**What are the ways to install Jenkins?**

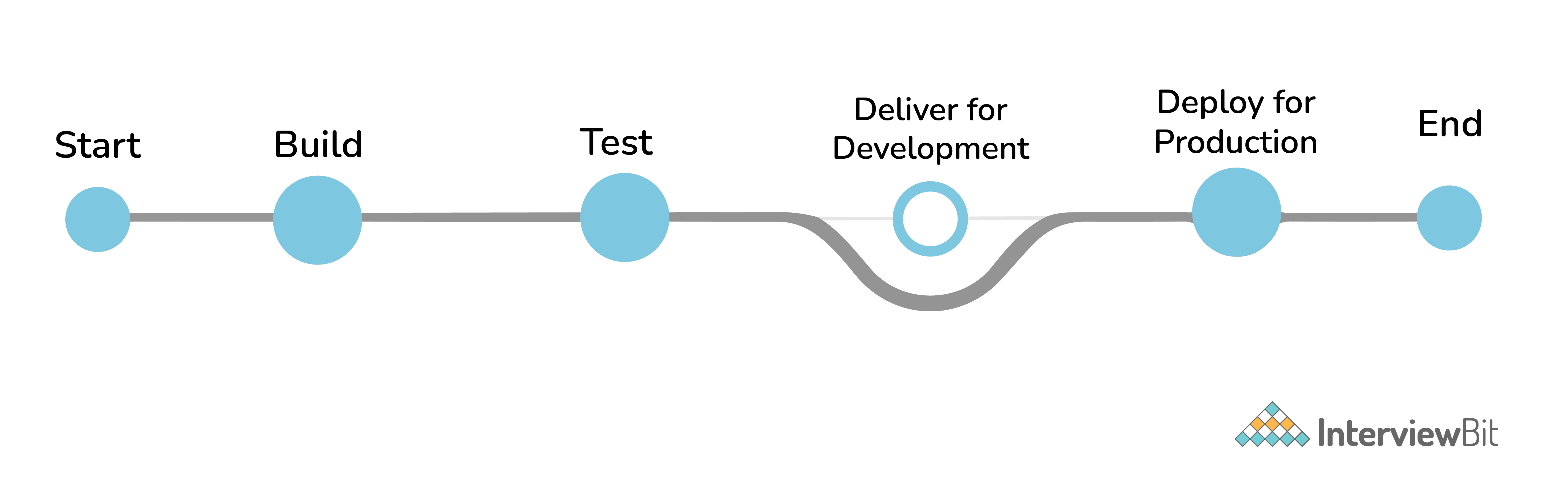
Jenkins can be installed using -

1. Native System Package Manager like - apt (Linux), brew (Mac), etc.
2. Docker (popular docker images for Jenkins is available for different platforms like Unix/Mac/Windows in the docker registry)
3. Kubernetes (available as a helm chart and can be installed on our Kubernetes clusters)
4. Standalone (on any machine with a Java Runtime Environment installed)

### What are the types of Jenkins pipelines?

Jenkins Pipelines can be either - a Declarative pipeline or a Scripted Pipeline. Declarative pipeline makes use of numerous, generic, predefined build steps/stages (i.e. code snippets) to build our job according to our build/automation needs whereas, with Scripted pipelines, the steps/stages can be custom-defined & used using a groovy syntax which provides better control & fine-tuned execution levels.

### 8. Explain Jenkins Multibranch Pipeline?

Jenkins Multibranch Pipeline

It is a pipeline job that can be configured to Create a set of Pipeline projects according to the detected branches in one SCM repository. This can be used to configure pipelines for all branches of a single repository e.g. if we maintain different branches (i.e. production code branches) for different configurations like locales, currencies, countries, etc.

### 9. How do you store credentials in Jenkins securely?

Credentials can be stored securely in Jenkins using the Credentials plugin, which stores different types of credentials like - Username with a password, SSH username with the private key, AWS Credentials, Jenkins Build Token, Secret File/Text, X509 & other certificates, Vault related credentials securely with proper encryption & decryption as and when required

### . How can we stop a scheduled job from being executed temporarily?

Disable the job from the job details page to temporarily stop all scheduled executions & other factors/events from triggering the job and enable it back to resume the job schedules/triggers. If a job is not required permanently, we can delete the job from the jobs list view page

**What are the ways to trigger a Jenkins Job/Pipeline?**

There are many ways we can trigger a job in Jenkins. Some of the common ways are as below -

* Trigger an API (POST) request to the target job URL with the required data.
* Trigger it manually from the Jenkins web application.
* Trigger it using Jenkins CLI from the master/slave nodes.
* Time-based Scheduled Triggers like a cron job.
* Event-based Triggers like SCM Actions (Git Commit, Pull Requests), WebHooks, etc.
* Upstream/Downstream triggers by other Jenkins jobs.

**What are the credential types supported by Jenkins?**

In Jenkins, credentials are a set of information used for authentication with internal/external services to accomplish an action. Jenkins credentials are provisioned & managed by a built-in plugin called - Credentials Binding - plugin. Jenkins can handle different credentials as follows -

* Secret text - A token such as an API token, JSON token, etc.
* Username and password - Basic Authentication can be stored as a credential as well.
* Secret file - A secret file used to authenticate some secure data services & security handshakes.
* SSH Username with a private key - An SSH public/private key pair for Machine to Machine authentication.
* Certificate - a PKCS#12 certificate file and an optional password.
* Docker Host Certificate Authentication credentials.

And as we can guess, this can be extended to several other extensible credential types like - AWS credential, Azure secrets, etc. using commonly available plugins

### What is a Jenkins Shared Library and how it is useful?

As an organization starts using more and more pipeline jobs, there is a chance for more and more code being duplicated in every pipeline job, since a part of the build/automation processes will be the same for most of the jobs. In such a situation, every other new upcoming job should also duplicate the same piece of code. To avoid duplications, the Jenkins project brought in the concept of Shared Libraries, to code - DRY - Don't Repeat Yourself.

Shared libraries are a set of code that can be common for more than one pipeline job and can be maintained separately. Such libraries improve the maintenance, modularity & readability of the pipeline code. And it also speeds up the automation for new jobs

**Jobs with parameters**

Simple example - sending "String Parameters":

curl JENKINS\_URL/job/JOB\_NAME/buildWithParameters  --user USER:TOKEN --data id=123 --data verbosity=high

### How to configure inclusions & exclusions in Artifacts Archival?

Artifact archival takes in a pattern for matching target files. Similarly, it also takes in a pattern (ANT build system pattern for matching files) for exclusion as well which will be ignored while selecting the files for archival.

For e.g.  
archiveArtifacts artifacts: 'output/\*.txt', excludes: 'output/specific\_file.txt'

The above command will archive all the text files from the output folder except specific\_file.txt

### How can we share information between different build steps or stages in a Jenkins Job?

Every build step or stage will be running in its process and hence sharing information between two different build steps is not so direct. We can use either a File, a Database Entry, an Environment Variable, etc. to share info from one build step to another or a post-build action.

### How code coverage is measured/tracked using Jenkins in a CI environment?

Using language-specific code coverage plugins like JaCoCo, CodeCov, etc or generic tools/plugins like Sonarqube which will add the code coverage data to builds with some minor tweaks in the code and the same can be displayed as a graph in Jenkins

### Default Environment Variables by Jenkins & How to introduce custom environment variables?

Jenkins provides several environment variables by default like - **BRANCH\_NAME, BUILD\_NUMBER, BUILD\_TAG, WORKSPACE, etc**

### How can a job configuration be reset to an earlier version/state?

From the Job details page, we can use Job Config History to - See diff, Review & Revert the Job configs from the history of changes we have made to a particular job. This will be super useful when a job is misconfigured by someone by mistake, it can be reviewed and reverted easily to any of its earlier states

**How to create & use a Shared Library in Jenkins?**

Basic requirements for a Jenkins shared library to be used in a Pipeline Code are -

* A Repository with pipeline shared library code in SCM.
* An appropriate SCM Plugin configuration for the Jenkins instance.
* Global Shared Library should be configured in Jenkins Global configuration.
* Include the Shared Library in the Pipeline Code and use the methods defined in the Jenkins Shared Library.

**Jenkins Remote Access API?**

Jenkins provides remote access API to most of its functionalities (though some functionalities are programming language-dependent). Currently, it comes in three flavors -

* XML
* JSON with JSONP support
* Python

Remote access API is offered in a REST-like style. That is, there is no single entry point for all features, and instead, they are available under the ".../api/" URL where the "..." portion is the data that it acts on.

### What is In-process Script Approval and how it works?

Jenkins, and several plugins, allow users to execute Groovy scripts in Jenkins. To protect Jenkins from the execution of malicious scripts, these plugins execute user-provided scripts in a Groovy Sandbox that limits what internal APIs are accessible.

This protection is provided by the Script Security plugin. As soon as an unsafe method is used in any of the scripts, the "In-process Script Approval" action should appear in "Manage Jenkins" to allow Administrators to make a decision about which unsafe methods, if any, should be allowed in the Jenkins environment.

This in-process script approval inherently improves the security of the overall Jenkins ecosystem.

### 35. Can we monitor Jenkins using common Observability tools?

Common monitoring platforms like **DataDog, Prometheus, JavaMelody** & few others - have their corresponding Jenkins plugin, which when configured, sends Metrics to the corresponding Monitoring platform, which can then be Observed with the latest tools & technologies. The same can be configured with Alarms & Notifications for immediate attention when something goes wron

### What is a Ping Thread in Jenkins and how it works?

Jenkins installs "ping thread" on every remote connection, such as Controller/Agent connections, regardless of its transport mechanism (such as SSH, JNLP, etc.). The lower level of the Jenkins Remoting Protocol is a message-oriented protocol, and a ping thread periodically sends a ping message that the receiving end will reply to. The ping thread measures the time it takes for the reply to arrive, and if it’s taking excessive time (currently 4 minutes and configurable), then it assumes that the connection was lost and initiates the formal close down.

This is to avoid an infinite hang, as some of the failure modes in the network cannot be detected otherwise. The timeout is also set to a long enough value so that a temporary surge in the load or a long garbage collection pause will not trip off the close-down.

Ping thread is installed on both controller & agent; each side pings the other and tries to detect the problem from their sides.

The ping thread time out is reported through java.util.logging. Besides, the controller will also report this exception in the agent launch log. Note that some agent launchers, most notably SSH agents, writes all stdout/stderr outputs from the agent JVM into this same log file, so you need to be careful.