PW SKILLS | DevOps and Cloud Computing

Getting Started with Jenkins

Objective

- Understand the architecture and components of Jenkins.
- Install and configure Jenkins for CI/CD automation.
- Integrate Jenkins with GitHub for automated builds.
- Set up and manage build triggers and scheduling.

PW SKILLS

Explain Jenkins as a CI/CD automation tool

Jenkins

Jenkins is an open-source automation server that plays a crucial role in implementing Continuous Integration/Continuous Delivery and Deployment (CI/CD) pipelines.

- It is primarily used to automate the software development lifecycle, ensuring that code changes are integrated, tested, and deployed efficiently.

Key Features of Jenkins:

1. CI/CD Pipelines
2. Automation and Efficiency
3. Plugin Ecosystem
4. Flexibility and Scalability

Jenkins

How Jenkins Works:

1. Triggering Builds
2. Automated Testing
3. Deployment
4. Monitoring and Feedback

Benefits of Using Jenkins:

- Improved Quality
- Faster Deployment
- Efficient Collaboration

Pop Quiz

Q. How can Jenkins trigger builds automatically?

A

Only through scheduled builds

B

Through polling SCM, webhooks, or scheduled builds

Pop Quiz

Q. How can Jenkins trigger builds automatically?

A

Only through scheduled builds

B

Through polling SCM, webhooks, or scheduled builds

PW SKILLS

Discuss the core components

Components of Jenkins

Core Components of Jenkins:

1. **Jenkins Master (Controller)**

- Role: The Jenkins Master acts as the central hub, controlling and managing all automation tasks. It schedules jobs, monitors their execution, and handles plugin integration.
- Functionality: It holds the central Jenkins configuration, manages agents, and coordinates project workflows.

Components of Jenkins

2. Jenkins Agents (Nodes)

- Role: Agents are responsible for executing build jobs. They connect to the Jenkins Master to run tasks such as building and testing code.
- Functionality: Agents can be installed on physical machines, virtual machines, or cloud instances, allowing for distributed builds and scalability.

3. Plugins

- Role: Plugins extend Jenkins' functionality by integrating various tools and platforms. They support features like testing frameworks, version control systems, and build tools.
- Functionality: Plugins can be easily installed and updated from the Jenkins dashboard, enhancing its capabilities beyond its core features.

PW SKILLS

Demonstrating installing Jenkins on a system

Let’s do it

1. **Install Prerequisites**
- Java
- Dependencies
2. **Download and Install Jenkins**

Windows:

- Download the Jenkins LTS version from the official Jenkins website.
- Run the .msi installer and follow the on-screen instructions.

Linux:

- Add the Jenkins repository:

```
 wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |
sudo apt-key add -
 sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/
> /etc/apt/sources.list.d/jenkins.list'
```

- **Update and install Jenkins:**

```
sudo apt-get update
sudo apt-get install openjdk-11-jdk
sudo apt-get install jenkins
```

Docker:

- **Pull the Jenkins image:**

```
docker pull jenkins/jenkins:lts
```

- **Run the Jenkins container:**

```
docker run -p 8080:8080 -v jenkins-data:/var/jenkins_home
jenkins/jenkins:lts
```

3. Configure Jenkins

- Open a web browser and navigate to ***http://localhost:8080.***

4. Verify Installation

- Create a new job to test the installation by selecting "New Item" and configuring a simple build task.

This setup ensures Jenkins is ready to automate CI/CD pipelines.

PW SKILLS

The Jenkins web interface and initial setup

Jenkins web interface

The Jenkins web interface is the primary way users interact with Jenkins. It provides a centralized dashboard for managing jobs, plugins, and configurations.

- Dashboard
- Job Configuration
- Plugin Management
- User Management

Initial Setup of Jenkins:

1. Accessing Jenkins
2. Unlocking Jenkins
3. Customizing Jenkins
    a. Install Plugins
    b. Create Admin User

Jenkins web interface

4. Configuring Jenkins
- Global Tool Configuration
- Node Management

5. Creating a Job
- Pipeline Creation
- Pipeline Script

Take A 5-Minute Break!

- Stretch and relax
- Hydrate
- Clear your mind
- Be back in 5 minutes

PW SKILLS

The benefits of integrating Jenkins with GitHub

Benefits

Integrating Jenkins with GitHub offers several benefits for automated builds, testing, and deployment:

1. Automated Builds
2. Efficient Testing
3. Streamlined Deployment
4. Bi-directional Integration
5. Enhanced Security and Authentication

PW SKILLS

Demonstrate linking Jenkins to GitHub

Let’s do it

Here's a concise guide on linking Jenkins to GitHub, focusing on configuring GitHub Webhooks and setting up credentials and authentication in Jenkins:

Step 1: Install Necessary Plugins in Jenkins:

- Git Plugin
- GitHub Plugin
- GitHub Authentication Plugin

Step 2: Configure GitHub Webhooks:

- Create a Jenkins Job
- Configure GitHub Hook Trigger
- Add Webhook in GitHub

Let’s do it

Step 3: Set Up Credentials and Authentication in Jenkins:

- Create Credentials
- Configure Authentication

Step 4: Test the Integration

Pop Quiz

Q. What is a primary benefit of integrating Jenkins with GitHub?

A

Automated builds and testing on code changes

B

Manual build triggers only

Pop Quiz

Q. What is a primary benefit of integrating Jenkins with GitHub?

A

Automated builds and testing on code changes

B

Manual build triggers only

PW SKILLS

Different build triggers in Jenkins

Build triggers

Here's a concise overview of different build triggers in Jenkins:

1. **Manual Builds**

- **Trigger Type:** Manual
- **Description:** Builds are triggered manually by clicking the "Build Now" button in the Jenkins dashboard. This is useful for ad-hoc or on-demand builds.

2. **SCM Polling (Checking for GitHub Changes)**

- **Trigger Type: S**CM Polling
- **Description:** Jenkins periodically checks the source code repository (e.g., GitHub) for changes. If changes are detected, a build is triggered automatically.

Build triggers

3. Webhooks (Trigger on GitHub Push Events)

- **Trigger Type:** Webhooks
- **Description:** GitHub sends notifications to Jenkins via webhooks when events like push occur. Jenkins listens for these notifications and triggers builds accordingly.

4. Cron-Based Scheduling

- **Trigger Type:** Scheduled
- **Description:** Builds are scheduled to run at specific intervals using cron expressions. For example, a build can be set to run every night at 2:00 AM using the cron expression 0 2 * * *13.

PW SKILLS

Demonstrating setting up a scheduled build using the Jenkins cron syntax

Let’s do it

To set up a scheduled build in Jenkins using the cron syntax, follow these steps:

1. Navigate to the Job Configuration
2. Enable Scheduled Builds
3. Enter Cron Syntax:

In the "Schedule" field, enter your desired cron expression. For example:

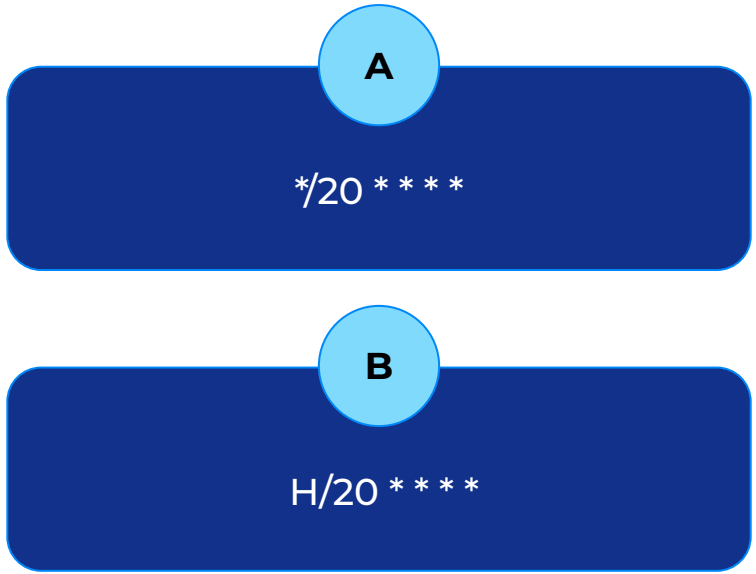
- Build every hour: H * * * *
- Build daily at midnight: 0 0 * * *
- Build every 20 minutes: H/20 * * * *

4. Save Changes:

- Click "Save" to apply your changes.

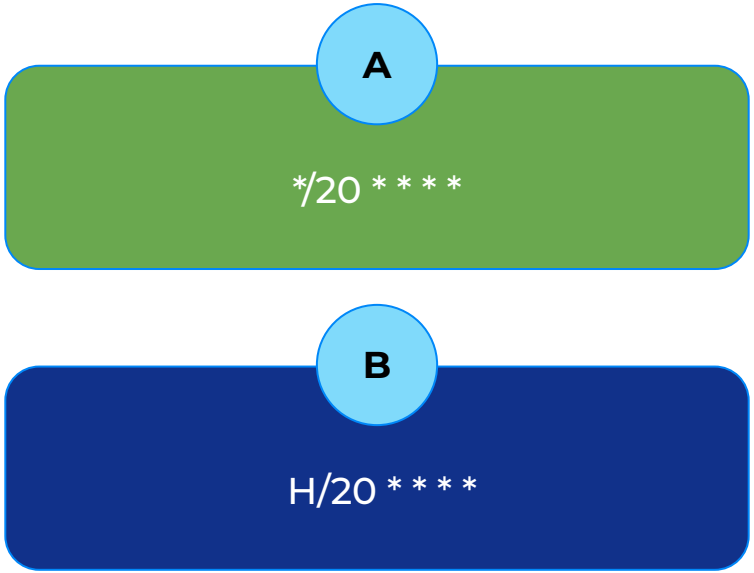
Pop Quiz

Q. To schedule a Jenkins job to run every 20 minutes, which cron expression should be used?



Pop Quiz

Q. To schedule a Jenkins job to run every 20 minutes, which cron expression should be used?



PW SKILLS

Time for case study!

Important

- Complete the post-class assessment
- Complete assignments (if any)
- Practice the concepts and techniques taught in this session
- Review your lecture notes
- Note down questions and queries regarding this session and consult the teaching assistants

Thanks !