Caltech | Center for Technology & Management Education

Post Graduate Program in DevOps

Getting Started with Jenkins

Learning Objectives

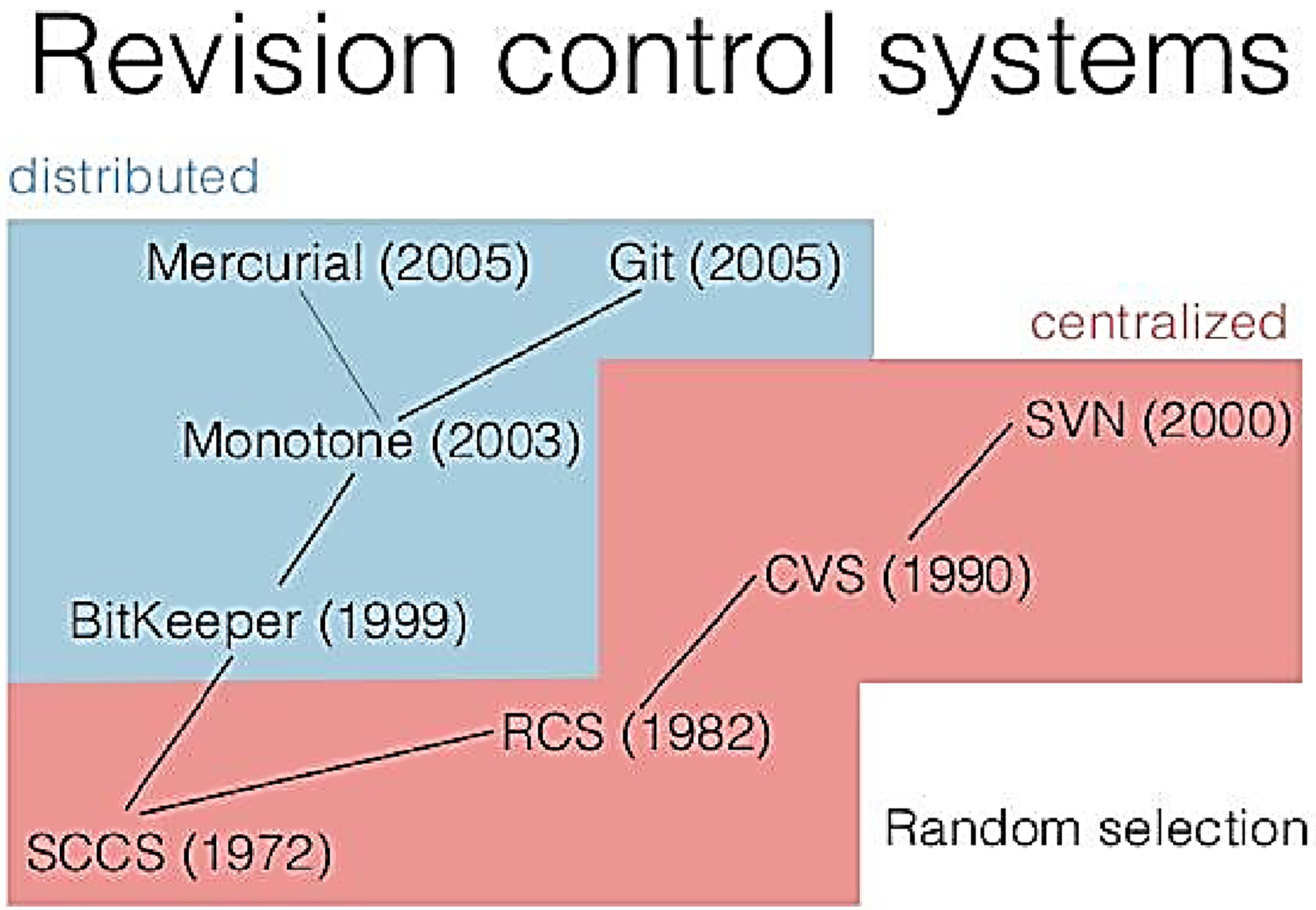
By the end of this lesson, you will be able to:

- Explain the basics of Jenkins
- Set up Jenkins in your machine
- Configure Git with SSH keys
- Explain Jenkins architecture
- Establish Maven and JDK in Jenkins

Setting Up Git

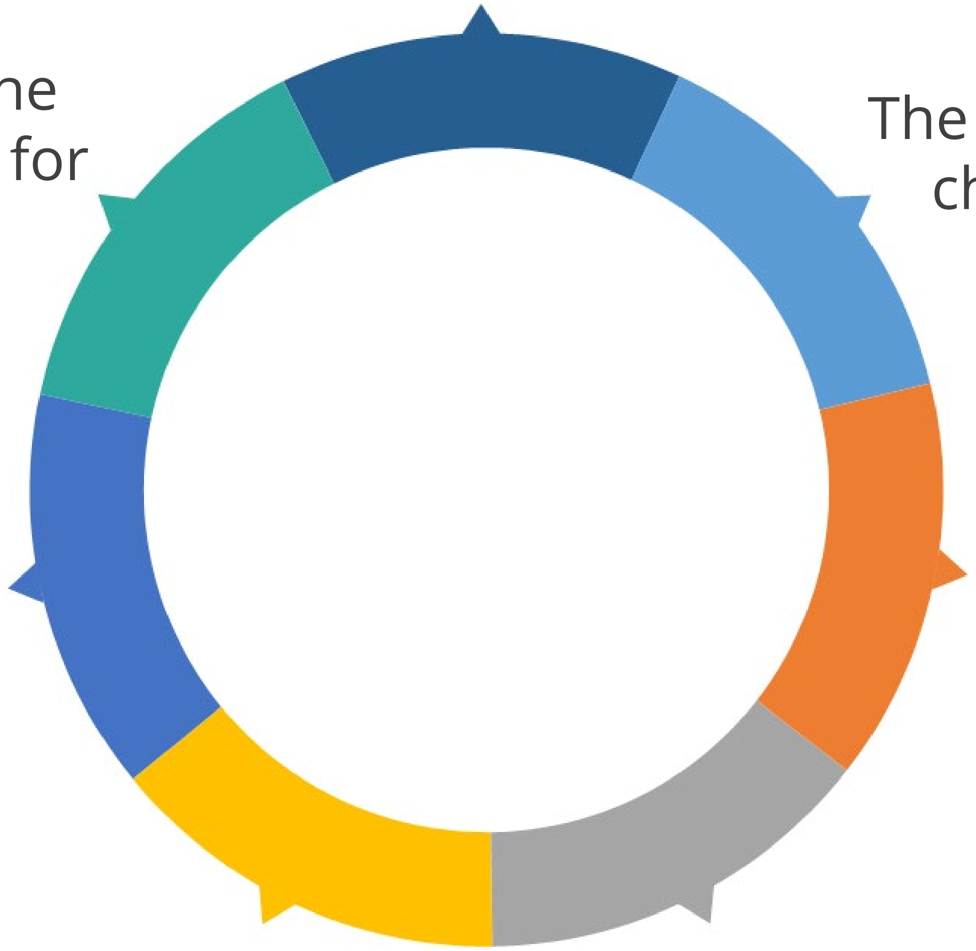
History of Version Control Systems

Version Control Systems (also known as revision control or source control) majorly focus on managing the changes to the files, programs, logs, and other information related to code development, code deployment, and code operation.



Need for Version Control Systems

Version control systems can be used to accomplish various tasks.



Files are checked into version control by registered users only.

The username and timestamp of the check-in are associated with the revision.

Each file check-in gets a new version which is usually a number.

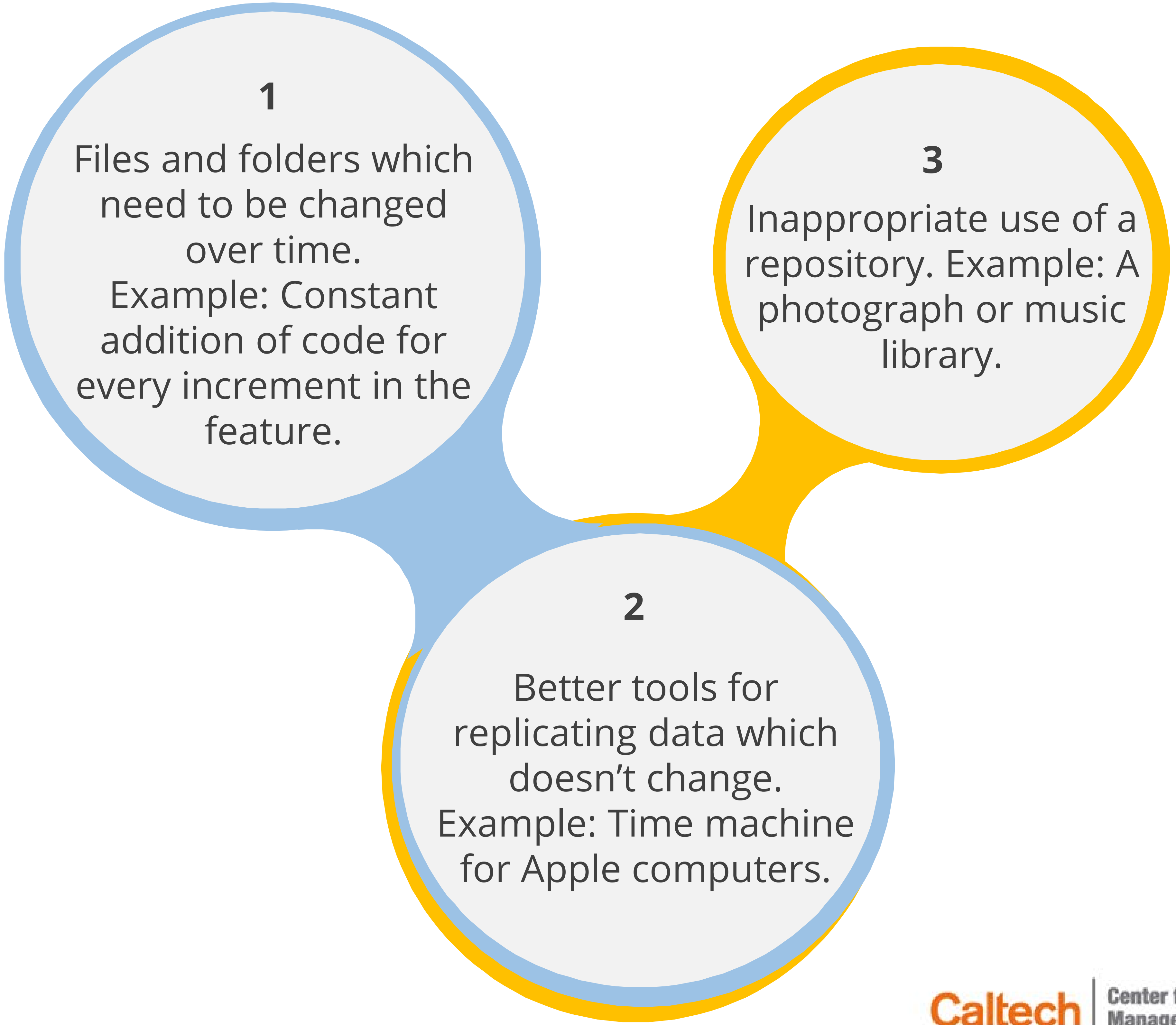
All the previous versions of a file can be easily extracted.

The latest versions of all files are often referred to as “heads.”

Source control systems are organized into repositories.

Supports branching where the head of the repository is split for parallel development.

Repository Usage



1

Files and folders which need to be changed over time.
Example: Constant addition of code for every increment in the feature.

2

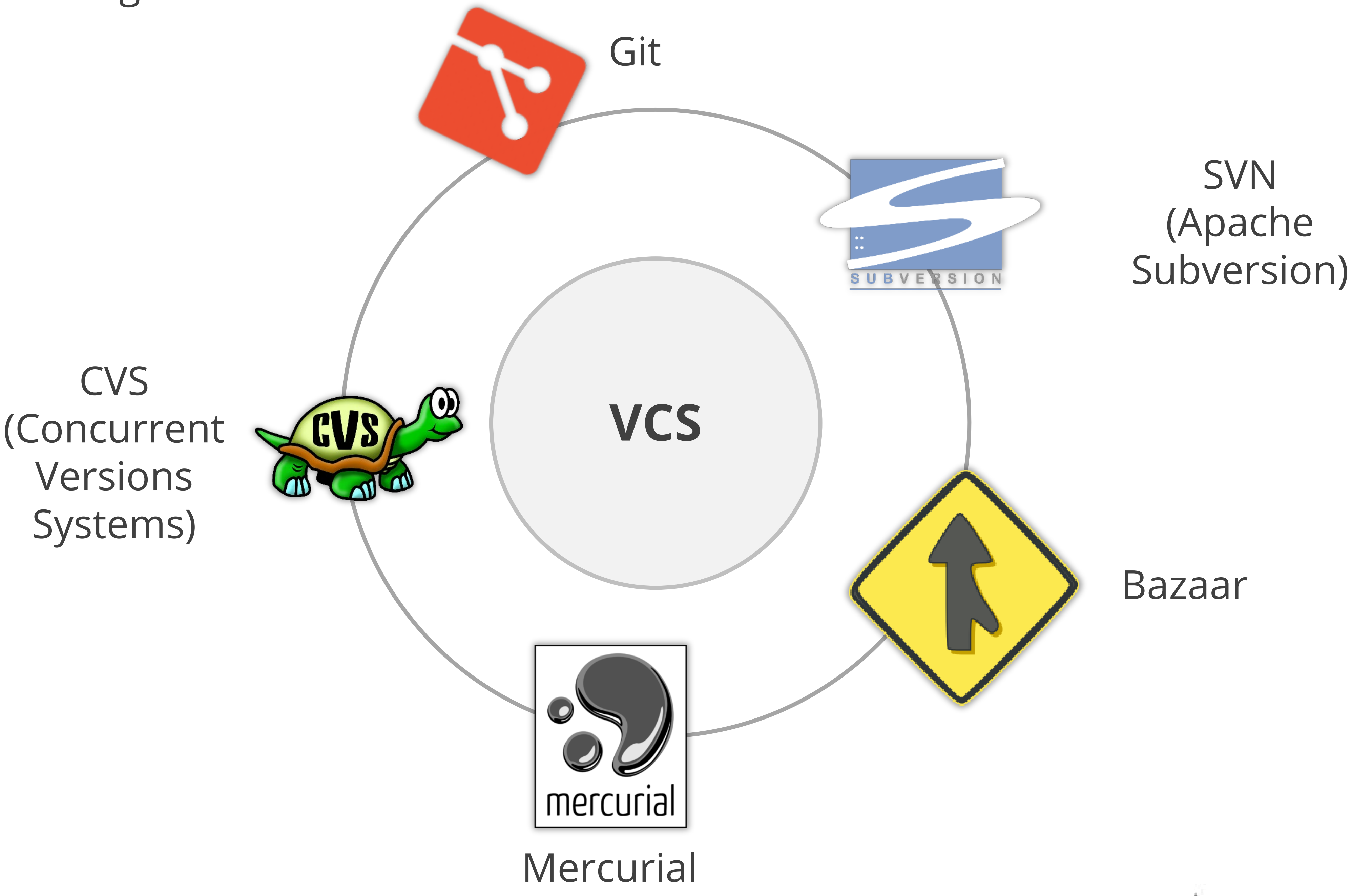
Better tools for replicating data which doesn’t change.
Example: Time machine for Apple computers.

3

Inappropriate use of a repository. Example: A photograph or music library.

Popular Version Control Systems

Some of the most preferred and popular open-source version control systems and tools are listed in the image below:



VCS

Git

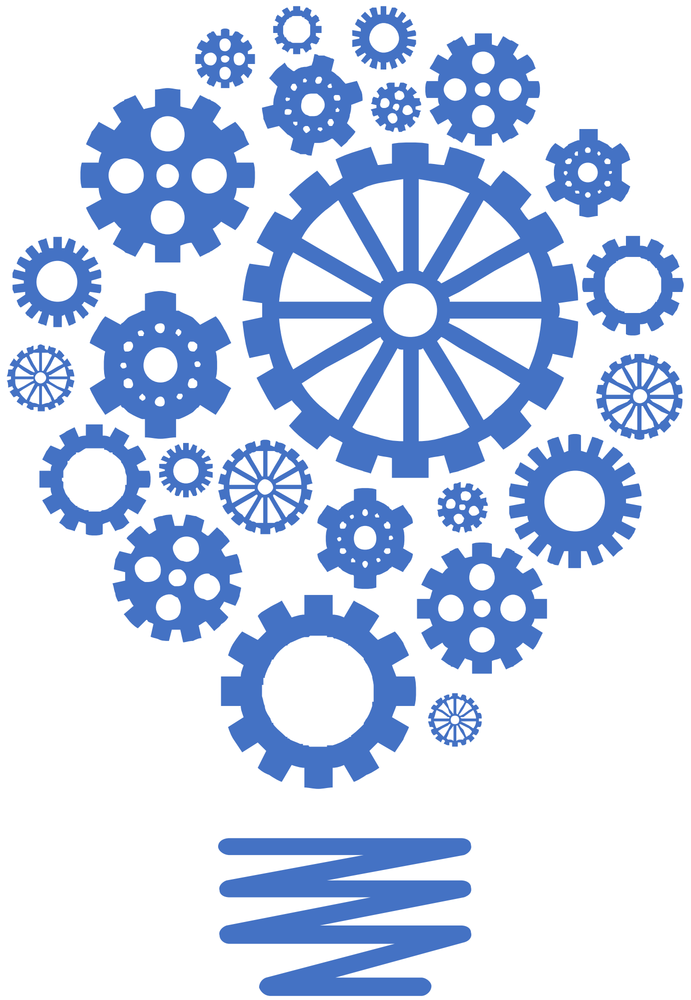
SVN (Apache Subversion)

Bazaar

Mercurial

CVS (Concurrent Versions Systems)

Role of VCS in DevOps



Organize product-related artifacts

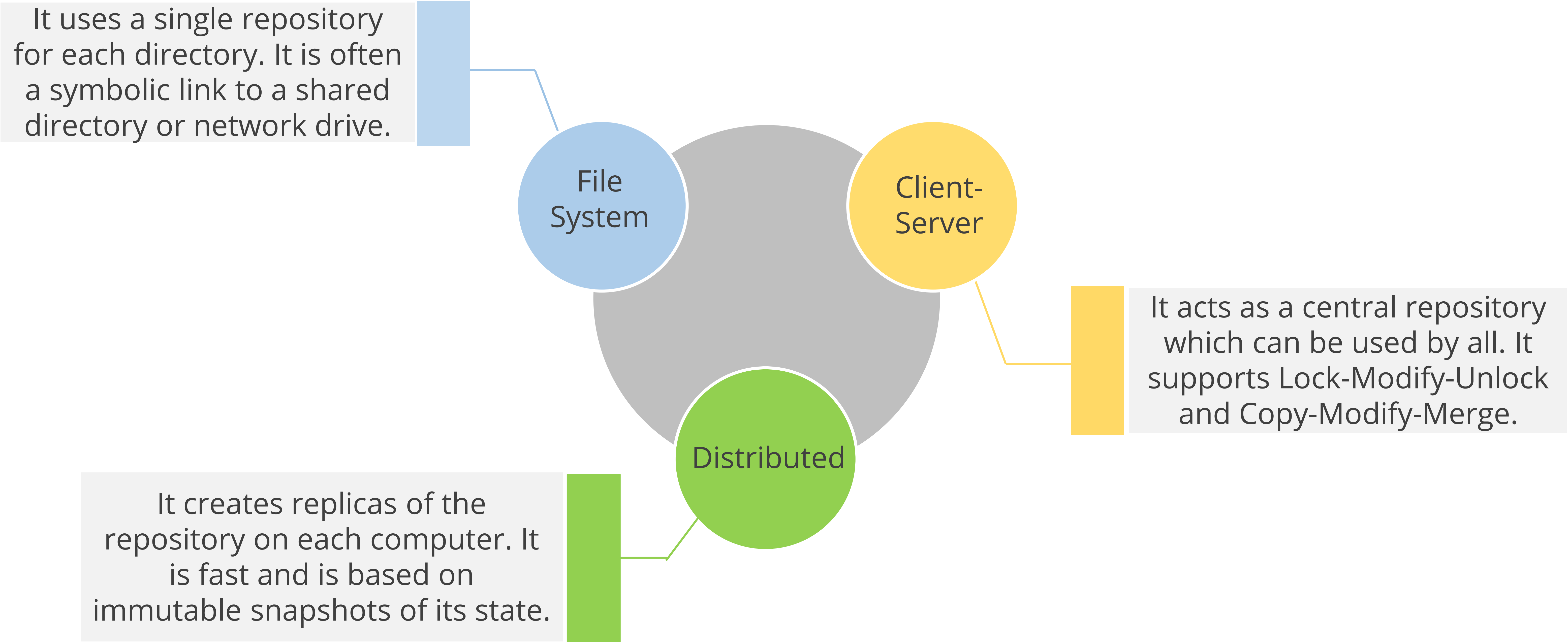
Adopt peer reviews to inspect changes

Provide tried and tested guaranteed results

Regulate monitoring of documents

Establish win-win relationship between Dev and Ops

Types of VCS

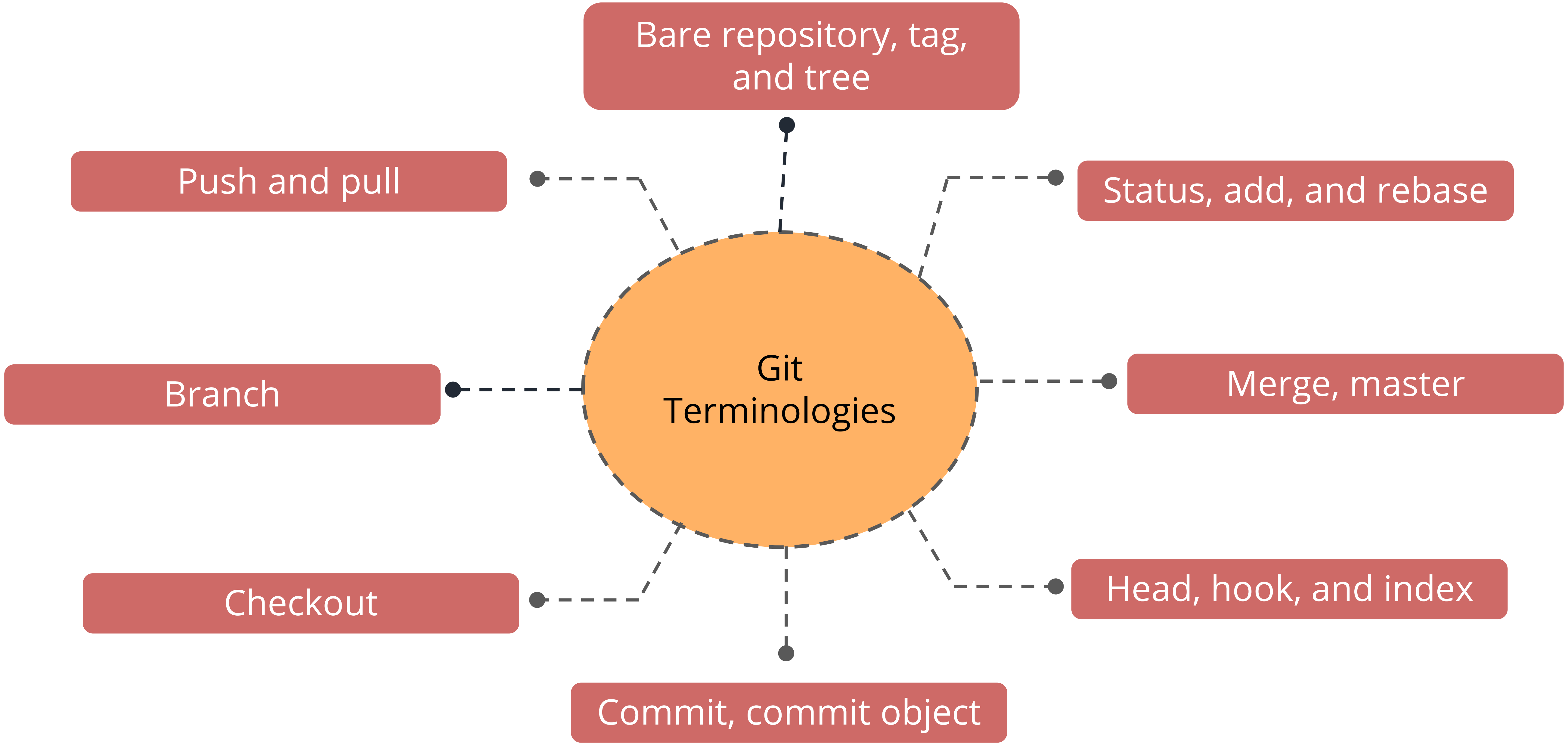


It uses a single repository for each directory. It is often a symbolic link to a shared directory or network drive.

It acts as a central repository which can be used by all. It supports Lock-Modify-Unlock and Copy-Modify-Merge.

It creates replicas of the repository on each computer. It is fast and is based on immutable snapshots of its state.

Git Terminologies


Git Terminologies

Bare repository, tag, and tree

Status, add, and rebase

Merge, master

Head, hook, and index

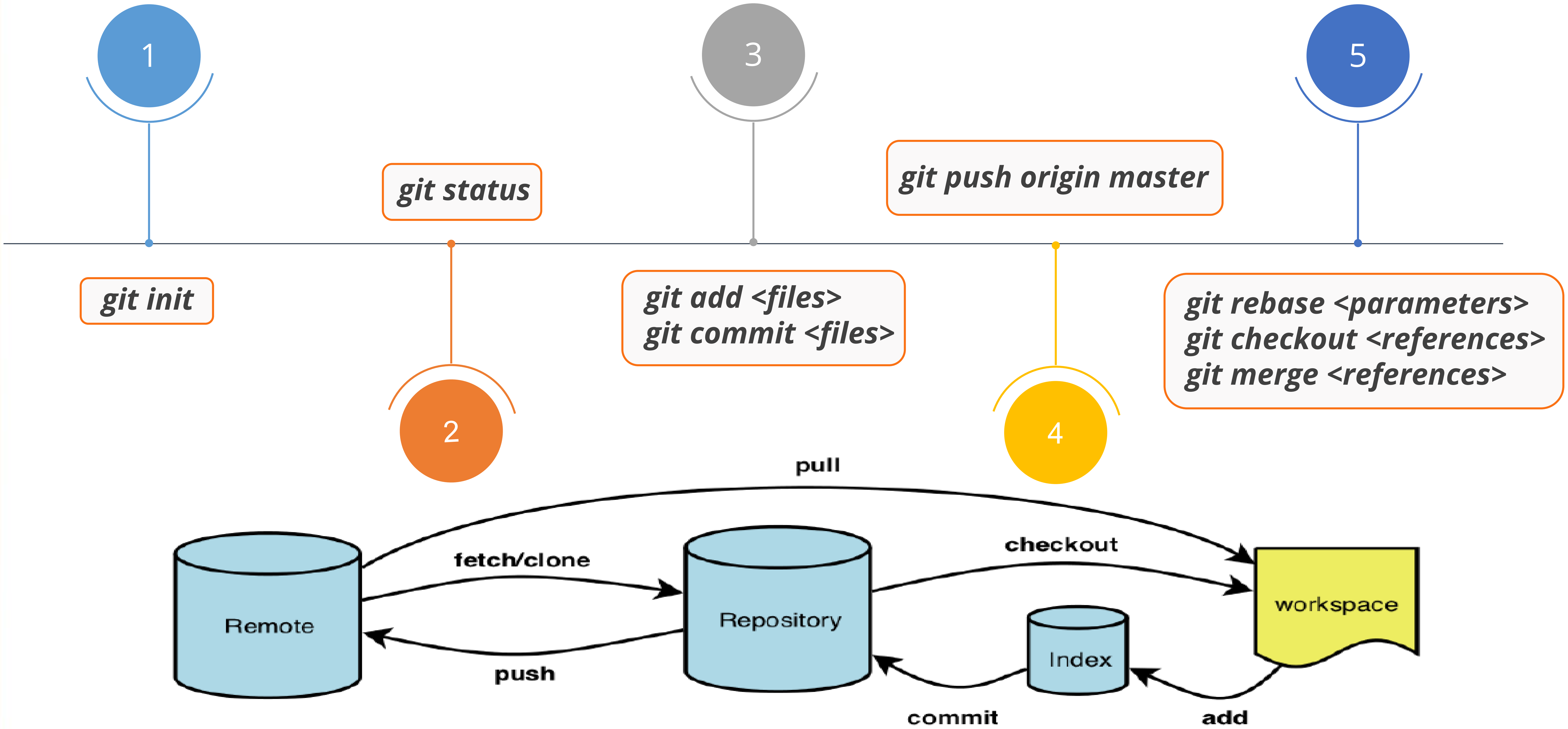
Commit, commit object

Checkout

Branch

Push and pull

Git Workflow



1 *git init*

2 *git status*

3 *git add <files>*
git commit <files>

4 *git push origin master*

5 *git rebase <parameters>*
git checkout <references>
git merge <references>

Git Installation

Git can be installed in both Linux and Windows platforms. Follow the below instructions to install Git to your system:

For Windows:

- Download Git from the below website and run the executable file in your machine and Git will be installed to your system.

https://git-scm.com/download/win

For Linux (Ubuntu or Debian):

- Run the command given below in the terminal to install Git to your system.

```
sudo spt-get install git
```

For Fedora (any other RPM-based distributed system):

- Run the command given below in the terminal to install Git to your system

```
sudo yum install git-core
```

Assisted Practice

Git Setup

Problem Statement: You are given a project to push files from the local Git folder to the GitHub repository.

Steps to perform:

1. Create an account in GitHub and create a repository.
2. Initialize the Git repo locally.
3. Create files with every extension, and add random content in it (.txt, .html, .java).
4. Sign in to the GitHub account.
5. Generate SSH key for local Git repository.
6. Connect the local and remote repository using SSH key.
7. Add the codes to the repositories of GitHub.

Setting Up Jenkins

What Is Jenkins?

Definition by Jenkins.io: Jenkins is a self-contained, open-source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

- One can install Jenkins as a standalone software, Docker, and also as a package on the machine that has JRE pre-installed.

Prerequisites:

1. A machine with 256 MB of RAM, although more than 512MB is recommended.

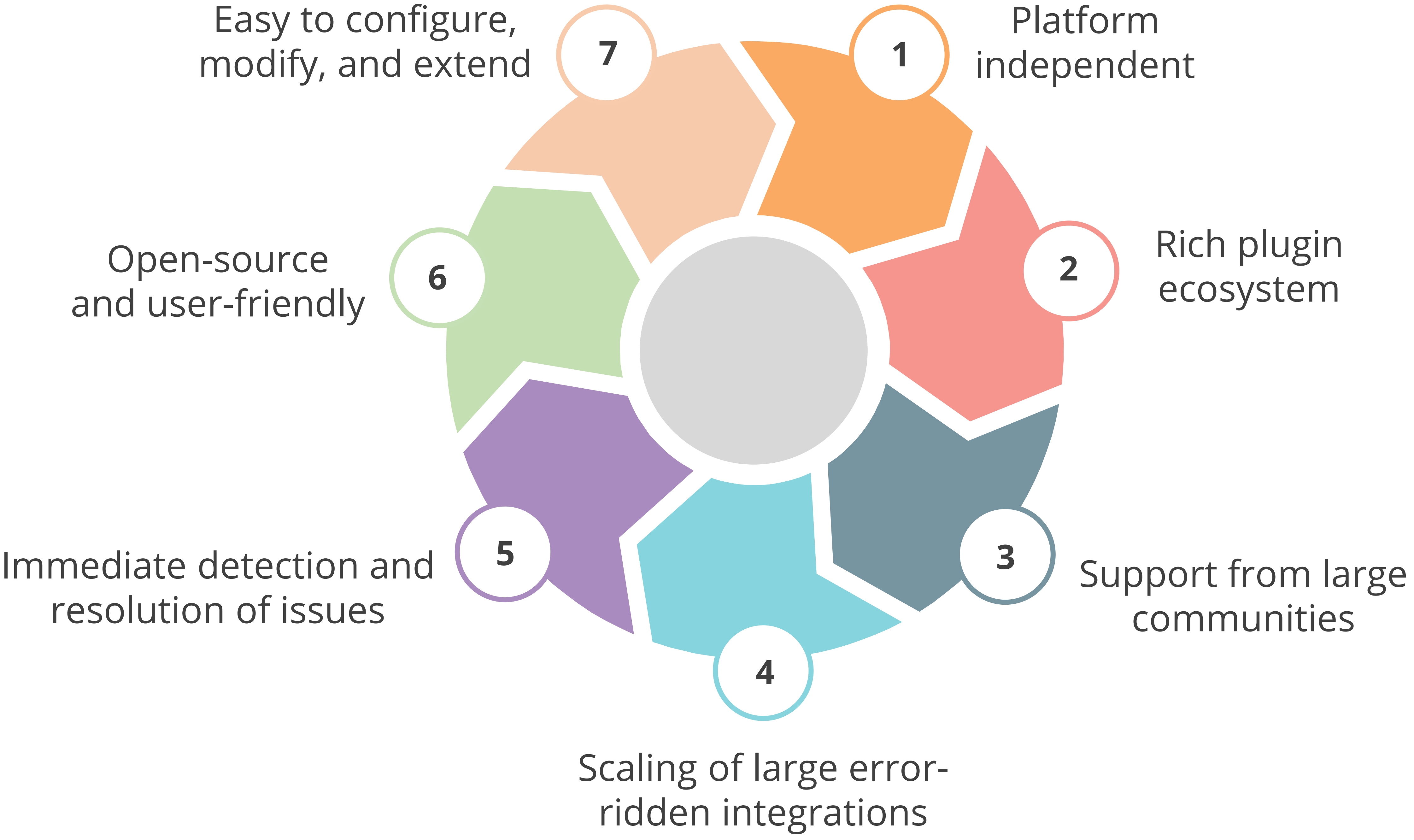
1. 10 GB of drive space (for Jenkins and your Docker image)

The following software installed:

1. Java 8 (either a JRE or Java Development Kit (JDK) is fine)

1. Docker suitable for your system.

Features of Jenkins



7 Easy to configure, modify, and extend

1 Platform independent

6 Open-source and user-friendly

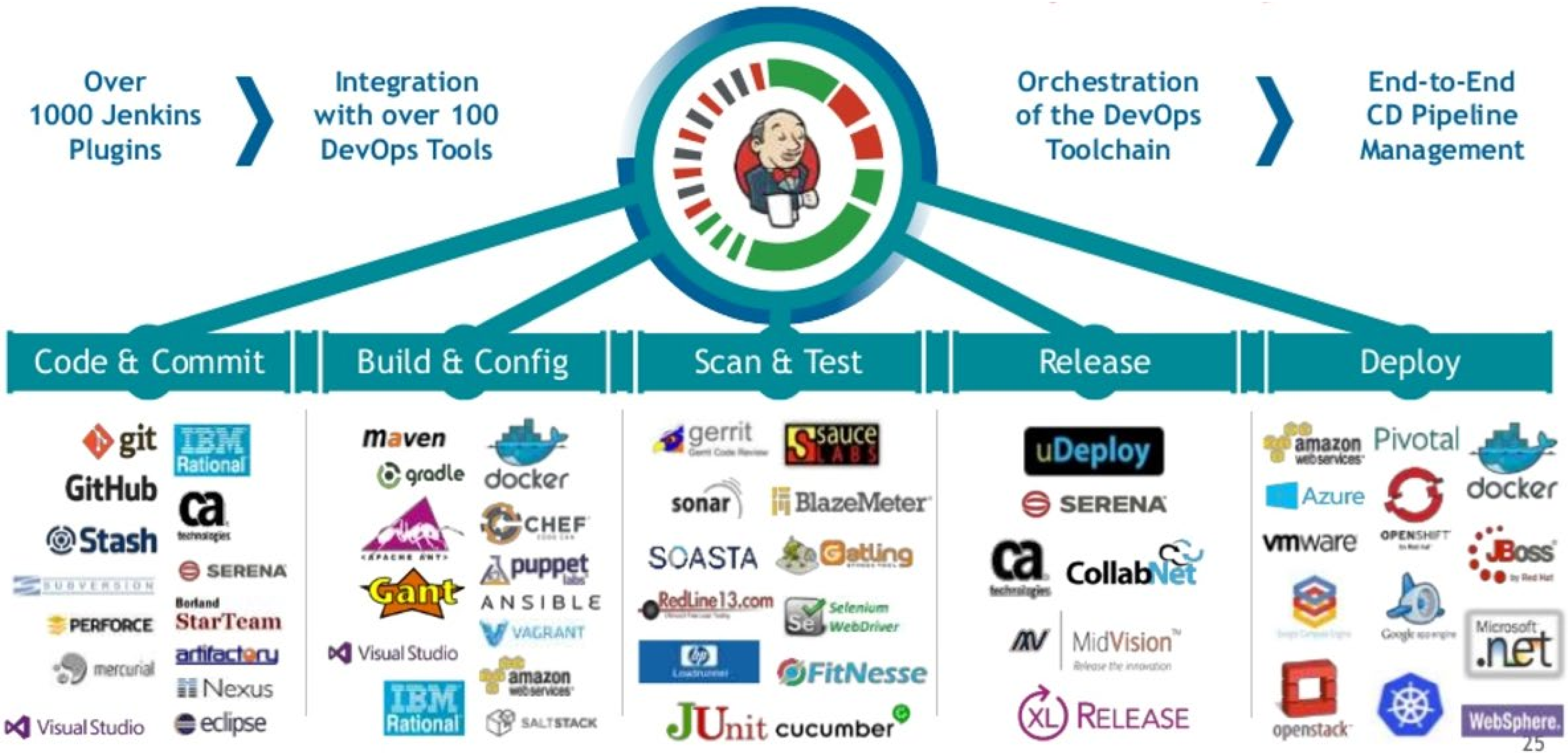
2 Rich plugin ecosystem

5 Immediate detection and resolution of issues

3 Support from large communities

4 Scaling of large error-ridden integrations

CI with Jenkins



Jenkins Installation

Jenkins can be installed in both Linux and Windows platforms. Follow the below instructions to install Jenkins to your system:

For Windows:

- Download Jenkins from the below website and run the executable file in your machine and Jenkins will be installed to your system.

https://jenkins.io/download/thank-you-downloading-windows-installer-stable/

For Linux (Ubuntu or Debian):

- Run the command given below in the terminal to install Jenkins to your system.

```
sudo apt-get install jenkins
```

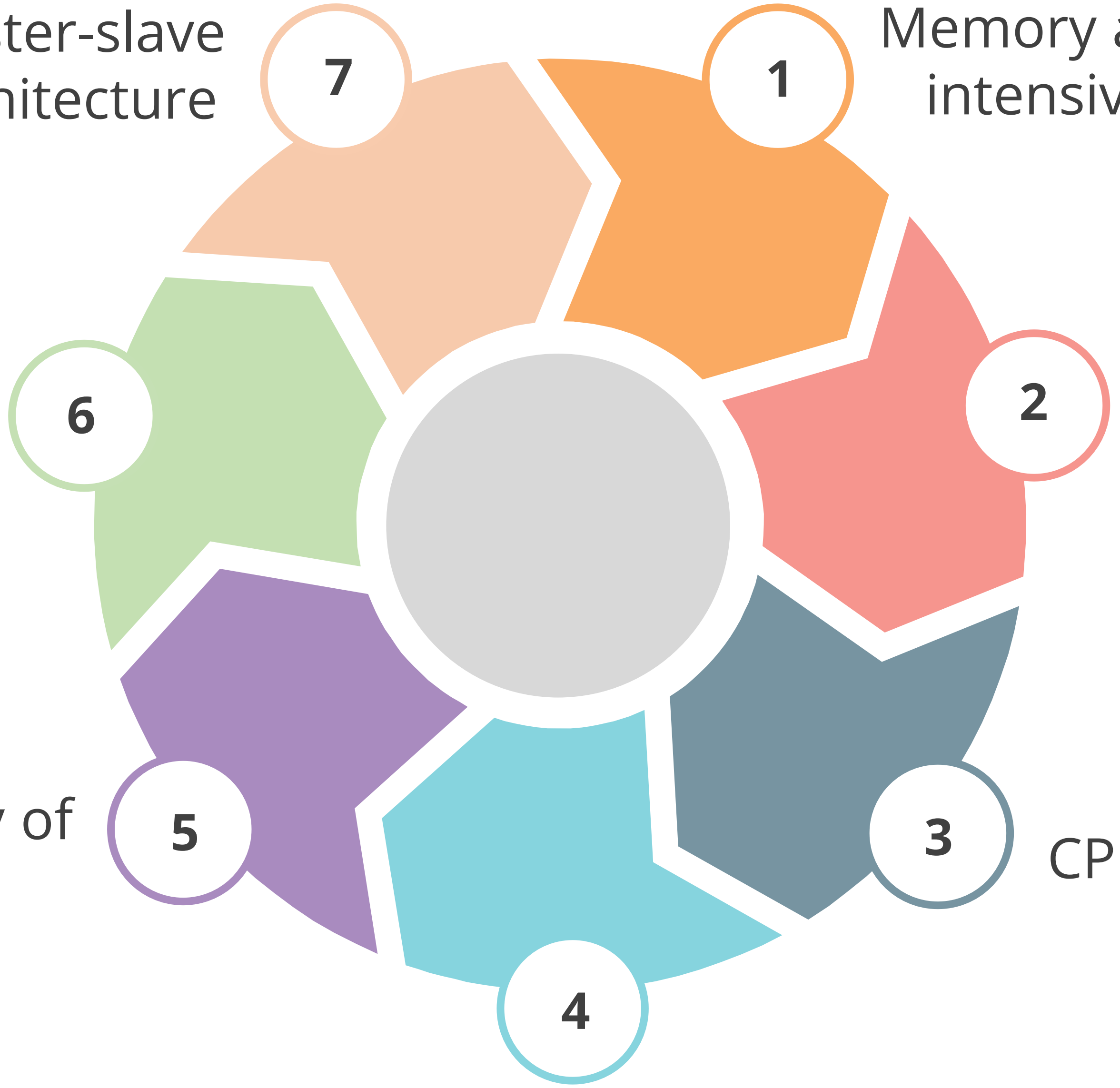
For Fedora(any other RPM-based distributed system):

- Run the command given below in the terminal to install Jenkins in your system

```
sudo yum install jenkins
```

Building A Build Server

Installing Jenkins on your local development machine is one thing, but installing Jenkins on a proper build server deserves a little more forethought and planning. Below are the points to be considered while preparing a reliable build server:



1. Memory and processor-intensive operations
2. Number of build jobs
3. CPU horsepower
4. One processor per parallel build
5. Type and complexity of application
6. Use a virtual machine
7. Master-slave architecture

Building A Build Server

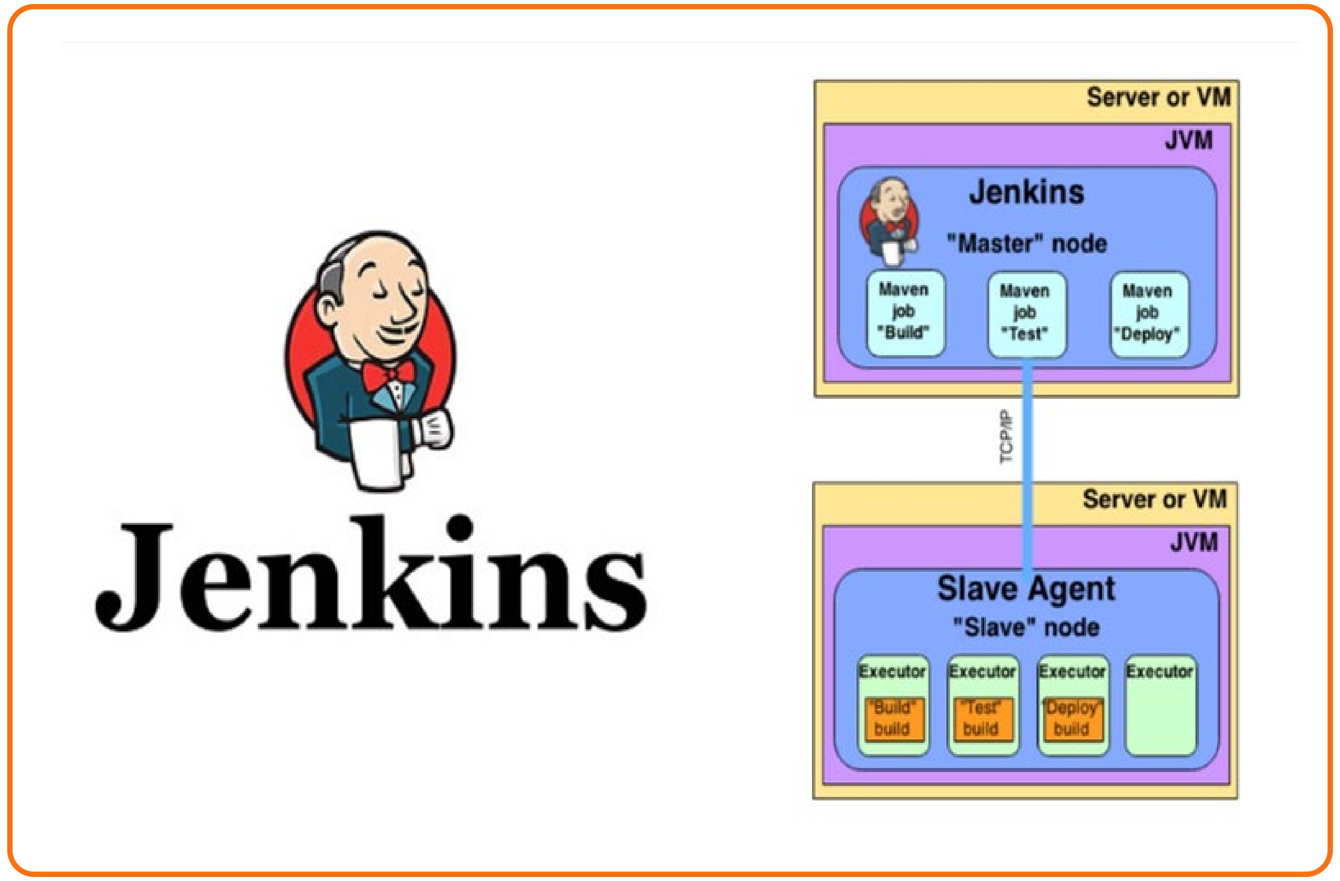
- If you are installing Jenkins on a Linux or Unix build server, it is a good idea to create a special user (and user group) for Jenkins.

- This makes it easier to monitor the system resources being used by the Jenkins builds, and to troubleshoot problematic builds in real conditions.

- The native binary installation packages, discussed below, do this for you. If you did not use one of these, you can create a dedicated Jenkins user.

- Run the following command as shown below or you can use Jenkins UI to do the same:

```
$ sudo groupadd build
$ sudo useradd --create-home --shell /bin/bash --groups build jenkins
```



Assisted Practice

Jenkins Setup

Problem Statement: You are given a project to set up and run Jenkins on your system.

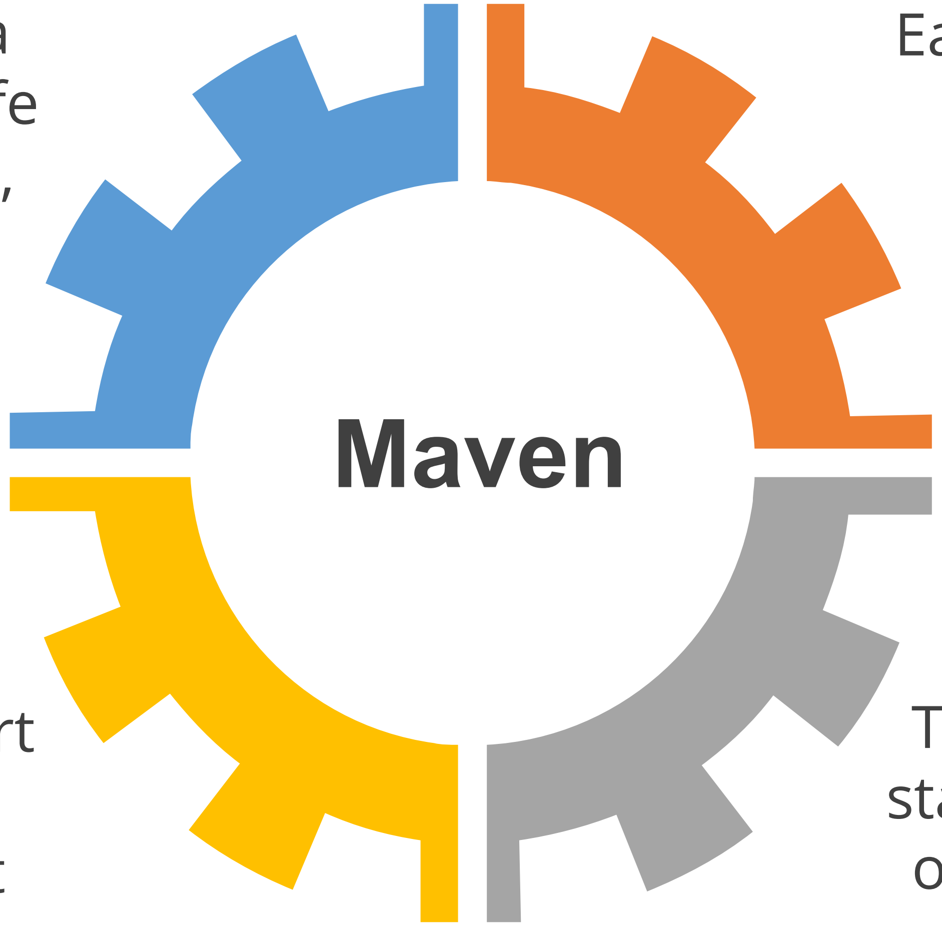
Steps to perform:

1. Download the Java Runtime Environment
2. Download and installing the Jenkins app

Configuring Dev Tools

Maven Configuration

Maven is a high-level build scripting framework for Java that uses notions such as a standard directory structure and standard life cycles, Convention over Configuration, and Declarative Dependency Management to simplify a lot of the low-level scripting that you find in a typical Ant build script.



In Maven, your project uses a standard, well defined build life cycle—compile, test, package, deploy, and so forth.

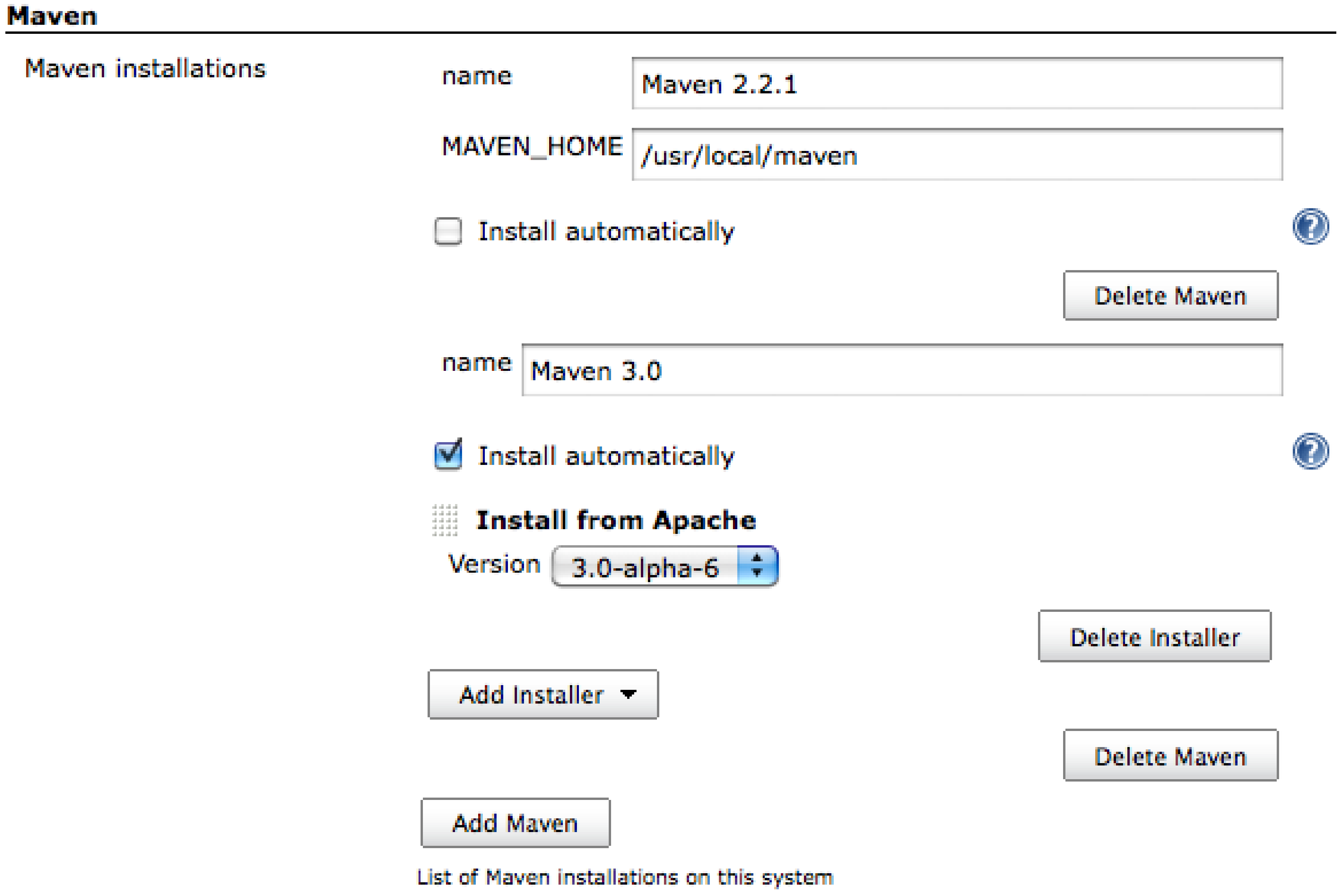
Each life cycle phase is associated with a Maven plugin.

Jenkins provides excellent support for Maven, and has a good understanding of Maven project structures and dependencies

The various Maven plugins use the standard directory structure to carry out these tasks with a minimum of intervention on your part.

Maven Configuration

You can either get Jenkins to install a specific version of Maven automatically or provide a path to a local Maven installation



Maven

Maven installations

name: Maven 2.2.1

MAVEN_HOME: /usr/local/maven

□ Install automatically

name: Maven 3.0

☑ Install automatically

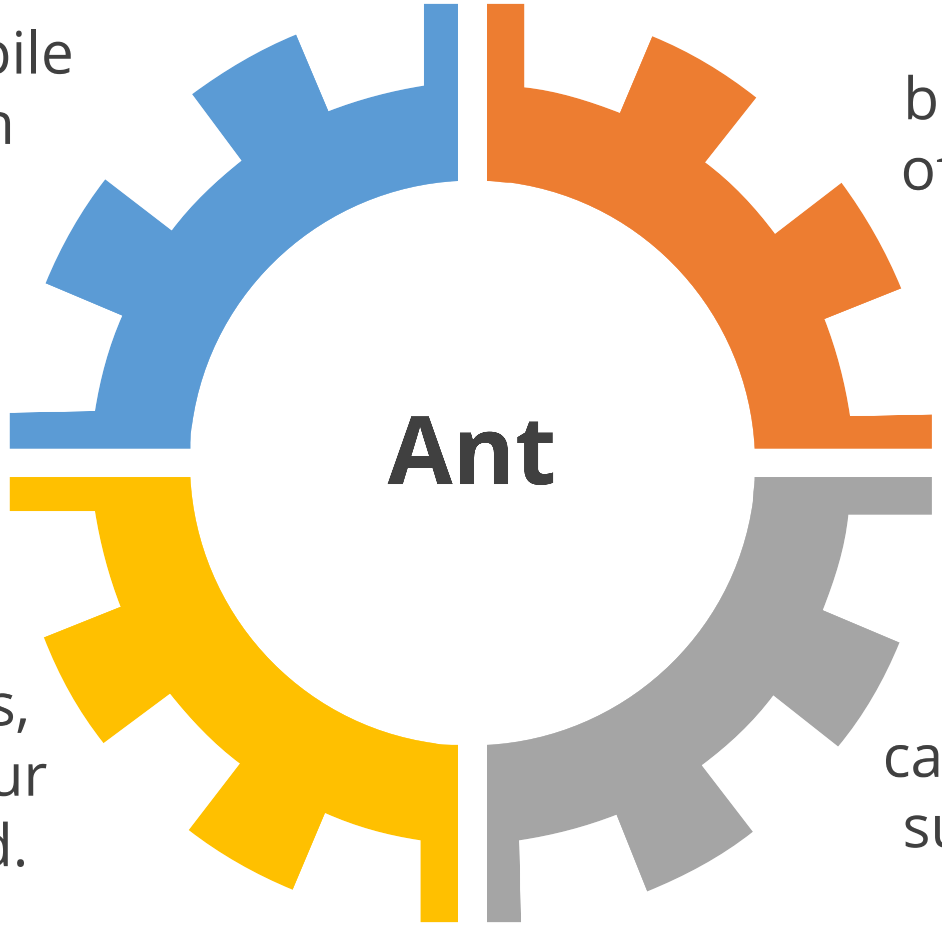
Install from Apache

Version: 3.0-alpha-6

List of Maven installations on this system

Ant Configuration

Ant is a widely-used and very well-known build scripting language for Java. It is a flexible, extensible, relatively low-level scripting language, used in a large number of open source projects.



An Ant build script (typically build.xml) is made up of a number of targets, which perform different tasks in build process.

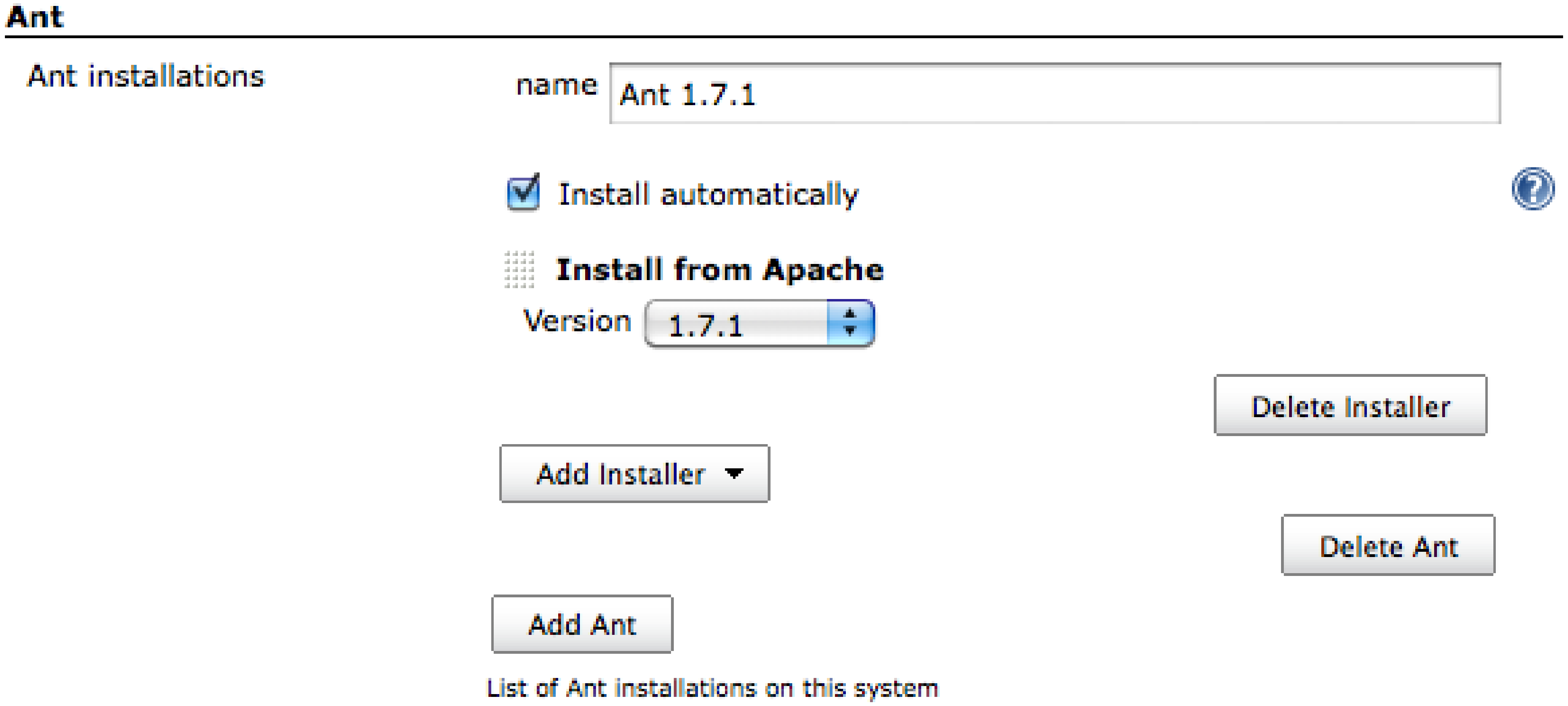
It does so by executing tasks, which carry out a specific part of the build job, such as invoking javac to compile your code, or creating a new directory.

Targets also have dependencies, indicating the order in which your build tasks need to be executed.

For example, you need to compile your code before you can run your unit tests.

Ant Configuration

Jenkins provides excellent build-in support for Ant. If Ant is available on the system path, Jenkins will find it. However, you can configure as many installations of Ant as required by providing name and installation directory of the versions.



Ant

Ant installations

name: Ant 1.7.1

☑ Install automatically

Install from Apache

Version: 1.7.1

List of Ant installations on this system

Assisted Practice

Maven Set Up

Problem Statement: You are given a project to set up Maven with Jenkins.

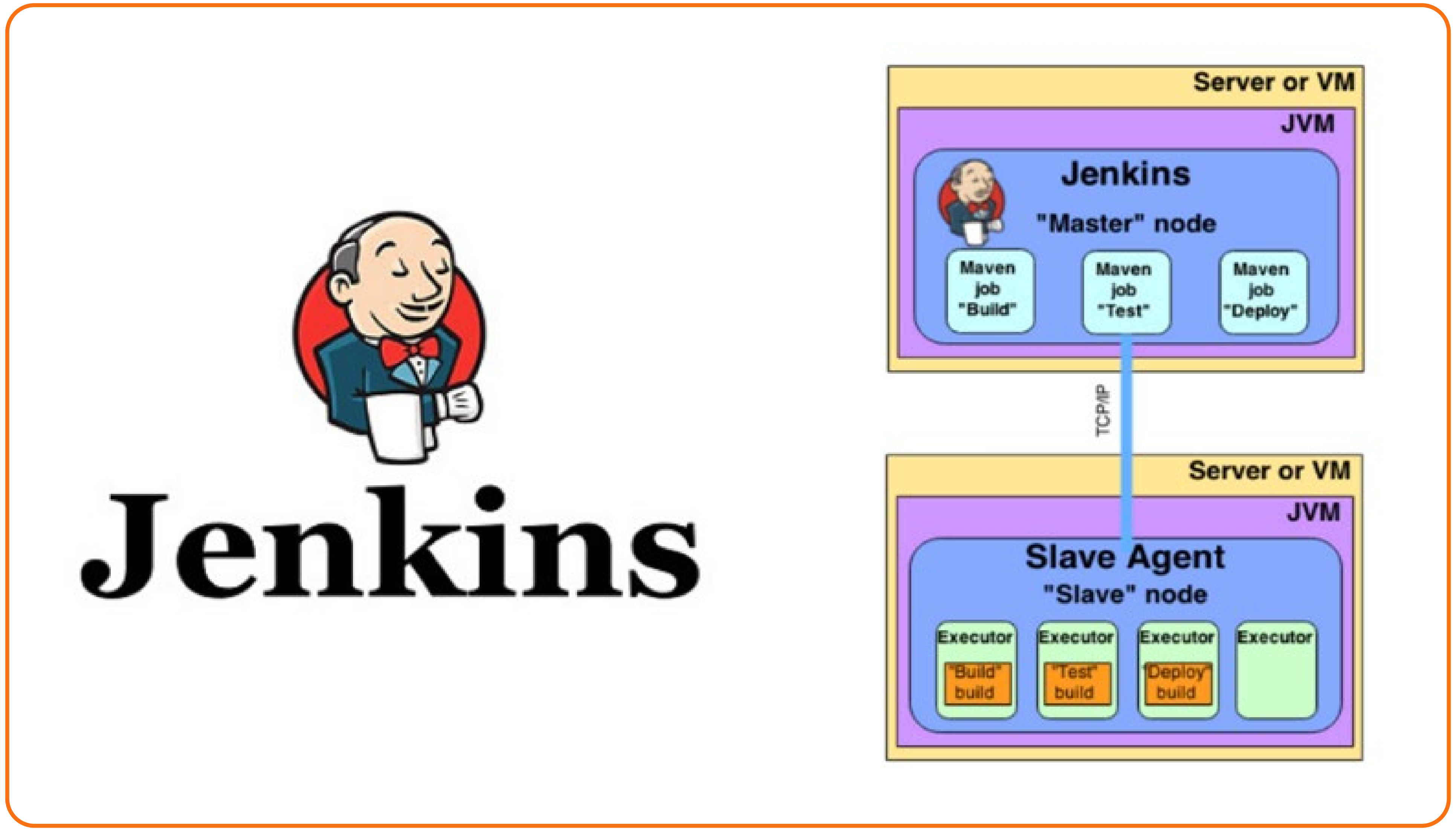
Steps to perform:

1. Configure the Maven
2. Configure the JDK

Exploring Jenkins

Jenkins Architecture

Jenkins follows the **Master-Slave** architecture to manage builds. Slave and master communicate through a TCP/IP connection.



Jenkins Master and Slave Architecture

Jenkins Master

- By default, it runs on the 8080 port.
- With the help of dashboard, jobs can be configured
- The build is executed in the slave.
- Can add more nodes using IP address, username and password using the ssh, jnlp or webstart methods.
- The master's job is to:
  1. Scheduling build jobs
  2. Dispatching builds to the slaves
  3. Monitoring the slaves
  4. Recording and displaying the build results

Jenkins Master/Slave Architecture

Jenkins Slave

- Jenkins slave is used to execute the build jobs.
- Build jobs are distributed by the master node.
- You can configure a project to always run on a particular slave machine.
- By default, Jenkins automatically selects the available slave node for executing builds.
- As Jenkins is developed using Java, it is platform independent and can be configured in Linux, Windows, and Mac.

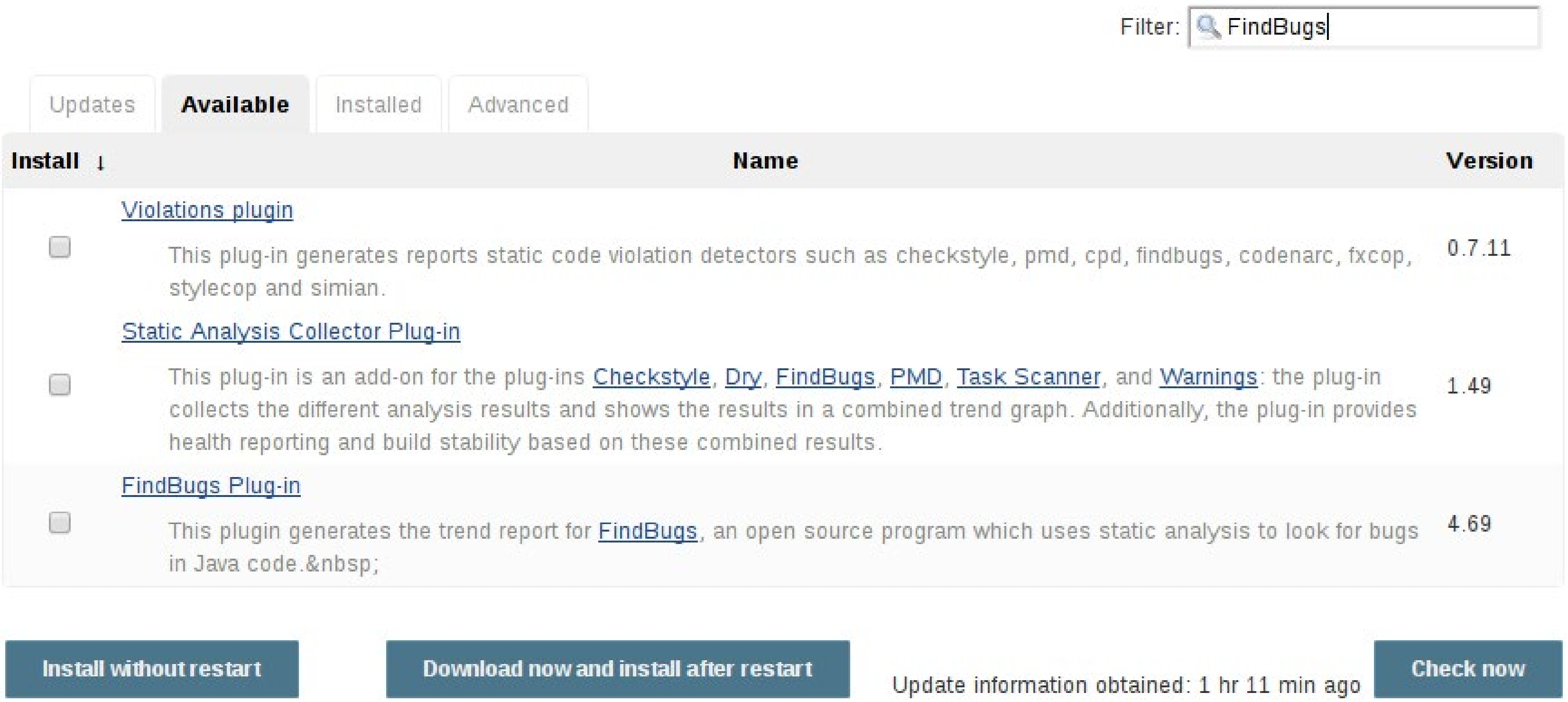
Plugins

Plugins are used to enhance the functionality of a Jenkins environment to complement and support the needs that are organization, or user-specific.

- The different ways to install plugins are:

1. Using the **Plugin Manager** in the web UI (the screenshot is given below)

2. Using the Jenkins CLI **install-plugin** command



Filter: FindBugs

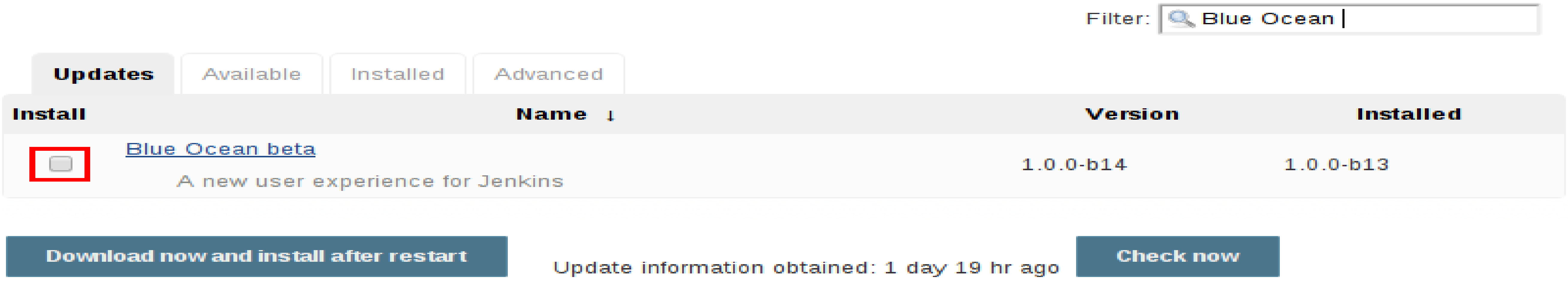
Install ↓	Name	Version
□	Violations plugin This plug-in generates reports static code violation detectors such as checkstyle, pmd, cpd, findbugs, codenarc, fxcop, stylecop and simian.	0.7.11
□	Static Analysis Collector Plug-in This plug-in is an add-on for the plug-ins Checkstyle, Dry, FindBugs, PMD, Task Scanner, and Warnings: the plug-in collects the different analysis results and shows the results in a combined trend graph. Additionally, the plug-in provides health reporting and build stability based on these combined results.	1.49
□	FindBugs Plug-in This plugin generates the trend report for FindBugs, an open source program which uses static analysis to look for bugs in Java code.	4.69

Update information obtained: 1 hr 11 min ago

Plugin Operations

Update and delete are the two operations that can be performed on a plugin.

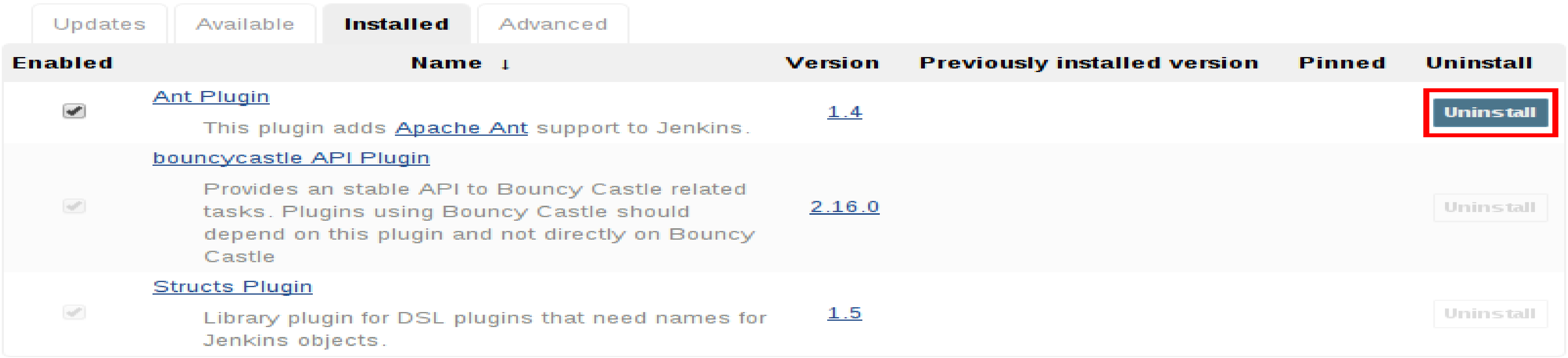
The screenshots of updating and deleting a plugin from the Plugin Manager are given below:



Filter: Blue Ocean

Install	Name ↓	Version	Installed
□	Blue Ocean beta A new user experience for Jenkins	1.0.0-b14	1.0.0-b13

Update information obtained: 1 day 19 hr ago



Enabled	Name ↓	Version	Previously installed version	Pinned	Uninstall
☑	Ant Plugin This plugin adds Apache Ant support to Jenkins.	1.4			Uninstall
☑	bouncycastle API Plugin Provides an stable API to Bouncy Castle related tasks. Plugins using Bouncy Castle should depend on this plugin and not directly on Bouncy Castle	2.16.0			Uninstall
☑	Structs Plugin Library plugin for DSL plugins that need names for Jenkins objects.	1.5			Uninstall

Notifications

- The importance of setting up notifications in Jenkins include:
- When a Jenkins build breaks, the entire team is notified via email messages
- Using plugins, you can also send instant SMS messages, or post entries on Twitter.
- Easy set up.
- If the local SMTP server address is present, you can just provide its value in the email notification section

Security

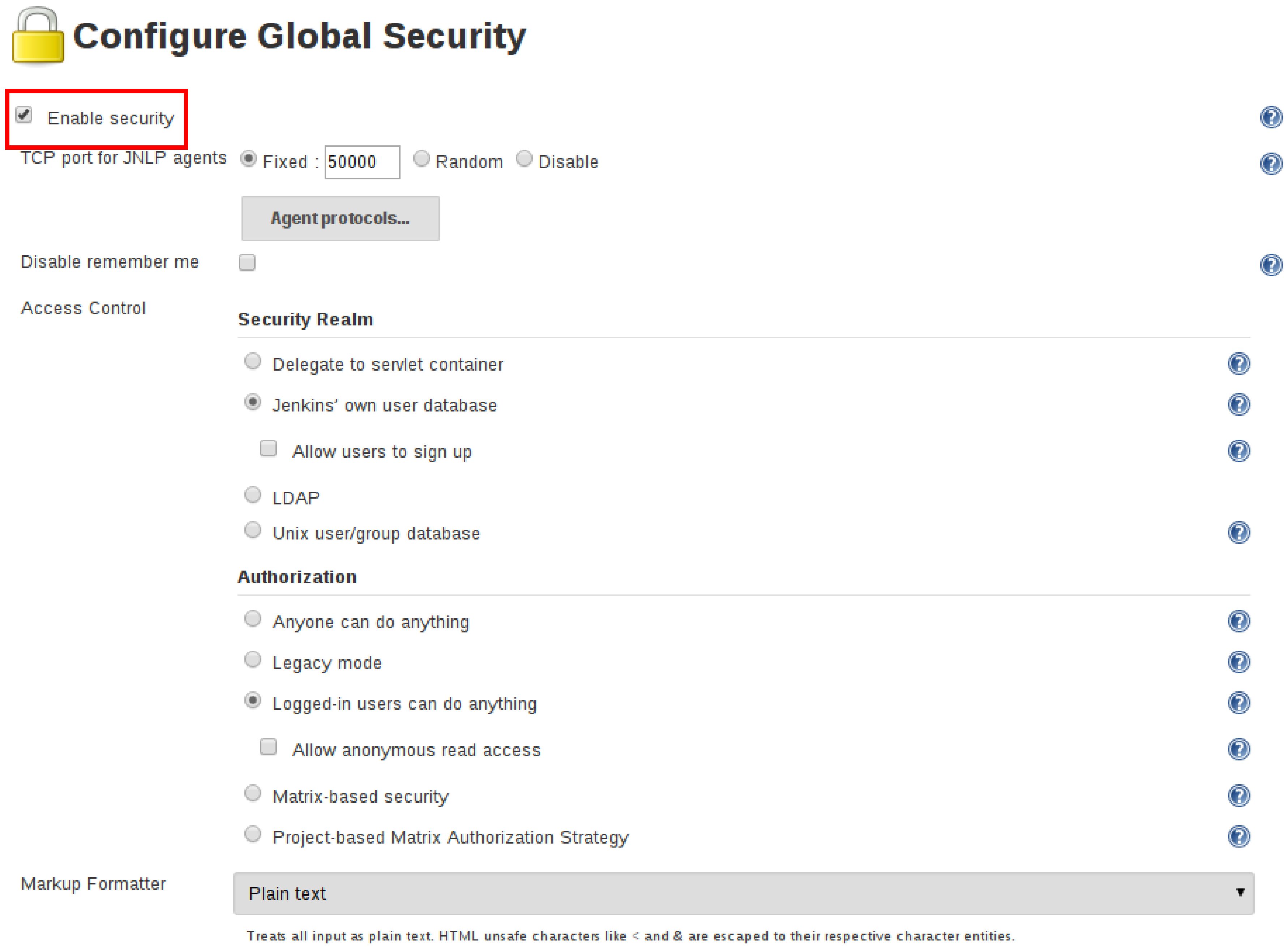
As Jenkins is used in many organization, Jenkins offers many configuration options for enabling, editing, or disabling various security features.

- **Enable security** section allows a Jenkins administrator to enable, configure, or disable security features.

- Access control is the primary security mechanism for a Jenkins environment and have two major parameters:

1. **Security Realm** which informs the Jenkins environment about the user authentication.

1. **Authorization** configuration states the users and/or groups that have the permission to access Jenkins.



Configure Global Security

☑ Enable security

TCP port for JNLP agents ● Fixed : 50000 ○ Random ○ Disable

Agent protocols...

Disable remember me □

Access Control

Security Realm

○ Delegate to servlet container

● Jenkins' own user database

□ Allow users to sign up

○ LDAP

○ Unix user/group database

Authorization

○ Anyone can do anything

○ Legacy mode

● Logged-in users can do anything

□ Allow anonymous read access

○ Matrix-based security

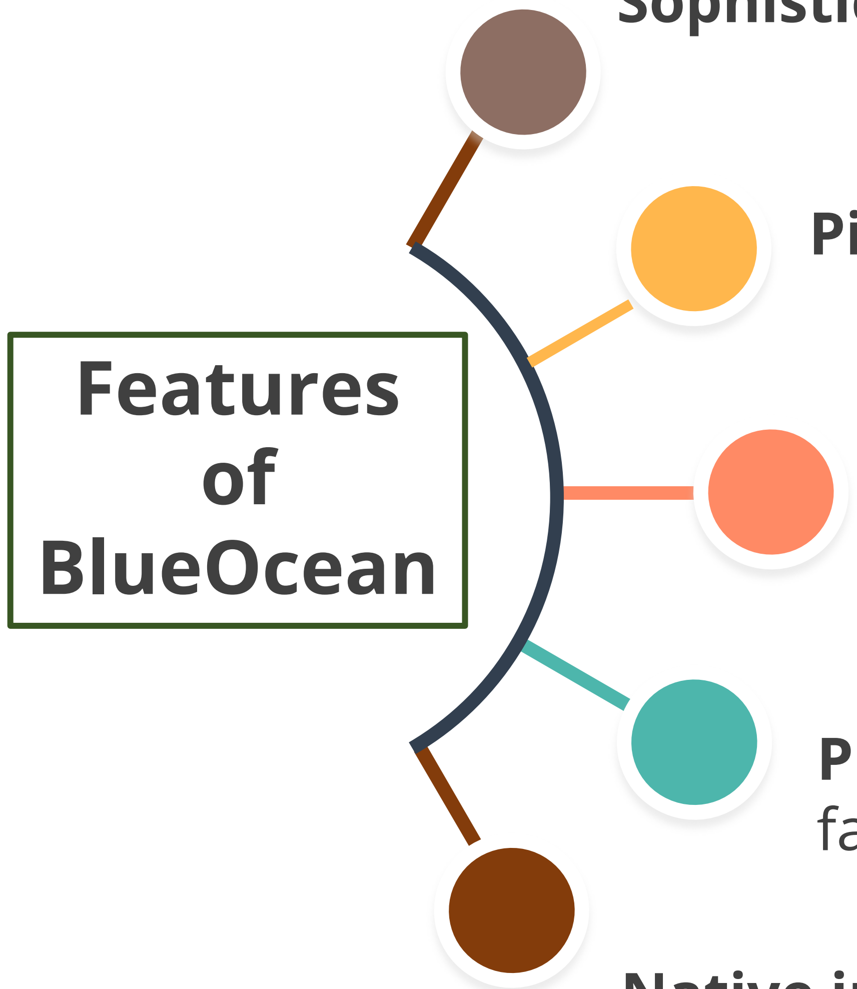
○ Project-based Matrix Authorization Strategy

Markup Formatter: Plain text ▾

Treats all input as plain text. HTML unsafe characters like < and & are escaped to their respective character entities.

Blue Ocean Strategy

Blue Ocean is an approach strategy and user experience enhancement in Jenkins that goes on-par with pipeline, which in itself is an optimized build strategy.



Features of BlueOcean

Sophisticated visualizations of Continuous Delivery (CD) pipelines for monitoring its status.

Pipeline editor enables creation of pipelines through visual process (UI).

Personalization allows to fulfill the needs of each member of the team.

Pinpoint precision indicates where the attention is required in the pipeline to facilitate exception handling.

Native integration for branch and pull requests enables maximum developer productivity when collaborating with GitHub and Bitbucket.

Blue Ocean Installation

Blue Ocean can be installed both on the existing Jenkins instance and as a part of Jenkins Docker.

Existing Jenkins instance

- To install the Blue Ocean suite of plugins on an existing Jenkins instance, your Jenkins instance must be running Jenkins 2.7.x or later.
- Plugins can be installed by any Jenkins user who has the **Administrator** access.
- You can install Blue Ocean plugin from **Manage Plugin** page.

Jenkins Docker

- Image has a tag for every release of blue ocean.
- To run the latest configurations, ensure to run **docker pull jenkinsci/blueocean**.
- To run Jenkins docker, use the command **docker run -p 8080:8080 jenkinsci/blueocean**
- Browse http://localhost:8080/blue to open the dashboard.

Assisted Practice
Build a Maven Project

Problem Statement: You are given a project to set up and run a Maven job in Jenkins.

Steps to perform:

1. Configure git on Jenkins.
2. Fork a sample repository.
3. Add the Maven plugin to Jenkins
4. Create a Jenkins build job of Maven type.
5. Build the Maven project.

Key Takeaways

- Jenkins is a middleware for the development cycle that connects different tools with one another and helps is smooth data flow.
- Git is one of the most popular tools for version control that helps in organizing code and executing testing operations.
- Jenkins provides more than thousand plugins to cover all the requirements of a CI/CD pipeline.

Knowledge Check

Knowledge Check 1

Which command is used to install GIT in an RPM-based distributed system?

A. sudo apt-get install git

B. sudo yum install git-core

C. Both

D. None of the above

Knowledge Check 1

Which command is used to install GIT in an RPM-based distributed system?

A. sudo apt-get install git

B. sudo yum install git-core

C. Both

D. None of the above

The correct answer is **B**

sudo yum install git-core is used in Fedora and other RPM-based distributed systems.

Knowledge Check 2

How to convert a folder into a Git repository?

A. Run git init

B. Run git initialise

C. Both

D. None of the above

Knowledge Check 2

How to convert a folder into a Git repository?

A. Run git init

B. Run git initialise

C. Both

D. None of the above

The correct answer is **A**

git init is the command to convert any folder into a Git repository.

Knowledge Check 3

Which of the following is NOT a feature of Jenkins?

A. Platform independent

B. Open-source

C. Scaling of large error-ridden integrations

D. None of the above

Knowledge Check 3

Which of the following is NOT a feature of Jenkins?

A. Platform independent

B. Open-source

C. Scaling of large error-ridden integrations

D. None of the above

The correct answer is **D**

All of the above mentioned features can be implemented in Jenkins.

Knowledge Check 4

How is Maven project added in Jenkins?

A. By adding the project folder in Jenkins storage

B. By adding the location of **pom.xml** in Maven settings

C. Both

D. None of the above

Knowledge Check
4

How is Maven project added in Jenkins?

A. By adding the project folder in Jenkins storage

B. By adding the location of **pom.xml** in Maven settings

C. Both

D. None of the above

The correct answer is **B**

Jenkins is capable of building the entire application with the help of pom.xml.

Knowledge Check 5

What is Jenkins?

A. Version control system

B. Middleware between development and production environment

C. Monitoring tool

D. Configuration management tool

Knowledge Check 5

What is Jenkins?

A. Version control system

B. Middleware between development and production environment

C. Monitoring tool

D. Configuration management tool

The correct answer is **B**

Jenkins is majorly used to integrate various tools and softwares included in SDLC to work in-sync for better flow of development.

Lesson-End Project

Building a Maven Project with Jenkins

Problem Statement:

You’re a DevOps engineer at PlanMyTrip, which is a web and mobile based app company that helps users plan their itineraries based on distance, climate, availability of transport, and accommodation. The company wants to use the OpenWeatherMap API for weather predictions. You’re required to write a microservice that checks if the API is up and set up a Jenkins build job to compile and package the app with Jenkins. The program and the build job is expected to be expanded in the future to add additional functionalities like redirection, notification, etc. You’re expected to create the program and build job to scale.

Requirements:

- The program should be built with Maven
- The build job should be designed to build a local repository
- The build job should compile and run the code if the build succeeds