

JENKINS

1. Plugins used in your project?

1. Maven
2. Active directory
3. Role based authorization strategy
4. Sonarqube scanner
5. GIT
6. Pipeline
7. Mailer
8. JUnit
9. Build Monitor

2. Master slave config in Jenkins. How we do?

- 1) Choose Manage Jenkins
- 2) Click on Manage Nodes and Clouds
- 3) Click New Node
- 4) Choose Node Name and Permanent Node
- 5) Configure the Below for the Node,
 - # of executors
 - Update Remote root directory
 - Name a Labels
 - Usage - Use this Node as much as Possible
 - Launch Method(Launch agent agents via SSH)
 - Add Hostname
 - Credentials
 - Manually Trusted Key Verificatio Strategy
 - Rest as Default

3. Upstream vs downstream jobs in Jenkins?

An upstream job is a configured project that triggers a project as part of its execution.

A downstream job is a configured project that is triggered as part of a execution of pipeline.

Upstream and downstream jobs help you to configure the sequence of execution for different operations and hence you can orchestrate the flow of execution. We can configure one or more projects as downstream jobs in Jenkins.

4. Parameterized job in Jenkins means?

- Create a freestyle project.
- Once you've created the job, Go to the configuration page and you'll find an unchecked box saying *This project is parametrized*. Check that button and make the project parameterized
- Once you check this button it'll give you options to choose the kind of parameters you want to use in your project. You can choose multiple parameters and add a default value to it.
- Once you've selected the required parameters and added their default values, execute this build.

5. How will you give access to the users in Jenkins?

To enable you to assign different roles and privileges to different users in Jenkins, you should have to install the **Role Strategy Plugin**.

Step 1: Open your Jenkins dashboard by visiting <http://localhost:8080/jenkins>

Steps 2: Click on 'Manage Jenkins' and select the 'Available' tab.

Step 3: On the filter option, type "role-based" and press Enter.

Step 4: Now, select the plugin and click on 'Install without restart' button.

Step 5: Click on 'Go back to the top page'.

Enable Role-Based Strategy on Jenkins

Step 1: After Plugin installation, go to the 'Manage Jenkins' and then click on 'Configure Global Security'. Check on Enable security option.

Step 2: On the Security Realm section, select 'jenkins' own user database'.

Step 3: You will be prompted to add your first user. Here, we are setting up an admin user for the system

Managing users and roles in Jenkins

Step 1)

1. Go to **Manage Jenkins**

2. Select **Manage and Assign Roles**

Step 2) Click on **Manage Roles** to add new roles based on your organization.

Step 3) To create a new role called "developer",

1. Type "developer" under "role".
2. Click on "Add" to create a new role.
3. Now, select the Jenkins user permissions you want to assign to the "Developer" role.
4. Click Save

How to Assign Roles in Jenkins

Step 1) Now that you have created roles, let us assign them to specific users.

1. Go to **Manage Jenkins**
2. Select Manage and Assign Roles

Step 2) We shall add the new role "developer" to user "**guru99**"

1. Selector developer role checkbox
2. Click Save

5. Backup on Jenkins?

Step 1 – Click on Manage Jenkins and choose the ‘Manage Plugins’ option.

Step 2 – In the available tab, search for ‘Backup Plugin’. Click On Install without Restart. Once done, restart the Jenkins instance

Step 3 – Now when you go to Manage Jenkins, and scroll down you will see ‘Backup Manager’ as an option. Click on this option.

Step 4 – Click on Setup.

Step 5 – Here, the main field to define is the directory for your backup. Ensure it’s on another drive which is different from the drive where your Jenkins instance is setup. Click on the Save button.

Step 6 – Click on the ‘Backup Hudson configuration’ from the Backup manager screen to initiate the backup.

The next screen will show the status of the backup

To recover from a backup, go to the Backup Manager screen, click on Restore Hudson configuration.

The list of backup’s will be shown, click on the appropriate one to click on Launch Restore to begin the restoration of the backup.

6. Scripted vs declarative pipeline?

- The **key difference** between Declarative pipeline and Scripted pipeline would be with respect to their **syntaxes** and their **flexibility**.
- Declarative pipeline is a relatively new feature that supports the pipeline as code concept. It makes the pipeline code easier to read and write. This code is written in a Jenkinsfile which can be checked into a source control management system such as Git.
- Whereas, the scripted pipeline is a traditional way of writing the code. In this pipeline, the Jenkinsfile is written on the Jenkins UI instance.
- Though both these pipelines are based on the groovy DSL, the scripted pipeline uses stricter groovy based syntaxes because it was the first pipeline to be built on the groovy foundation. Since this Groovy script was not typically desirable to all the users, the declarative pipeline was introduced to offer a simpler and more optioned Groovy syntax.

- The declarative pipeline is defined within a block labelled 'pipeline' whereas the scripted pipeline is defined within a 'node'.

7. what is multistage pipeline in jenkins?

8. Ways to secure Jenkins?

You should lock down the access to Jenkins UI so that users are authenticated and appropriate set of permissions are given to them. This setting is controlled mainly by two axes:

- **Security Realm**, which determines users and their passwords, as well as what groups the users belong to.

- **Authorization Strategy**, which determines who has access to what.

These two axes are orthogonal, and need to be individually configured. For example, you might choose to use external LDAP or Active Directory as the security realm, and you might choose "everyone full access once logged in" mode for authorization strategy. Or you might choose to let Jenkins run its own user database, and perform access control based on the permission/user matrix.

VPC