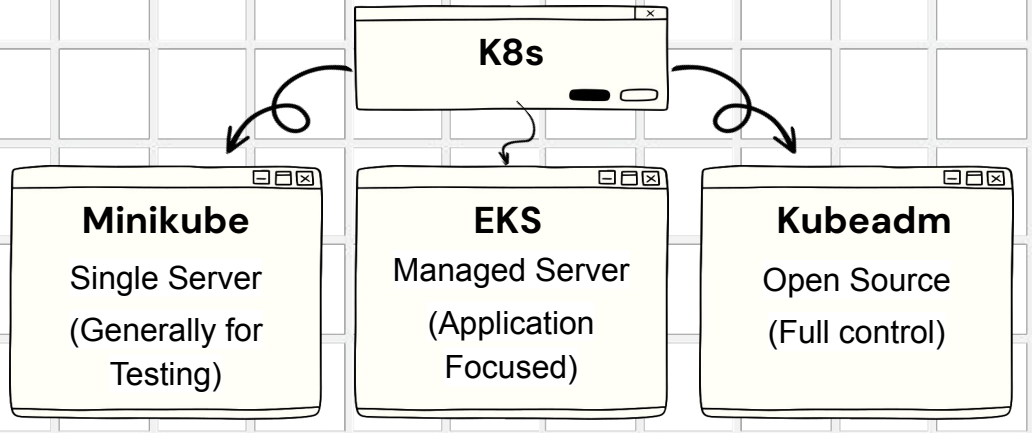
- Kubernetes (often shortened to K8s) is an open-source platform designed for automating the deployment, scaling, and management of containerized applications.

Benefits

- Automated Container Orchestration
- Self-healing Capabilities
- High Availability
- Faster Time to Market
- Portability

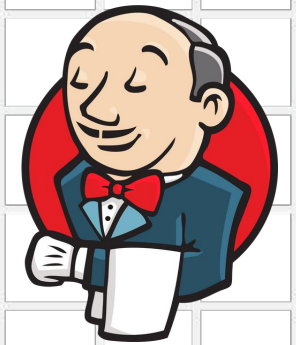


Different Flavors



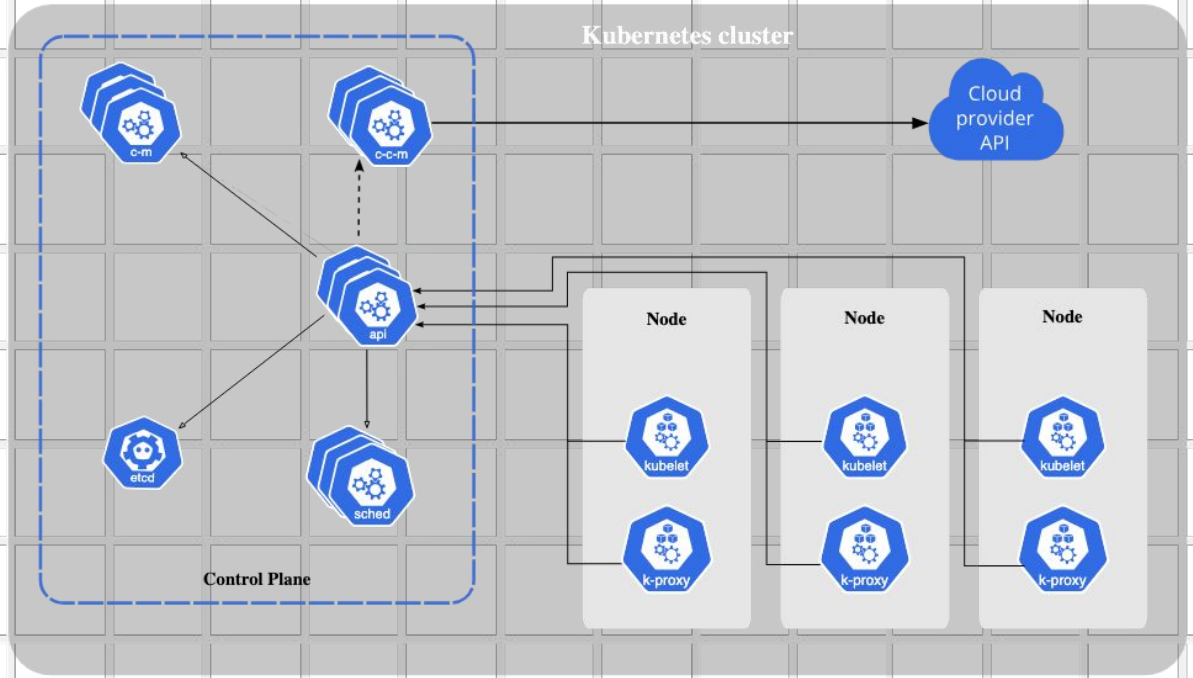
Commands

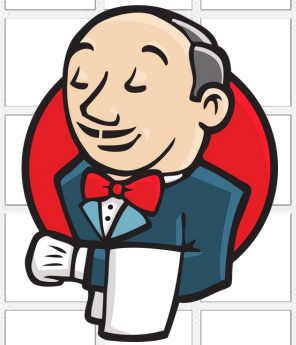
- kubectl apply -f <filename>
- kubectl get <resource> <resource name>
- kubectl describe <resource> <resource name>
- Kubectl delete <resource> <resource name>



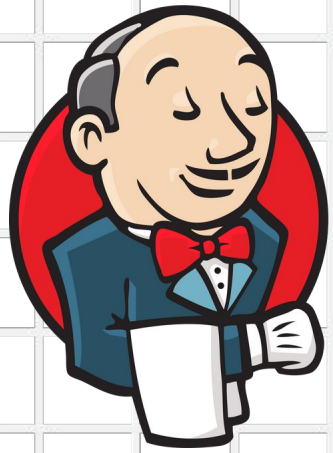
Architecture

Karan Gupta





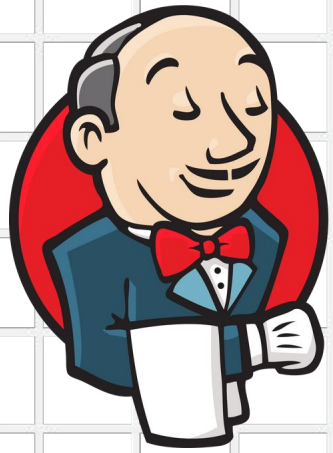
Karan Gupta



Docker

Server Setup

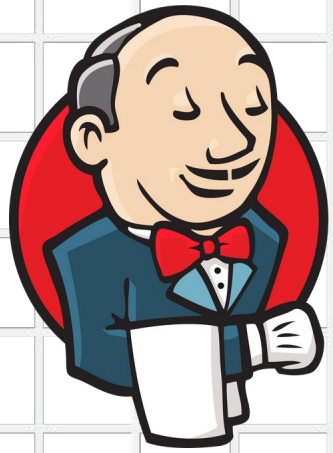
Karan Gupta



Jenkinsfile

22 - Kubernetes

Karan Gupta



Shared Library

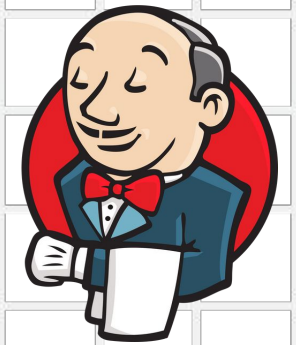
Shared Library

Karan Gupta

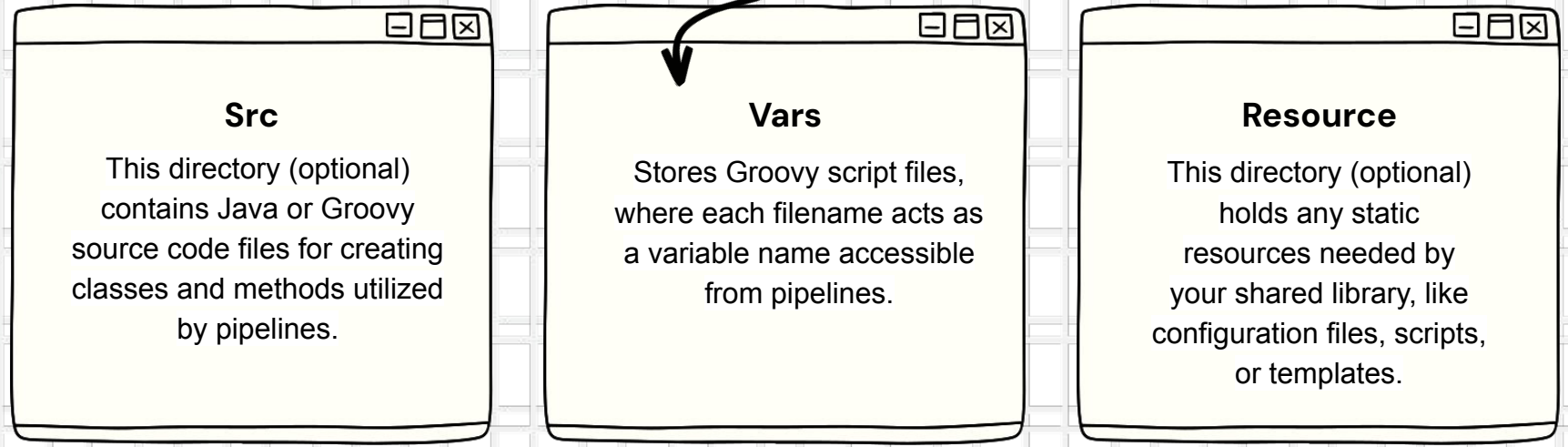
- A shared library in Jenkins is a reusable collection of Groovy scripts that can be used by multiple Jenkins jobs and pipelines. It allows you to modularize your code, improve consistency, and simplify pipeline development.

Benefits

- Reduce duplication
- Code Reuse
- Consistency and Maintainability
- Collaborations



Folder Structure



Src

This directory (optional) contains Java or Groovy source code files for creating classes and methods utilized by pipelines.

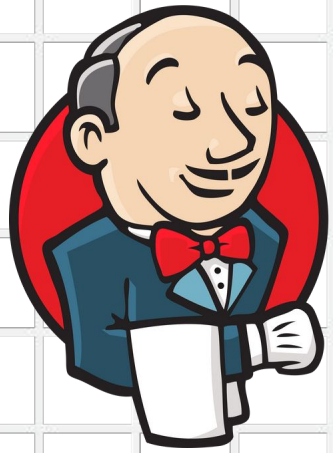
Vars

Stores Groovy script files, where each filename acts as a variable name accessible from pipelines.

Resource

This directory (optional) holds any static resources needed by your shared library, like configuration files, scripts, or templates.

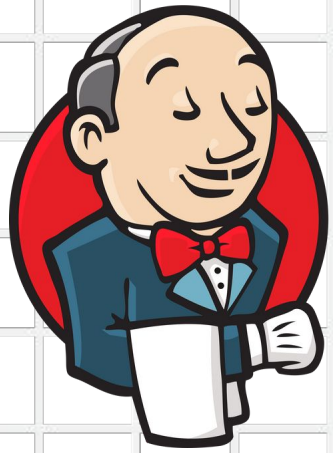
Karan Gupta



Jenkinsfile

23 - Shared Library

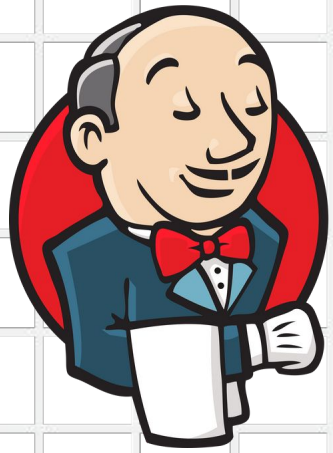
Karan Gupta



Jenkinsfile

23 - Shared Library (a)

Karan Gupta



SonarQube

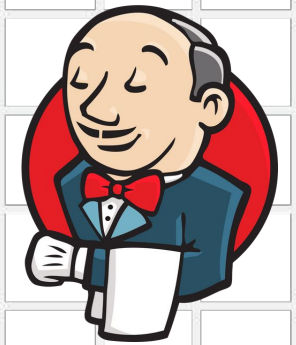
SonarQube

Karan Gupta

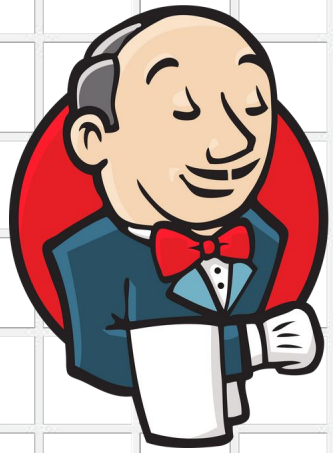
- SonarQube is an open-source platform that provides continuous code quality and security analysis for various programming languages.

Benefits

- Improved Code Quality
- Enhanced Security
- Early Defect Detection
- Continuous Monitoring



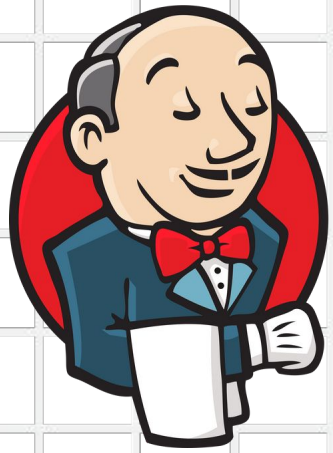
Karan Gupta



SonarQube

Server Setup

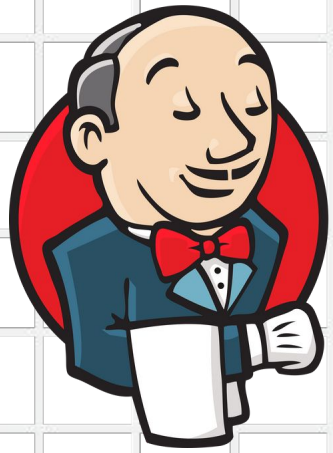
Karan Gupta



Jenkinsfile

24 - Sonarqube

Karan Gupta



Nexus Repository

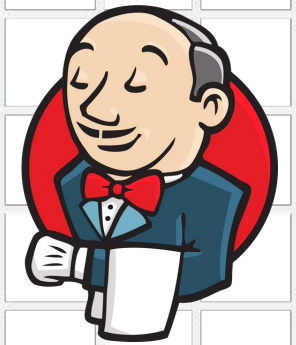
Nexus Repository

Karan Gupta

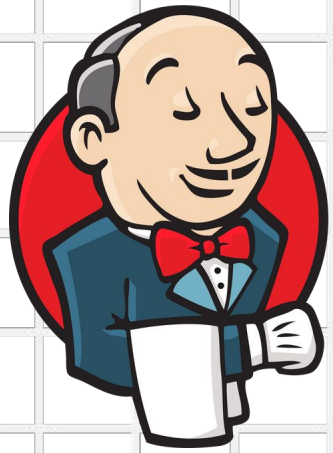
- Nexus acts as a central hub for storing, managing, and distributing various software artifacts like build outputs, libraries, packages, and container images.

Benefits

- Artifact Management
- Improved Build Efficiency
- Security and Access Control
- Collaborations



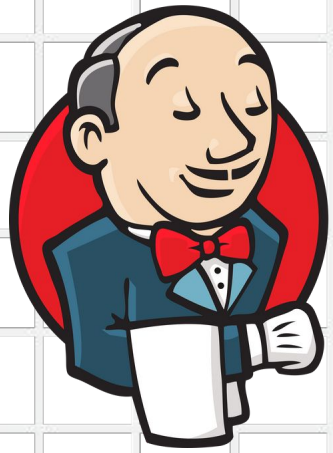
Karan Gupta



Nexus

Server SetUp

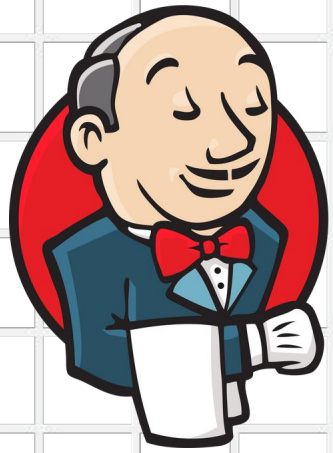
Karan Gupta



Jenkins

Port Change

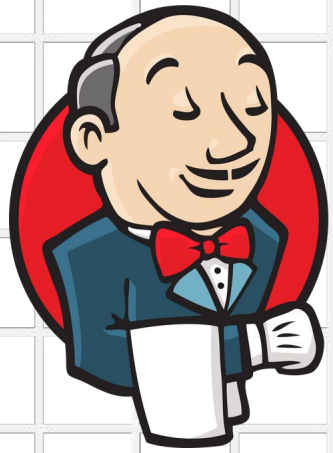
Karan Gupta



Nexus

Freestyle

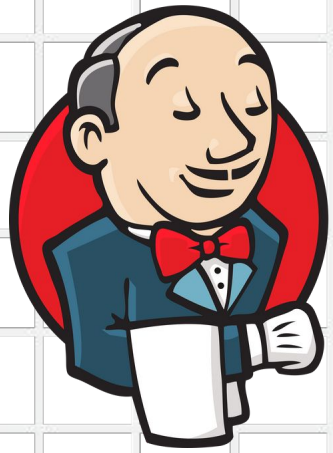
Karan Gupta



Jenkinsfile

25 - Nexus Repo

Karan Gupta



Terraform Refresher

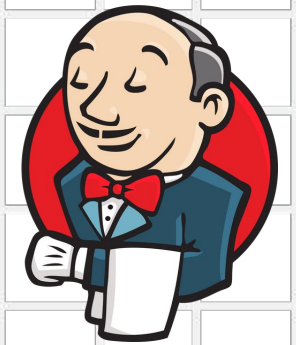
Terraform

Karan Gupta

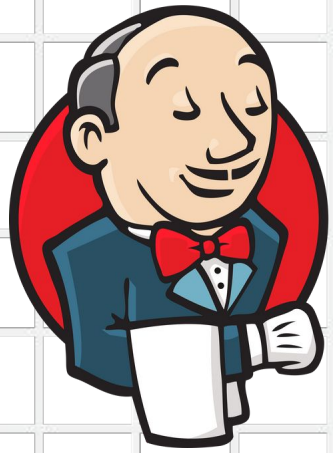
- **Terraform** is an open-source tool developed by HashiCorp that enables you to manage infrastructure as code (IaC). It allows you to define and provision infrastructure resources like servers, networks, databases, and other components **using human-readable configuration files**.

Commands

- terraform init
- terraform plan
- terraform apply -auto-approve
- terraform destroy -auto-approve



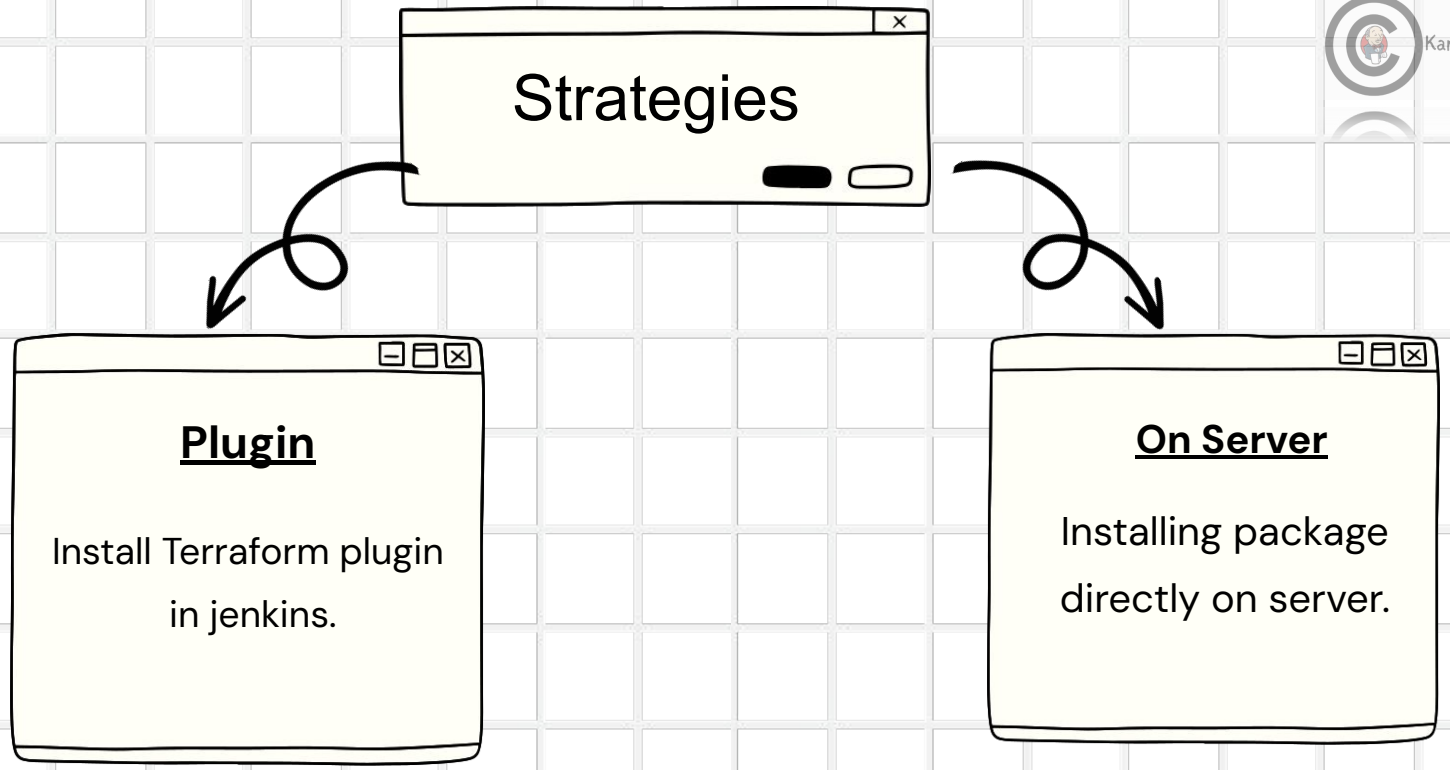
Karan Gupta



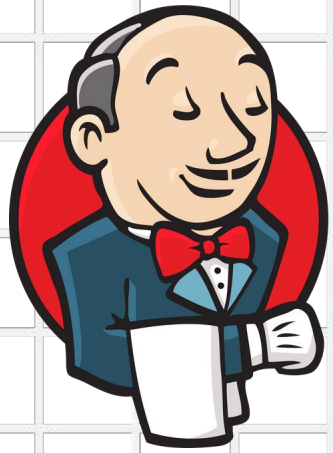
Terraform

SetUp

Karan Gupta


Strategies
Plugin
Install Terraform plugin in jenkins.
On Server
Installing package directly on server.

Karan Gupta



Jenkinsfile

26 - Terraform

Karan Gupta

Jenkinsfile

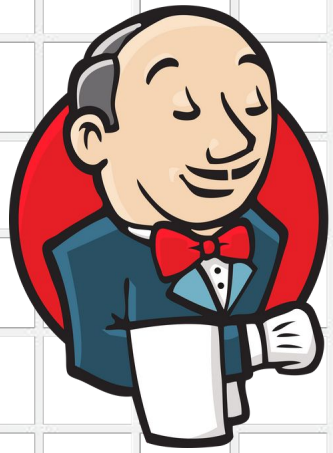
26 - Terraform (a)

Karan Gupta

Jenkinsfile

26 - Terraform (b)

Karan Gupta



Ansible Refresher

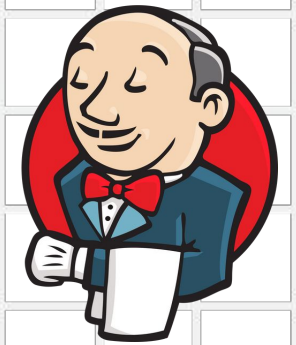
Ansible

Karan Gupta

- Ansible is an open-source automation tool that automates IT tasks such as configuration management, application deployment, cloud provisioning, and orchestration.

Benefits

- Automated Deployment
- Integration
- Consistency
- Scalability



Karan Gupta

Strategies

Plugin

Install Ansible plugin in jenkins.

On Server

Installing package directly on server.

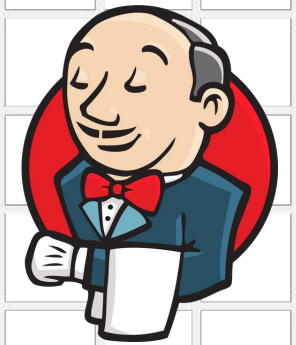
Ad-Hoc Commands

Karan Gupta

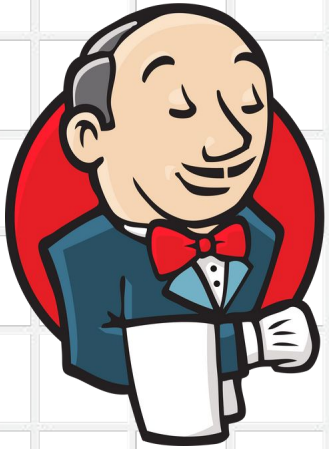
- **Ping**
  - ansible all -i <inventory_file> -m ping
- **Disk Space Check**
  - ansible all -i <inventory_file> -m shell -a "df -h"

Playbook Working

- Write Playbook
- Create Inventory File
- ansible-playbook -i <inventory_file> <playbook_file.yml>
- Review Results



Karan Gupta



Jenkinsfile

27 - Ansible

Karan Gupta

Jenkinsfile

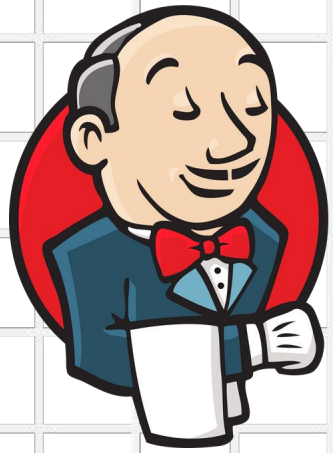
27 - Ansible (a)

Karan Gupta

Jenkinsfile

27 - Ansible (b)

Karan Gupta



Add Users

Permissions

Karan Gupta

Strategies

Project Based

Access is given inside the job configuration.

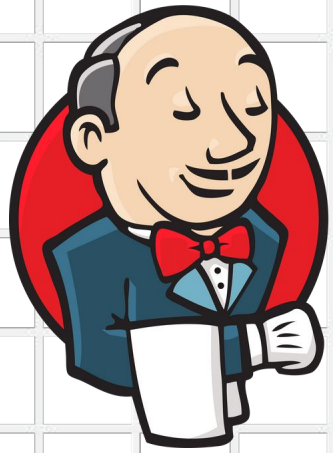
Logged-In can do Anything

Role Based

Access is based on roles and groups.

Anyone can do Anything

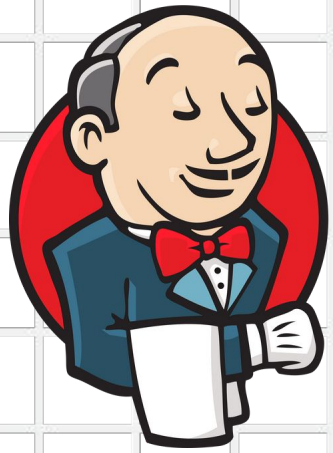
Karan Gupta



Strategy -1

Project Based

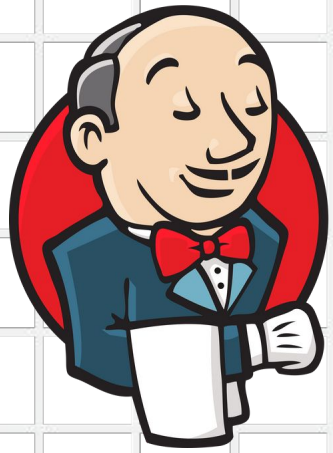
Karan Gupta



Strategy -2

Role Based

Karan Gupta



Jenkins Backup

Karan Gupta

Strategies

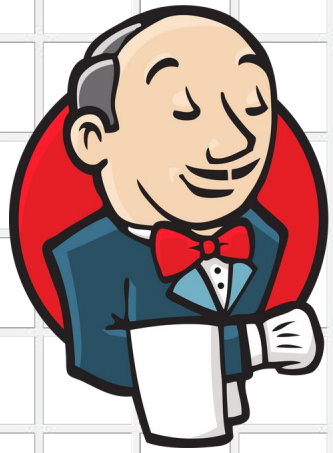
Cron Job

Create a cron job to take backup of jenkins directory.

Plugin Based

ThinBackup plugin can be used for backup.

Karan Gupta



Jenkins Upgradation

Jenkins Update

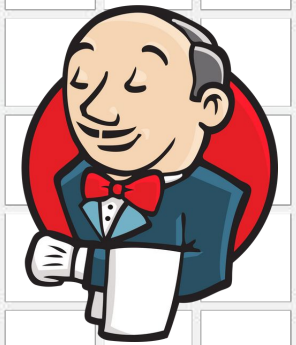
Karan Gupta

- A Jenkins upgrade refers to the process of updating your Jenkins installation to a newer version.

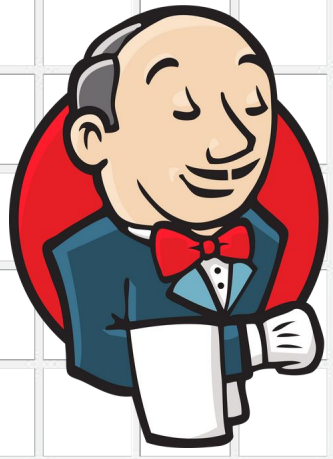
  The preferred way is the step case manner for the updation.

Process

- Backup
- Review Release Notes
- Check Plugin Compatibility:
- Prepare for Downtime
- Rollback Plan (For emergency)
- Test Environment and Monitor



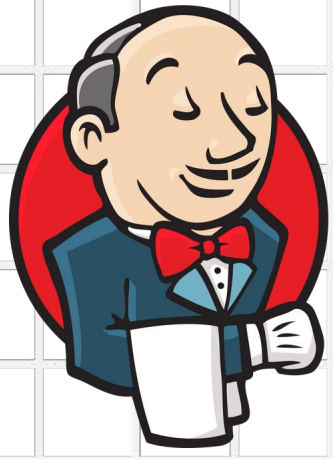
Karan Gupta



Jenkins

Script Console

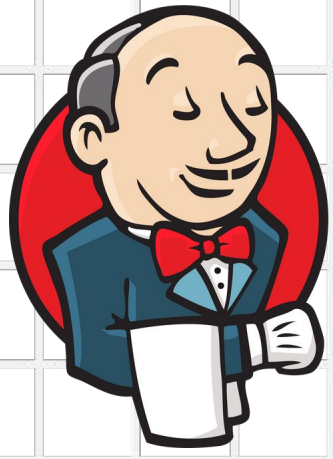
Karan Gupta



Jenkins

Troubleshooting Password Issue

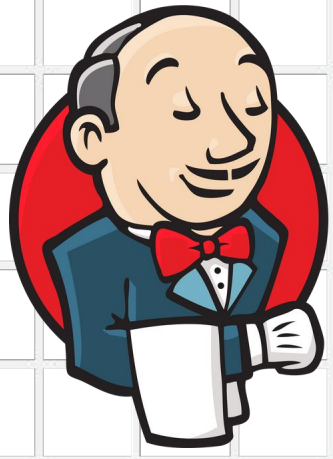
Karan Gupta



Jenkins

Timestamps

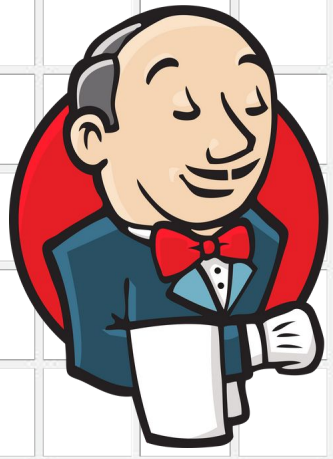
Karan Gupta



Jenkins

CLI

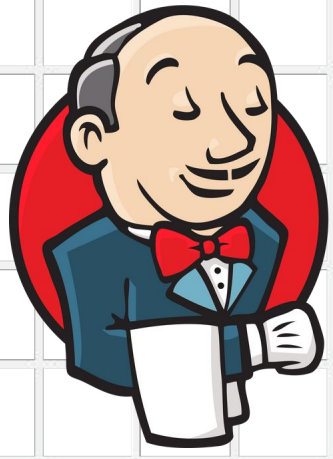
Karan Gupta



Jenkins

Disable, Enable and Delete Job

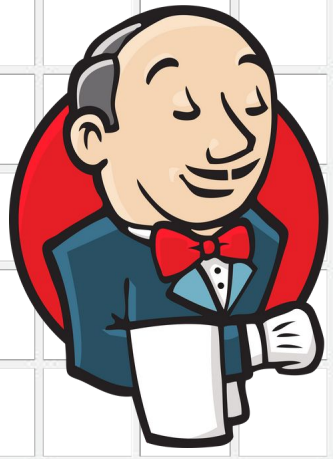
Karan Gupta



Jenkins

Restart from a Stage

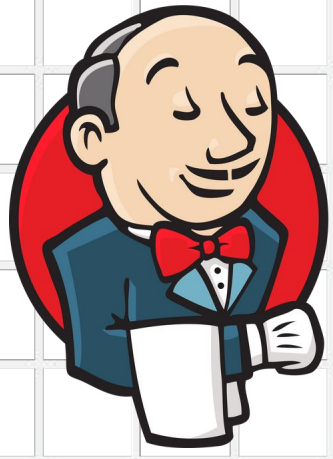
Karan Gupta



Jenkins

Logs

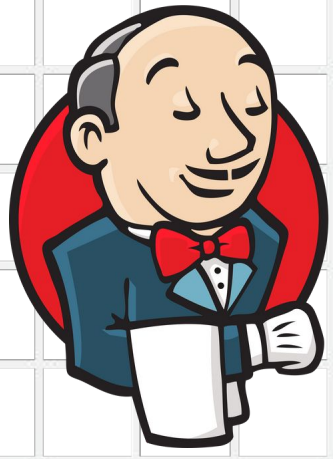
Karan Gupta



Jenkins

Blue Ocean

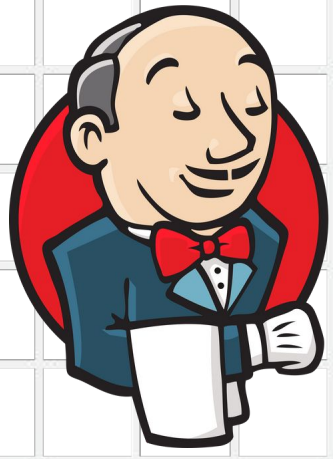
Karan Gupta



Jenkins

Folder

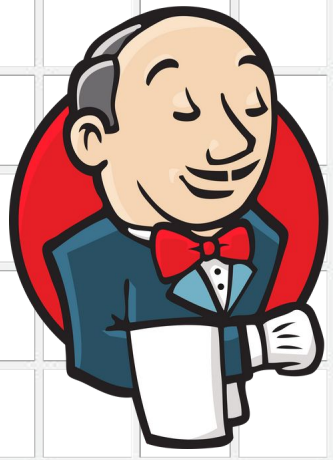
Karan Gupta



Jenkins

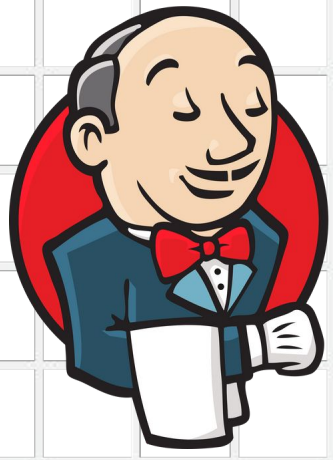
Timeout

Karan Gupta



Jenkins

Taking Input during Run-Time



Jenkins

Schedule Jobs

Scheduling a Job

In Jenkins, job scheduling refers to the functionality that allows you to automate the execution of build, test, and deployment workflows for your applications at specific times

Cron Table

*	*	*	*	*
Minute(0-59)	Hour(0-23)	Date(1-31)	Month(1-12)	Day(0-6)

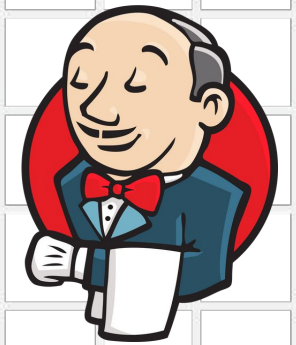
Karan Gupta

PROJECT

K8s Deployment

Project
Kubernetes Deployment

Karan Gupta



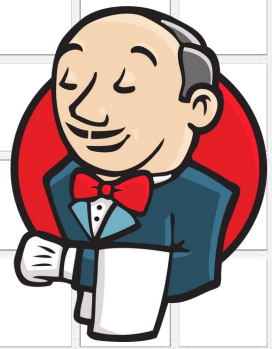
Karan Gupta

Industry PROJECT

Project Process:

Karan Gupta

- **Step 1: Customer Business Requirement**
  - The first step is acquiring requirement from business
- **Step 2: Short listing of the services to be utilised**
  - Listing all possible options with alternatives
  - Shortlisting them with the business alignment
- **Step 3: Creating a pseudo model or process**
  - Try to build the process of the working with the team
- **Step 4: Build the application**
  - Firstly try to build the application in dev
- **Step 5: Deploy it to the production**
  - After testing and approval finally deploy to production env.

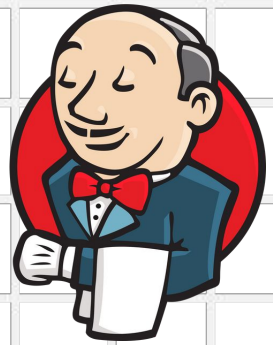


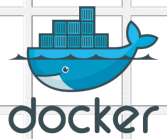
Major Project

Karan Gupta

Tools

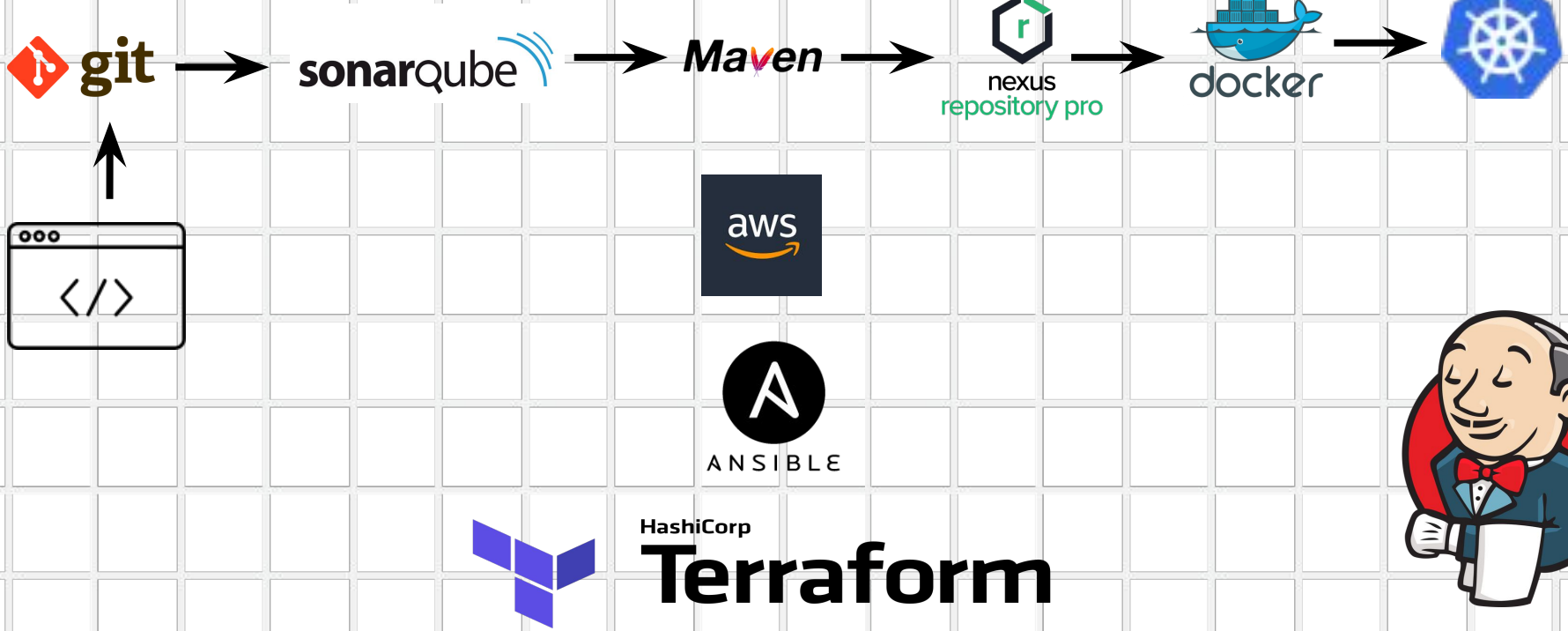
- Git + GitHub
- AWS
- Jenkins
- Maven
- Docker
- Sonarqube
- Nexus
- Kubernetes (EKS)
- Terraform
- Ansible





Architecture

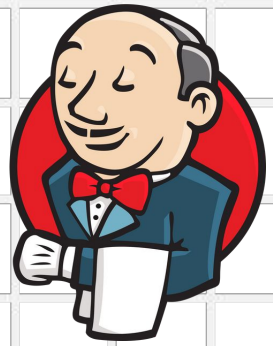
Karan Gupta

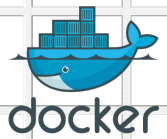


Tools Usage

- **Linux**
  - For script and commands
- **Git + GitHub**
  - As a SCM
- **AWS**
  - Cloud service provider
- **Sonarqube**
  - For the code analysis
- **Maven**
  - For building project
- **Nexus**
  - For storing artifacts

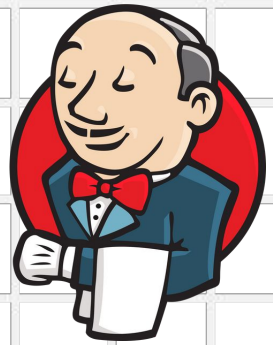
Karan Gupta

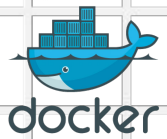




Tools Usage

- **Docker**
  - For creating images
- **Docker Hub**
  - For storing Images
- **Kubernetes (EKS)**
  - For Deployment purpose
- **Terraform**
  - For creating servers infra
- **Ansible**
  - For installing softwares





Karan Gupta

Set-Up

Phase-1

Infrastructure setup.

In this phase we try to lay foundation of the project.

Phase-2

Dependency setup.

In this phase, we add dependencies of the project like credentials, tools.

Phase-3

Deployment phase.

Final build of project with all the jenkins stages.

Karan Gupta

Phase-1

Infrastructure

Phase-1 - Infrastructure Setup

Karan Gupta

1. Setting Jenkins Server

A. Create script to install jenkins on a server.

B. Either using plugins or installing package directly on server install terraform, jenkins, docker K8s and git on same server.

C. After setting up the jenkins server, add maven tool.

D. Configure AWS user too

Phase-1

Karan Gupta

2. Git Setup - Public/Private

A. Create a repository for placing all codes in the repository.

B. For public repo:

   a. No credentials required

C. For private repo

   a. Credentials required

Phase-1

Karan Gupta

3. Kubernetes Setup - K8s

A. Using EKS or Kubeadm setup the K8s cluster.

   a. Setup worker nodes

   b. Have details of users/roles

   c. Information about kubeconfig file

Phase-1

Karan Gupta

4. Terraform Scripts

A. Create terraform modules for having 3 servers for the working using jenkins job.

   a. Sonar Server

   b. Nexus Server

   c. Test Server

B. Place the terraform script in the github for the working.

Phase-1

Karan Gupta

5. Nexus and Sonar Server Setup

A. Download and Install the Nexus server

   a. Verify its working at 8081

B. Download and Install the Sonar server

   a. Verify its working at 9000

Phase-1

Karan Gupta

6. Test Server Setup

A. Using ansible playbook script from jenkins job install the following:

a. Docker

b. Git

c. Maven

Phase-1

Karan Gupta

7. Plugins and Credentials

A. Plugins
   a. Install Sonarqube Plugin
   b. Install AWS credentials plugin

B. Credentials
   a. AWS keys
   b. Private Key for Ansible
   c. Nexus Username and Password
   d. Docker Hub Creds (Optional in case of ECR)

Karan Gupta

Phase-2

Dependency

Phase-2 - Repository, Project Setup

Karan Gupta

1. Docker Repository

A. Create Repository for docker images

   a. ECR

   b. Docker Hub

Phase-2

Karan Gupta

2. Sonarqube Project Setup

A. Create a sonarqube Project

B. Then API token from security

C. Update setting of sonarqube server in system and scanner in tools

D. Add token in credentials

Karan Gupta

Phase-3

Deployment

Phase-3 - Jenkins Job

Karan Gupta

1. Options

A. Build Discarder

B. Timeout

C. Clean Workspace

2. Tools

A. Maven

Phase-3 - Jenkins Job

Karan Gupta

3. Environment

A. Custom Variables

B. Credentials

4. Agents

A. Based on requirement

Phase-3 - Jenkins Job

Karan Gupta

5. Stages

A. Checkout

B. Sonar test

C. Maven Build

D. Nexus Artifacts

E. Docker Build

F. Docker Hub Push/ ECR push

G. K8s Deployment

H. Post actions

Project Outcome

Karan Gupta

Things achieved:

A. Understanding of Continuous Integration and Continuous Deployment (CI/CD)

B. Proficiency in Jenkins Installation and Configuration

C. Creating and Managing Jenkins Jobs

D. Working with Jenkins Pipelines

E. Integration with Version Control Systems

F. Artifact Management and Distribution:

G. Testing and Quality Assurance

H. Best Practices and Troubleshooting