

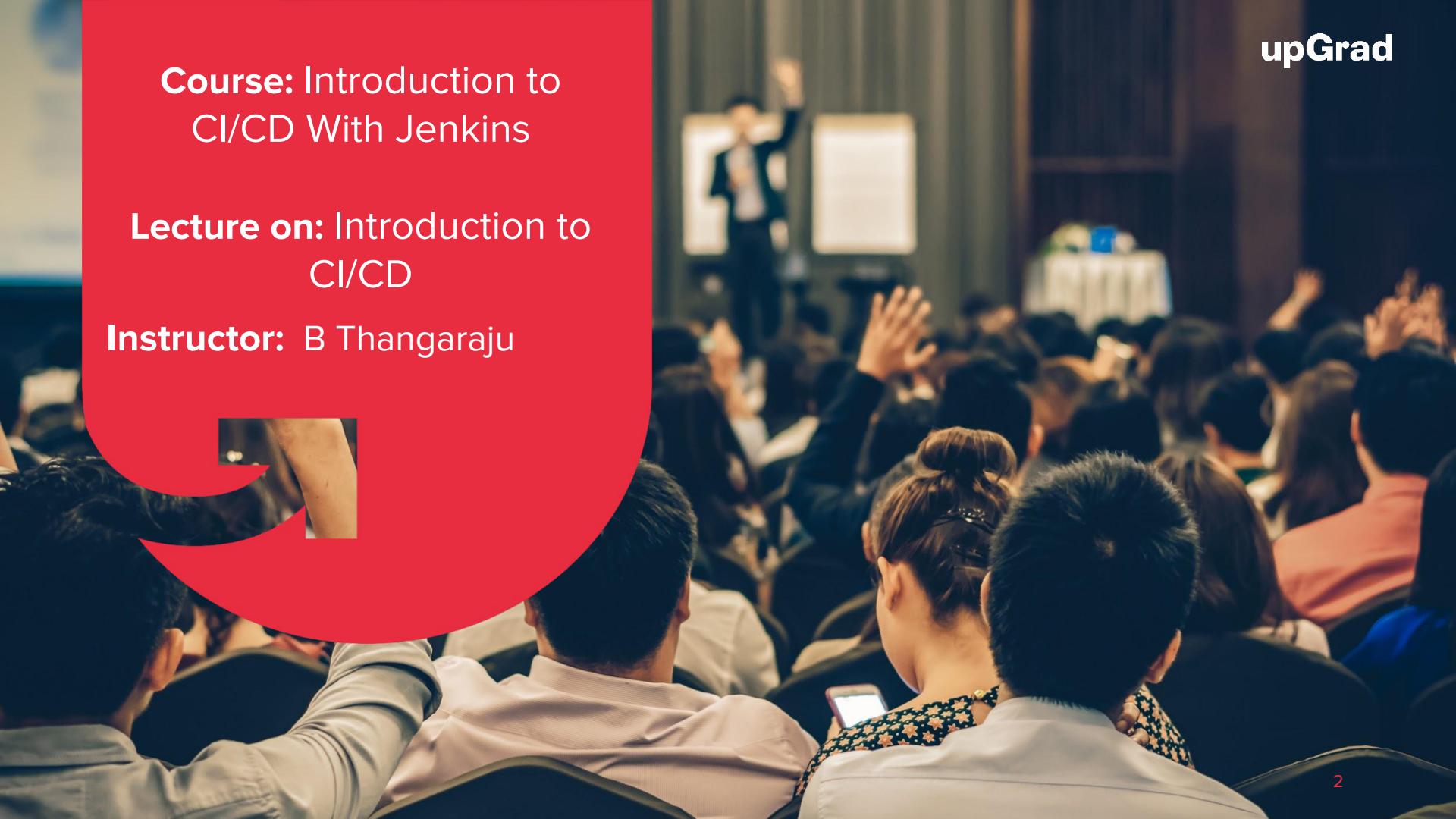


Introduction to CI/CD

Course: Introduction to
CI/CD With Jenkins

Lecture on: Introduction to
CI/CD

Instructor: B Thangaraju

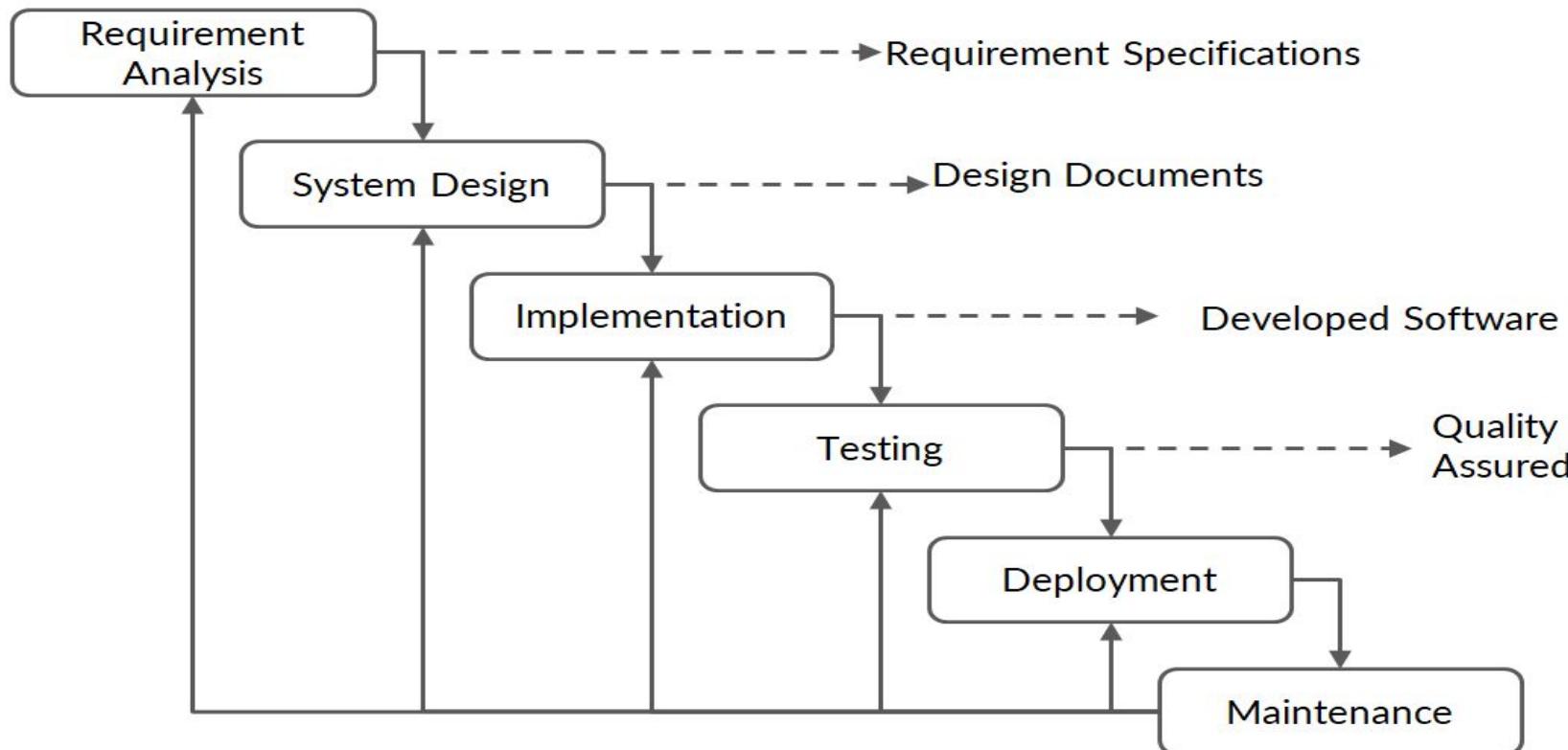


Today's Agenda

- Understanding the working of traditional SDLC approach
- How CI/CD works and solves the drawbacks of SDLC
- How Tesla leads market by using CI/CD
- Introduction to Jenkins as a CI tool
- Configuring and setting up Jenkins over AWS
- Creating Jobs in Jenkins and execute internally as well as remotely
- Sending email notifications to update status
- Parameterised jobs and their need

Introduction to CI/CD

Software Development Life Cycle



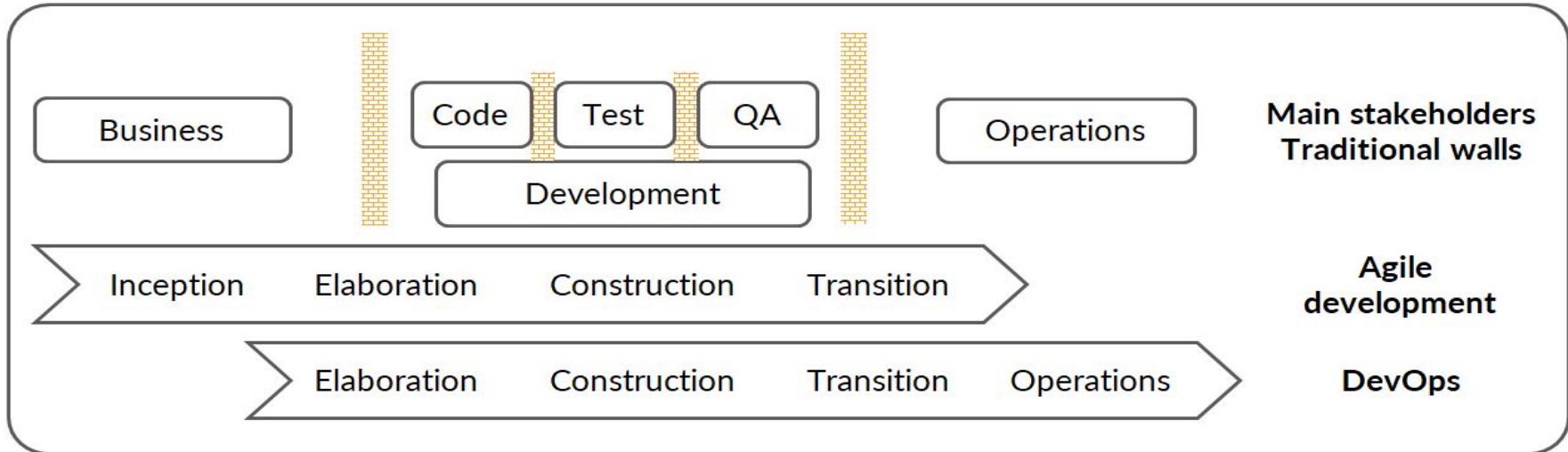
Introduction to CI/CD

- In software engineering, **continuous integration (CI)** is defined as a process of integrating all developers' code to a shared repository in a version-control system frequently.
- The term CI was first proposed by Grady Booch in 1991.
- CI is supposed to be used in conjunction with automated build, test, and QA.
- A build server compiles the code and reports the results to the developers on a regular basis, or even after each commit.
- Test: Unit test in the developer's local environment before committing to the mainline helps avoid one developer's work-in-progress while breaking another developer's copy.

Introduction to CI/CD

- Along with running the unit and integration checks, you can check the quality of your code and profile performance and format the documentation from the source code, thereby facilitating QA processes.
- This quality check application mainly aims to enhance software quality and reduce delivery time continuously.
- CI is connected closely with continuous delivery or continuous deployment, which is also called the CI/CD pipeline.
- ‘Continuous delivery’ ensures that the incremental feature of a software product checked in on the mainline is ready to deliver to the end user.
- ‘Continuous deployment’ makes the deployment process fully automated.

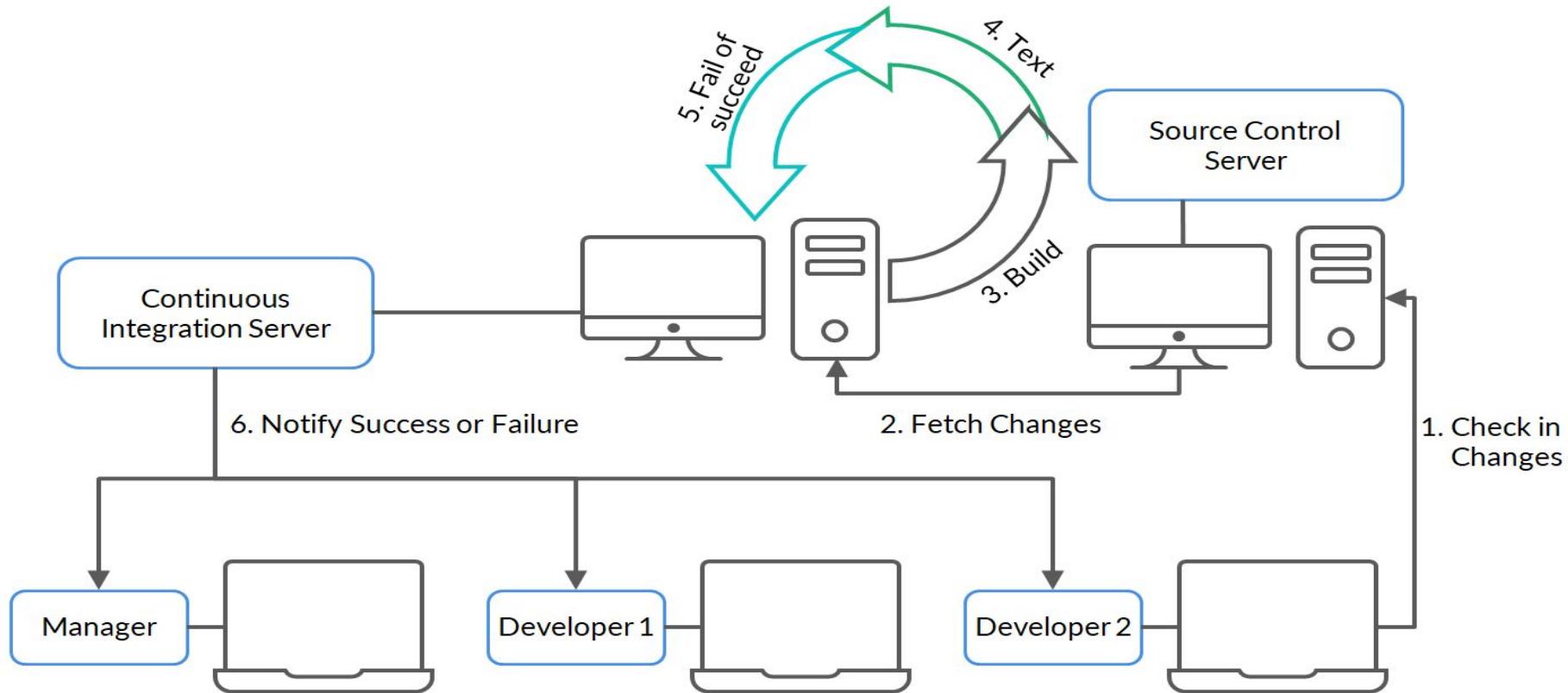
Drawbacks of Traditional SDLC



- Agile breaks the wall between Business and Development teams.
- DevOps breaks the wall between Development and Operations teams.
- DevOps centres on the concept of sharing (ideas, issues, processes, tools and goals).

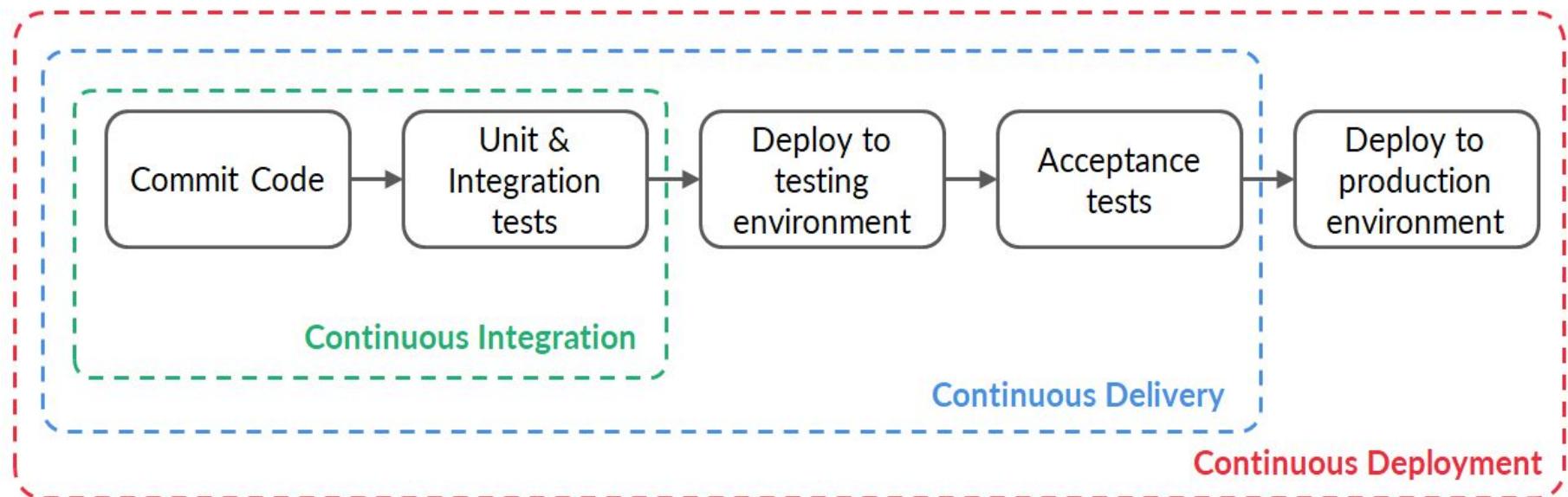
- CI, in its simplest form, involves a tool that continuously monitors version-control systems for any new changes.
- Whenever any new changes are pushed, this tool starts compiling and testing your application.
- If a bug is found or the code fails, the developers are immediately notified to fix the issue.
- If you integrate it with automated end-to-end acceptance checks, CI can serve as a feedback method, offering a straightforward image of the current state of development efforts.
- It will allow you to deploy the latest version of your application either automatically or as a one-click process.

CONTINUOUS INTEGRATION



Continuous – Integration, Delivery and Deployment

upGrad



Benefits of CI/CD

- Automation – ensures build, test, QA, delivery and deployment
- Improves consistencies and quality of code
- Delivers new features quickly to the end user
- Increases product visibility
- Helps avoid manual errors
- Helps fix any issues that may arise
- Reduces costs and labour

CI/CD Case Study

Problems that existed before CI/CD:

- Automobile companies made significant investment to create better hardware.
- They integrated software with various functionalities that a vehicle can offer to owners:
 - Braking system
 - Anti-lock system
 - Back-up collision systems
- Problem - This is just a one-time activity that takes place during the manufacture of cars.
- Later, it is difficult to update and manage these features.



Now, let's see how Tesla has tackled these issues by using the CI/CD flow in order to become the leading automobile company that it is today.

- Tesla cars are basically connected to labs from where their parts can be easily checked and upgraded.
- Any repairs or bug fixes can be done remotely without requiring the owner to visit the repair centre.
- The company is continuously working to modify the cars by adding new features on a regular basis.
- Therefore, the value of these cars is increasing, which is actually the opposite of the fact that the value of vehicles usually decreases with time.

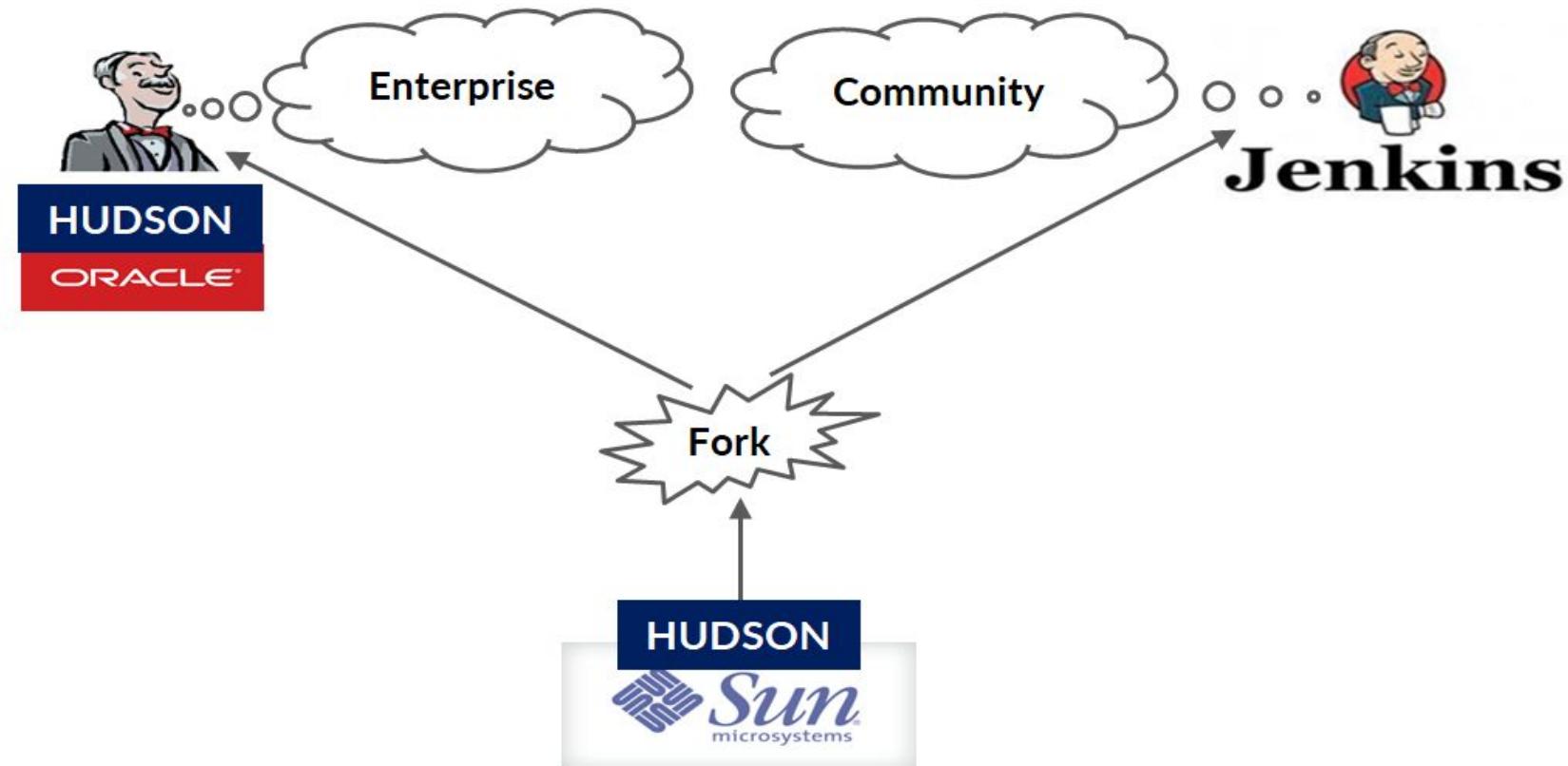
- This feedback system has helped Tesla in tackling some important problems.
- Years ago, several fire accidents were reported.
- Since then, the company has implemented CI/CD and initiated the continuous feedback mechanism, which helped them identify the problem.
- The problem was in the car's undercarriage as the chassis would collide with debris on roads.
- The company simply released an update, which helped to automatically fix this problem in all their vehicles.
- The company offers excellent service and delivery. These are some reasons for Tesla's dominant position in the market.

Introduction to Jenkins



- Jenkins is basically a free and open-source automation server.
- Jenkins can be used to automate all types of software development tasks such as building, testing, and delivering or deploying software.
- Jenkins can be installed via native system packages, Docker, or it can be run standalone on any computer.

History of Jenkins



- Jenkins is the outcome of an innovative developer, Kohsuke Kawaguchi, who began this project as a hobby under the name Hudson in late 2004 while working at Sun Microsystems.
- In 2009, Sun Microsystems was acquired by Oracle. At the end of 2010, tensions arose between the Hudson developer community and Oracle. Initially, the problem was with the Java.net infrastructure, which later worsened due to issues related to Oracle's claim to the Hudson trademark.
- In January 2011, the Hudson developer community decided to rename the project as Jenkins. Subsequently, they migrated the original Hudson code base to a new GitHub project¹ and continued their work there.
- After this, the majority of users joined the Jenkins developer community, migrating to Jenkins.

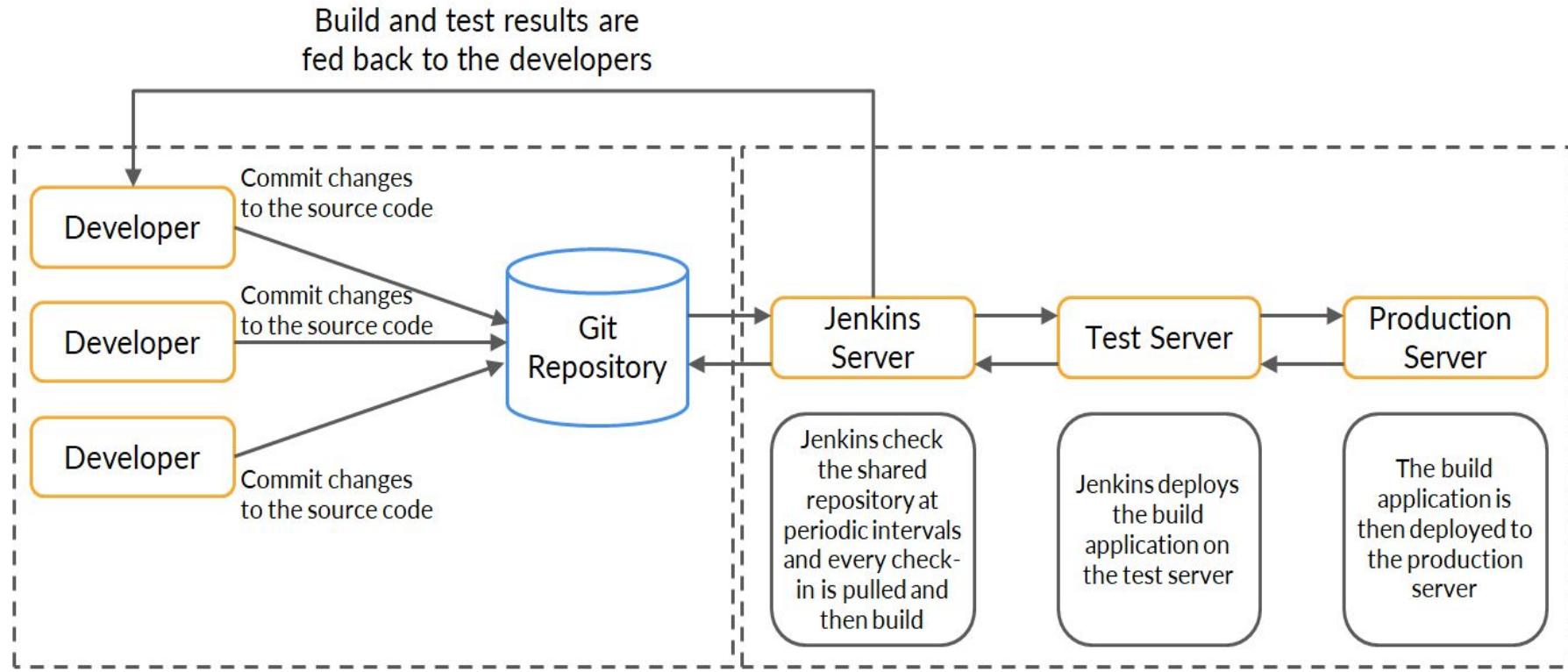
- Jenkins, originally called Hudson, is an open-source CI tool written in Java.
- Jenkins is a very user-friendly tool, with a simple, intuitive and visually appealing user interface.
- Jenkins has a very low learning curve.
- Jenkins is an extremely flexible tool and hundreds of open-source plugins are available, with more coming up every week.

- These plugins cover everything from version-control systems, build tools, code quality metrics and build notifiers to integration with external systems, UI customisations, and much more.
- The Jenkins community is a large, dynamic, reactive and welcoming bunch, with passionate champions, active mailing lists, IRC channels and a very vocal blog and Twitter account.
- The pace of development is quite fast, with the latest new features, bug fixes, and plugin updates being released on a weekly basis.
- Jenkins is a JAVA application, so it is platform independent.

Comparison of CI Tools

	Jenkins	TeamCity	Bamboo	Travis	Circle	Codeship
Pricing	Free	\$299-\$1999	\$10-\$800	\$69-\$489	\$50-\$3150	\$75-\$1500
Operating system	Windows, Linux, macOS, any Unix-like OS	Windows, Linux, macOS, Solaris, FreeBSD and more	Windows, Linux, macOS, Solaris	Linux, macOS	Linux, iOS, Android	Windows, macOS
Hosting	On premise/cloud	On premise	On premise/Bitbucket as cloud	On premise/cloud	Cloud	Cloud
Container support	✓	✓	✓	✓	✓	Yes for Pro version
Plugins	*****	****	**	***	***	****
Docs and support	Adequate	Good	Good	Poor	Good	Poor
Learning curve and usability	Easy	Medium	Medium	Easy	Easy	Easy
Use case	For big Projects	For enterprise needs	For Atlassian integrations	For small projects and startups	For fast development and high budget	For any project

Jenkins Architecture



Companies That Use Jenkins

Companies: 2836 companies reportedly use **Jenkins** in their tech stacks, including Facebook, Netflix, and Udemy

Developers: 33308 developers on StackShare have stated that they use **Jenkins**.



Facebook



Netflix



Udemy



Instacart



Robinhood



Twitch



Lyft



Delivery Hero



LinkedIn

Slack, Datadog, BrowserStack, Azure DevOps, and SonarQube are some of the popular 144 tools that integrate with Jenkins.



Slack



Datadog



BrowserStack



Azure DevOps



SonarQube



Rancher



AWS CodePipeline



Mattermost



AWS CodeCommit

Configuring and Setting Up Jenkins

Install Java

```
sudo apt install openjdk-11-jdk
```

To check java –version

```
openjdk 11.0.10 2021-01-19
```

```
OpenJDK Runtime Environment (build 11.0.10+9-Ubuntu-0ubuntu1.18.04)
```

```
OpenJDK 64-Bit Server VM (build 11.0.10+9-Ubuntu-0ubuntu1.18.04, mixed mode, sharing)
```

Install Jenkins

```
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'
```

```
sudo apt-get update
```

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

To check Jenkins version: vim /var/lib/jenkins/config.xml

```
<version>2.263.4</version>
```



EC2 > Instances > i-0e5f0719976cef9bd

Instance summary for i-0e5f0719976cef9bd (Thangaraju) [Info](#)

Updated less than a minute ago

Instance ID	Public IPv4 address
 i-0e5f0719976cef9bd (Thangaraju)	 3.95.168.69 open address 
Instance state	Public IPv4 DNS
 Running	 ec2-3-95-168-69.compute-1.amazonaws.com open address 

The screenshot shows a web browser window with the URL `3.95.168.69:8080/login?from=%2F` in the address bar. The page title is "Getting Started". The main content is titled "Unlock Jenkins" with a sub-instruction: "To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:". Below this, a red-highlighted path is shown: `/var/lib/jenkins/secrets/initialAdminPassword`. A note says "Please copy the password from either location and paste it below." A text input field is labeled "Administrator password". A yellow callout box contains the command `sudo cat /var/lib/jenkins/secrets/initialAdminPassword` followed by the password `5bd8dce3437a4a80886ba00389e2d16f`, with the instruction "Copy and paste as Administrator password". A "Continue" button is at the bottom right.

← → C Not secure | 3.95.168.69:8080/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
5bd8dce3437a4a80886ba00389e2d16f
Copy and paste as Administrator password
```

Continue

Getting Started



Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Getting Started

Suggested Plugins				++ SSM Credentials Credentials Binding ** SCM API ** Pipeline: API Timestamper ** Script Security ** Plugin Utilities API ** Font Awesome API ** Popper.js API ** JQuery3 API ** Bootstrap 4 API ** Snakeyaml API ** Jackson 2 API ** ECharts API ** Display URL API ** Pipeline: Supporting APIs ** Pipeline: Job ** Checks API ** JUnit ** Matrix Project ** - required dependency
✓ Timestamper	⌚ Workspace Cleanup	⌚ Ant	⌚ Gradle	
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline: Stage View	
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ PAM Authentication	
⌚ LDAP	⌚ Email Extension	⌚ Mailer		

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.263.4

[Skip and continue as admin](#)

[Save and Continue](#)

[Create User Accounts Later](#)

Instance Configuration

Jenkins URL:

<http://3.95.168.69:8080/>

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Getting Started

Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

[Start using Jenkins](#)

The screenshot shows the Jenkins dashboard interface. At the top, there's a navigation bar with the Jenkins logo, a search bar, and user account information for 'admin'. Below the header, the left sidebar contains links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. Under 'Build Queue' and 'Build Executor Status', there are sections indicating 'No builds in the queue.' and '1 Idle' and '2 Idle' executors respectively. The main content area features a large 'Welcome to Jenkins!' message, a brief description of the page's purpose, and two main call-to-action buttons: 'Start building your software project' and 'Set up a distributed build'. Each button has a corresponding link: 'Create a job', 'Set up an agent', and 'Configure a cloud'. There's also a link to 'Learn more about distributed builds'.

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

search

admin

log out

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

Set up a distributed build

Set up an agent →

Configure a cloud →

Learn more about distributed builds →

The screenshot shows the Jenkins dashboard interface. At the top left is the Jenkins logo, which is a cartoon character wearing a hard hat and holding a wrench. To the right of the logo, the word "Jenkins" is written in a large, bold, white sans-serif font. Below the header is a navigation bar with the word "Dashboard" and a right-pointing arrow. The main content area contains eight items, each with an icon and text:

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Lockable Resources
- New View

Enter an item name

Project1

* Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



GitHub Organization

Scans a GitHub organization (or user account) for all repositories matching some defined markers.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

Jenkins

Dashboard > All >

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

New View

Build Queue

No builds in the queue.

Build Executor Status

Icon: S M L

Legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

Build History of Jenkins

This history is not guaranteed to include all subtasks executed on the node, e.g. Jenkins Pipeline subtasks will not be displayed.

Mar 5	Mar 6	Mar 7	Mar 8	Mar 9	Mar 10	Mar 11
7hr	8hr	9hr	10hr	11hr	12hr	

Jenkins

Dashboard >

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Lockable Resources
- New View

Build Queue ^

No builds in the queue.

Build Executor Status ^

1 idle

Manage Jenkins

System Configuration

- Configure System**
Configure global settings and paths.
- Global Tool Configuration**
Configure tools, their locations and automatic installers.
- Manage Nodes and Clouds**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

- Configure Global Security**
Secure Jenkins; define who is allowed to access/use the system.
- Manage Credentials**
Configure credentials.
- Configure Credential Providers**
Configure the credential providers and types.

Manage Users

Create/delete/modify users that can log in to this Jenkins

2 Idle

Status Information

**System Information**
Displays various environmental information to assist trouble-shooting.

**System Log**
System log captures output from `java.util.logging` output related to Jenkins.

**Load Statistics**
Check your resource utilization and see if you need more computers for your builds.

Troubleshooting

**Manage Old Data**
Scrub configuration files to remove remnants from old plugins and earlier versions.

Tools and Actions

**Reload Configuration from Disk**
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

**Jenkins CLI**
Access/manage Jenkins from your shell, or from your script.

**Script Console**
Executes arbitrary script for administration/troubleshooting/diagnostics.

**Prepare for Shutdown**
Stops executing new builds, so that the system can be eventually shut down safely.

Jenkins Plugin Manager – Updates

upGrad

The screenshot shows the Jenkins Plugin Manager - Updates page. The left sidebar includes links for Back to Dashboard, Manage Jenkins, and Update Center. The main area has tabs for Updates, Available, Installed, and Advanced, with Updates selected. A search bar is at the top right. The table lists two plugins:

Name	Version	Released	Installed
Branch API	2.6.3	2.6.2	
GitHub Branch Source	2.10.2	2.9.7	

For the 'Branch API' plugin, it says 'UNAVAILABLE' and provides a description: 'This plugin provides an API for multiple branch based projects.' A note indicates it's not offered as an update because Jenkins requirements aren't met. For the 'GitHub Branch Source' plugin, it also says 'UNAVAILABLE' and provides a description: 'Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.' A similar note about update availability is present.

Jenkins Plugin Manager – Available

upGrad

The screenshot shows the Jenkins Plugin Manager interface. The top navigation bar includes links for Dashboard, Manage Jenkins, and Update Center, along with a search bar containing 'gitLab' and user account information for 'admin'. The main content area displays a list of available Jenkins plugins:

Install	Name	Version	Released
<input type="checkbox"/>	GitLab Build Triggers	1.5.19	5 days 22 hr ago
<p>This plugin allows GitLab to trigger Jenkins builds and display their results in the GitLab UI.</p> <p>This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.</p>			
<input type="checkbox"/>	Generic Webhook Trigger bitbucket bitbucket-server github gitlab jira notification Build Parameters Build Triggers webhook	1.72	4 mo 1 day ago
<p>Can receive any HTTP request, extract any values from JSON or XML and trigger a job with those values available as variables. Works with GitHub, GitLab, Bitbucket, Jira and many more.</p>			
<input type="checkbox"/>	Gitlab Hook Source Code Management related		
<p>Enables Gitlab web hooks to be used to trigger SMC polling on Gitlab projects</p>			

Jenkins Plugin Manager – Installed

upGrad

 Jenkins

Dashboard > Plugin Manager

Back to Dashboard | Manage Jenkins | Update Center

filter

Updates Available **Installed** Advanced

Enabled	Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Ant Plugin Adds Apache Ant support to Jenkins	1.11		<button>Uninstall</button>
<input checked="" type="checkbox"/>	Apache HttpComponents Client 4.x API Plugin Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins. This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	4.5.13-1.0		<button>Uninstall</button>
<input checked="" type="checkbox"/>	Bootstrap 4 API Plugin Provides Bootstrap 4 for Jenkins plugins.	4.6.0-2		<button>Uninstall</button>
<input checked="" type="checkbox"/>	bouncycastle API Plugin This plugin provides an stable API to Bouncy Castle related tasks.	2.20		<button>Uninstall</button>

The screenshot shows the Jenkins Plugin Manager - Advanced page. At the top, there is a navigation bar with links for Dashboard, Plugin Manager, Back to Dashboard, Manage Jenkins, and Update Center. The "Advanced" tab is selected in the top navigation bar. The main content area is divided into several sections:

- HTTP Proxy Configuration:** This section contains fields for Server, Port, User name, Password, and No Proxy Host. Each field has a question mark icon to its right for help.
- Upload Plugin:** A section for uploading a plugin from outside the central repository. It includes a "Choose File" button and an "Upload" button.
- Update Site:** A section for specifying the URL of the update site. The URL field contains "https://updates.jenkins.io/update-center.json".

Poll 1 (15 seconds)

Jenkins default admin password can be found at:

- A. /var/lib/jenkins/secrets/InitialAdminPassword
- B. /etc/passwd | grep admin
- C. /var/lib/Jenkins/secrets/InitialAdminPassword
- D. /var/lib/Jenkins/secrets/passwd

Poll 1 (15 seconds)

Jenkins default admin password can be found at:

- A. **/var/lib/jenkins/secrets/InitialAdminPassword**
- B. /etc/passwd | grep admin
- C. /var/lib/Jenkins/secrets/InitialAdminPassword
- D. /var/lib/Jenkins/secrets/passwd

Understand How to Create Jobs in Jenkins

```
ubuntu@ip-172-31-81-117:~$ sudo service jenkins start
ubuntu@ip-172-31-81-117:~$ sudo service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
  Loaded: loaded (/etc/init.d/jenkins; generated)
  Active: active (exited) since Tue 2021-03-09 01:13:37 UTC; 2min 52s ago
    Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 1140)
   CGroup: /system.slice/jenkins.service

Mar 09 01:13:34 ip-172-31-81-117 systemd[1]: Starting LSB: Start Jenkins at boot time...
Mar 09 01:13:35 ip-172-31-81-117 jenkins[842]: Correct java version found
Mar 09 01:13:35 ip-172-31-81-117 jenkins[842]: * Starting Jenkins Automation Server jenkins
Mar 09 01:13:35 ip-172-31-81-117 su[943]: Successful su for jenkins by root
Mar 09 01:13:35 ip-172-31-81-117 su[943]: + ??? root:jenkins
Mar 09 01:13:36 ip-172-31-81-117 su[943]: pam_unix(su:session): session opened for user jenkins by (uid=0)
Mar 09 01:13:36 ip-172-31-81-117 su[943]: pam_unix(su:session): session closed for user jenkins
Mar 09 01:13:37 ip-172-31-81-117 jenkins[842]: ...done.
Mar 09 01:13:37 ip-172-31-81-117 systemd[1]: Started LSB: Start Jenkins at boot time.
ubuntu@ip-172-31-81-117:~$ █
```

1. Know the status of **Jenkins**: `sudo service jenkins status`.
2. To **start Jenkins**: `sudo service jenkins start`.
3. To **stop Jenkins**: `sudo service jenkins stop`.
4. To **restart Jenkins**: `sudo service jenkins restart`. (stop and start)

Jenkins home directory:

```
ubuntu@ip-172-31-81-117:~$ sudo su - jenkins  
jenkins@ip-172-31-81-117:~$ pwd  
/var/lib/jenkins  
jenkins@ip-172-31-81-117:~$
```

Jenkins home directory: /var/lib/jenkins

It contains the details of the Jenkins server configuration, which is used to configure in the Manage Jenkins. The core configuration files are stored in config.xml.

It contains two important subdirectories: **jobs and workspace**.

Jobs directory contains configuration details of the build job in config.xml file.

Jenkins builds the given project in the workspace directory. Each project has its own directory in the workspace.

The screenshot shows the Jenkins dashboard. On the left sidebar, there are several navigation links: 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'Lockable Resources'. The main content area features a large 'Welcome to Jenkins!' heading. Below it, a text block says: 'This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.' At the bottom of this section is a button labeled 'Start building your software project'. A prominent red arrow points from the top-left towards this button. To the right of the button is another button labeled 'Create a job'.

Enter an item name

HelloWorld Python Program

» Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

Example: HelloWorld Python Program

upGrad

The screenshot shows the Jenkins dashboard. On the left sidebar, there are links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'. The main area displays a single job entry:

All	W	Name ↓	Last Success	Last Failure	Last Duration
		HelloWorld Python Program	N/A	N/A	N/A

Below the table, there are icons for 'Icon: S M L', 'Legend', and three 'Atom feed' options: 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. In the top right corner, there is a search bar, a help icon, a user icon labeled 'admin', and a 'log out' button.

```
#!/usr/bin/python3
# This Phyton program will print Hellow World...
print("\nHellow World...\n")
```

The screenshot shows the Jenkins interface for the "HelloWorld Python Program" project. The left sidebar contains links: "Back to Dashboard", "Status" (which is selected), "Changes", "Workspace", "Build Now", and "Configure". A red arrow points from the "Configure" link towards the "Recent Changes" section on the right. The main content area has a title "Project HelloWorld Python Program" and a subtitle "HelloWorld Python Script". It also includes sections for "Workspace" and "Recent Changes".

Jenkins

Dashboard > HelloWorld Python Program >

Back to Dashboard

Status

Changes

Workspace

Build Now

Configure

Recent Changes

Workspace

Project HelloWorld Python Program

HelloWorld Python Script

Jenkins

Dashboard > HelloWorld Python Program > #15

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Console Output

Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/HelloWorld Python Program
[HelloWorld Python Program] \$ /bin/sh -xe /tmp/jenkins9071003953682465907.sh
+ ./HelloWorld.py

Hello World...

Finished: SUCCESS

Check Log Files

```
ubuntu@ip-172-31-81-117:/var/lib/jenkins/jobs/HelloWorld Python Program/builds$ ls
1 10 2 3 4 5 6 7 8 9 legacyIds permalinks
ubuntu@ip-172-31-81-117:/var/lib/jenkins/jobs/HelloWorld Python Program/builds$ cd 4
ubuntu@ip-172-31-81-117:/var/lib/jenkins/jobs/HelloWorld Python Program/builds/4$ ls
build.xml changelog.xml log
ubuntu@ip-172-31-81-117:/var/lib/jenkins/jobs/HelloWorld Python Program/builds/4$ cat log
Started by user ha://401Xwh1EUGVcmahxd0uDY/VpeIbwK+Jgej0qfQuwtGn1AAAA1x+LCAAAAAAAAP9b85aBtbiIQTGjNKU4P08vOT+v
1x6OyILUoJzMv2y+/JJUBAhiZGBgqihhk0NSjKDwzXb3Rd1LBUSYGJk8GtpzUvPSSDB8G5tKinBIGIZ+sxLJE/ZzEvHT94JKizLx0a6BxUmjGOU
gZu/dLi1CL9xJTczDwAj6GcLcAAAAA=admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/HelloWorld Python Program
[HelloWorld Python Program] $ /bin/sh -xe /tmp/jenkins10839759464246091234.sh
+ echo Hello World
Hello World
Finished: SUCCESS
ubuntu@ip-172-31-81-117:/var/lib/jenkins/jobs/HelloWorld Python Program/builds/4$
```

Custom Workspace

upGrad

Dashboard > HelloWorld Python Program >

General	Source Code Management	Build Triggers	Build Environment	Build	Post-build Actions
<input type="checkbox"/> Execute concurrent builds if necessary					?
<input type="checkbox"/> Quiet period					?
<input type="checkbox"/> Retry Count					?
<input type="checkbox"/> Block build when upstream project is building					?
<input type="checkbox"/> Block build when downstream project is building					?
<input checked="" type="checkbox"/> Use custom workspace					?
Directory	/home/ubuntu				
Display Name					?

```
ubuntu@ip-172-31-81-117:/var/lib/jenkins/jobs/HelloWorld Python Program/builds$ ls
1 10 2 3 4 5 6 7 8 9 legacyIds permalinks
ubuntu@ip-172-31-81-117:/var/lib/jenkins/jobs/HelloWorld Python Program/builds$ cd 9
ubuntu@ip-172-31-81-117:/var/lib/jenkins/jobs/HelloWorld Python Program/builds/9$ cat log
Started by user ha://401Xwh1EUGVcmahxd0uDY/VpeIbwK+Jgej0qfQuwtGn1AAAAlx+LCAAAAAAAAP9b85aBtbiIQTGj
1x6OyILUoJzMv2y+/JJUBAhizZGBggihhk0NSjKDwzXb3RdlLBUSYGJk8GtpzUvPSSDB8G5tKinBIGIZ+sxEJ/EzzEvHT94JKiZl
gZu/dLi1CL9xJTczDwAj6GcLcAAAAA=admin
Running as SYSTEM
Building in workspace /home/ubuntu
[ubuntu] $ /bin/sh -xe /tmp/jenkins2121288571684403099.sh
+ ./HelloWorld.py

Hello World...

Finished: SUCCESS
ubuntu@ip-172-31-81-117:/var/lib/jenkins/jobs/HelloWorld Python Program/builds/9$ █
```

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#13'

Previous Build

Console Output

Started by user admin
Running as SYSTEM
Building in workspace /home/ubuntu
[ubuntu] \$ /bin/sh -xe /tmp/jenkins14930569387565051667.sh
+ ./HelloWorld.py

Hellow World...

Finished: SUCCESS

Build

Execute shell

Command

```
cd ../../  
echo The total number of directories in the $PWD  
ls -Rl | grep ^d | wc -l
```



Console Output

Started by user admin

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/test1

```
[test1] $ /bin/sh -xe /tmp/jenkins17988033407760365594.sh  
+ cd ../../  
+ echo The total number of directories in the /var/lib/jenkins  
The total number of directories in the /var/lib/jenkins  
+ wc -l  
+ grep ^d  
+ ls -Rl  
676  
Finished: SUCCESS
```

Check Log Files in test1

upGrad

```
jenkins@ip-172-31-81-117:~/jobs/test1/builds$ ls
1 2 3 legacyIds permalinks
jenkins@ip-172-31-81-117:~/jobs/test1/builds$ cd 3
jenkins@ip-172-31-81-117:~/jobs/test1/builds/3$ ls
build.xml changelog.xml log
jenkins@ip-172-31-81-117:~/jobs/test1/builds/3$ cat log
Started by user ha:///401Xwh1EUGVcmahxd0uDY/VpeIbwK+Jgej0qfQuwtGn1AAJ
1x6OyILUoJzMv2y+/JJUBAhizGBgqihhk0NSjKDwzXb3Rd1LBUSYGJk8GtpzUvPSSDB8G!
gZu/dLi1CL9xJTczDwAj6GcLcAAAAA=admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/test1
[test1] $ /bin/sh -xe /tmp/jenkins17988033407760365594.sh
+ cd ../../..
+ echo The total number of directories in the /var/lib/jenkins
The total number of directories in the /var/lib/jenkins
+ wc -l
+ grep ^d
+ ls -Rl
676
Finished: SUCCESS
jenkins@ip-172-31-81-117:~/jobs/test1/builds/3$
```

```
<project>
  <actions/>
  <description></description>
  <keepDependencies>false</keepDependencies>
  <properties/>
  <scm class="hudson.scm.NullSCM"/>
  <canRoam>true</canRoam>
  <disabled>false</disabled>
  <blockBuildWhenDownstreamBuilding>false</blockBuildWhenDownstreamBuilding>
  <blockBuildWhenUpstreamBuilding>false</blockBuildWhenUpstreamBuilding>
  <triggers/>
  <concurrentBuild>false</concurrentBuild>
  <builders>
    <hudson.tasks.Shell>
      <command>cd ../../
echo The total number of directories in the $PWD
ls -Rl | grep ^d | wc -l
</command>
      <configuredLocalRules/>
    </hudson.tasks.Shell>
  </builders>
  <publishers/>
  <buildWrappers/>
</project>
```

Poll 2 (15 seconds)

The sudo service jenkins restart command will:

- A. Start the Jenkins service
- B. Stop the Jenkins service
- C. First, it will stop if it is running, and then, it will start the service
- D. Give the status of the Jenkins service

Poll 2 (15 seconds)

The sudo service jenkins restart command will:

- A. Start the Jenkins service
- B. Stop the Jenkins service
- C. First, it will stop if it is running, and then, it will start the service**
- D. Give the status of the Jenkins service

Step 1: Choose the project.

Step 2: Configure – Build – enable Trigger builds remotely.

Step 3: Modify: JENKINS_URL/job/test1/build?token=TOKEN_NAME

<http://3.95.7.132:8080/job/test1/build?token=86470>.

Step 4: Open a browser and enter: <http://3.95.7.132:8080/job/test1/build?token=86470>

Step 5: Verify the build number.

Build Triggers

Trigger builds remotely (e.g., from scripts) ?

Authentication Token

Use the following URL to trigger build remotely: JENKINS_URL/job/test1/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME

Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

Build after other projects are built ?

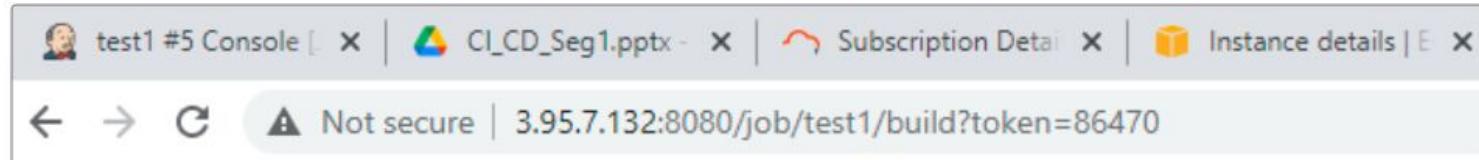
Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Save **Apply**

Open a browser, type the URL and enter



Check Build History in the Jenkins Server

A screenshot of the Jenkins 'Build History' page for a job named 'test1'. The page title is 'Build History'. A search bar contains the text 'find'. Below it, build #5 is listed with the status '(pending—In the quiet period. Expires in 1.9 sec)'. A red border surrounds the entire screenshot area.

Manual Build

Trigger Build Remotely



Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/test1
[test1] $ /bin/sh -xe /tmp/jenkins17988033407760365594.sh
+ cd ../..
+ echo The total number of directories in the /var/lib/jenkins
The total number of directories in the /var/lib/jenkins
+ wc -l
+ grep ^d
+ ls -Rl
676
Finished: SUCCESS
```



Console Output

```
Started by remote host 122.181.192.30
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/test1
[test1] $ /bin/sh -xe /tmp/jenkins1040342982316183531.sh
+ cd ../..
+ echo The total number of directories in the /var/lib/jenkins
The total number of directories in the /var/lib/jenkins
+ wc -l
+ grep ^d
+ ls -Rl
678
Finished: SUCCESS
```

- Create 3 Jobs:
 - Job1: CPU INFORMATION
 - Job2: RAM INFORMATION
 - Job3: DISK INFORMATION
- Job2 will start after Job1 build is successful
- Job3 will start after job2 build is successful

Job Chaining in Jenkins – Job1 Configuration

upGrad

The screenshot shows the 'Source Code Management' tab selected in the Jenkins job configuration. Under the 'Source Code Management' section, the 'None' radio button is selected. Below this, there is a 'Build Triggers' section containing several checkboxes:

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

The screenshot shows the 'Build' tab selected in the Jenkins job configuration. Under the 'Build' section, there is a single 'Execute shell' step. The 'Command' field contains the following Jenkinsfile code:

```
echo "CPU INFORMATION"  
lscpu
```

The screenshot shows the configuration page for 'RAM INFORMATION' under the 'Build Triggers' tab. It includes sections for 'Trigger builds remotely', 'Build after other projects are built' (checked), and 'Projects to watch' (set to 'CPU INFORMATION'). Below this, three build trigger options are listed: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. In the 'Build' section, there is one step named 'Execute shell' with the command 'echo "Physical Memory Information"\nfree -m'.

Dashboard > RAM INFORMATION >

General Source Code Management **Build Triggers** Build Environment Build Post-build Actions

Build Triggers

Trigger builds remotely (e.g., from scripts) ?
 Build after other projects are built ?
Projects to watch: CPU INFORMATION

Trigger only if build is stable
 Trigger even if the build is unstable
 Trigger even if the build fails

Build

Execute shell

Command: echo "Physical Memory Information"
free -m

Dashboard > DISK INFORMATION >

General Source Code Management Build Triggers

Build Triggers

Trigger builds remotely (e.g., from scripts)

Build after other projects are built

Projects to watch RAM INFORMATION,

Trigger only if build is stable

Trigger even if the build is unstable

Trigger even if the build fails

Build periodically

GitHub hook trigger for GITScm polling

Poll SCM

Dashboard > DISK INFORMATION >

General Source Code Management Build Triggers

Build

Execute shell

Command

```
echo "DISK INFORMATION"  
df -h
```

Build Job1 – CPU INFORMATION

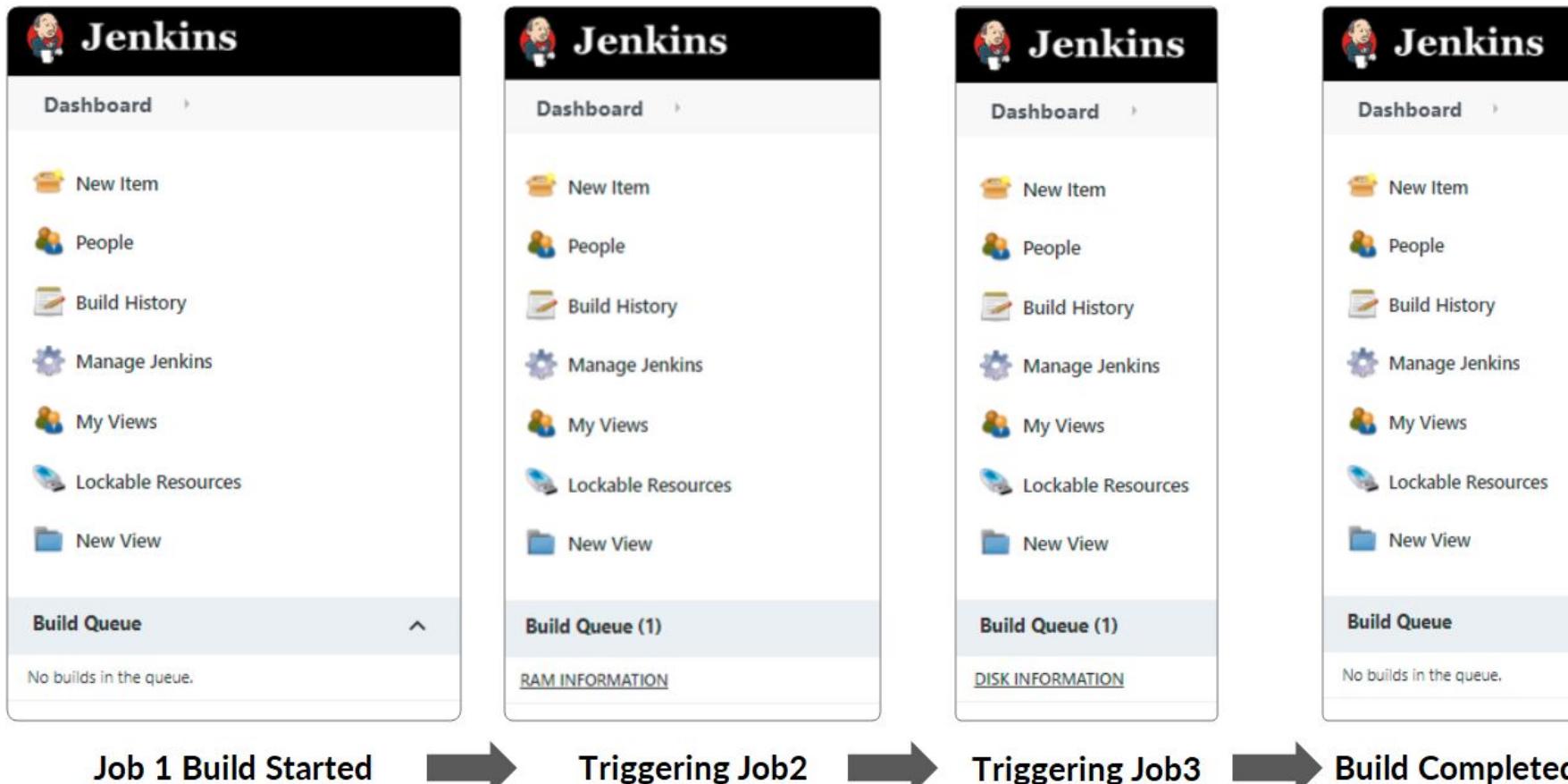
upGrad

S	W	Name ↓
		CPU INFORMATION
		DISK INFORMATION
		HelloWorld Python Program
		RAM INFORMATION
		test1

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		CPU INFORMATION	20 min - #1	N/A	18 ms	
			20 min - #1	N/A	31 ms	
		ram	2 hr 6 min - #15	4 hr 4 min - #8	43 ms	
			20 min - #1	N/A	26 ms	

Monitor Job Chaining Builds

upGrad



Dashboard > CPU INFORMATION > #1

 **Console Output**

Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/CPU INFORMATION
[CPU INFORMATION] \$ /bin/sh -xe /tmp/jenkins4789611751145793460.sh
+ echo CPU INFORMATION
CPU INFORMATION
+ lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 1
On-line CPU(s) list: 0
Thread(s) per core: 1

invpcid xsaveopt
Triggering a new build of RAM INFORMATION
Finished: SUCCESS

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'



Console Output

Started by upstream project "CPU INFORMATION" build number 1
originally caused by:

Started by user admin

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/RAM INFORMATION

[RAM INFORMATION] \$ /bin/sh -xe /tmp/jenkins16378467277877437643.sh

+ echo Physical Memory Information

Physical Memory Information

+ free -m

	total	used	free	shared	buff/cache	available
Mem:	978	565	84	0	328	273

Swap: 0 0 0

Triggering a new build of DISK INFORMATION

Finished: SUCCESS

Dashboard > DISK INFORMATION > #1

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[View as plain text](#)

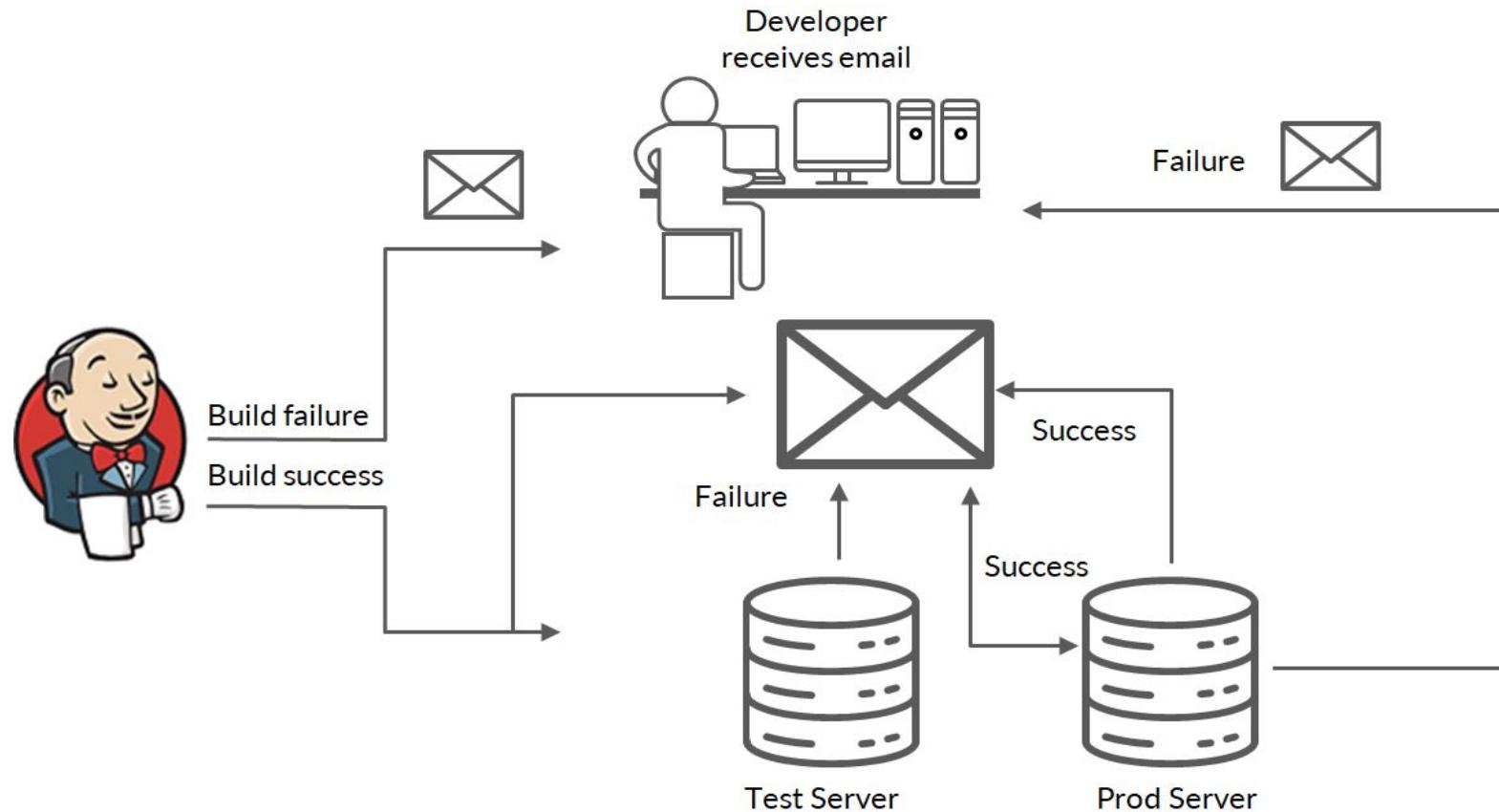
[Edit Build Information](#)

[Delete build '#1'](#)

Console Output

```
Started by upstream project "RAM INFORMATION" build number 1
originally caused by:
    Started by upstream project "CPU INFORMATION" build number 1
originally caused by:
        Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/DISK INFORMATION
[DISK INFORMATION] $ /bin/sh -xe /tmp/jenkins11158508944207481281.sh
+ echo DISK INFORMATION
DISK INFORMATION
+ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            476M   0    476M  0% /dev
tmpfs           98M  784K  98M  1% /run
/dev/xvda1       7.7G  2.3G  5.5G  30% /
tmpfs           490M   0   490M  0% /dev/shm
tmpfs           5.0M   0   5.0M  0% /run/lock
tmpfs           490M   0   490M  0% /sys/fs/cgroup
/dev/loop0        32M   32M   0  100% /snap/snapd/11036
/dev/loop1        34M   34M   0  100% /snap/amazon-ssm-agent/3552
/dev/loop2        56M   56M   0  100% /snap/core18/1988
tmpfs           98M   0   98M  0% /run/user/111
/dev/loop3        33M   33M   0  100% /snap/snapd/11107
tmpfs           98M   0   98M  0% /run/user/1000
Finished: SUCCESS
```

Email Notification



- If build failed or the build status changed from failed to success, then Jenkins has a provision to send an email notification to the given recipients.
- Install Email Extension Plugin and Mailer Plugin



Email Configuration Setup

upGrad

Jenkins

Dashboard > Plugin Manager

Back to Dashboard

Manage Jenkins

Update Center

search

admin log out

Updates Available Installed Advanced

Enabled	Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Email Extension Plugin	2.82		<button>Uninstall</button>
<input checked="" type="checkbox"/>	Mailer Plugin	1.32.1		<button>Uninstall</button>

email configuration setup



Manage Jenkins



Configure System

Configure global settings and paths.

Jenkins Location

Jenkins URL

http://3.95.7.132:8080/

System Admin e-mail address

Jenkins-Master <drbalat.raju@gmail.com>

Extended E-mail Notification

SMTP server

smtp.gmail.com

SMTP Port

465

E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix



Test configuration by sending test e-mail

Save

Apply

Email Configuration Setup

upGrad

E-mail Notification

SMTP server	<input type="text" value="smtp.gmail.com"/>	?
Default user e-mail suffix	<input type="text"/>	?
<input checked="" type="checkbox"/> Use SMTP Authentication		?
User Name	<input type="text" value="drbalat.raju@gmail.com"/>	
Password	<input type="password" value="Concealed"/> Change Password	?
Use SSL	<input checked="" type="checkbox"/>	?
Use TLS	<input type="checkbox"/>	?
SMTP Port	<input type="text" value="465"/>	?
Reply-To Address	<input type="text"/>	
Charset	<input type="text" value="UTF-8"/>	
<input checked="" type="checkbox"/> Test configuration by sending test e-mail		
Test e-mail recipient	<input type="text" value="drbhangaraju@gmail.com"/>	<input type="button" value="Test configuration"/>

Save **Apply**

Go to Dashboard

Select Project -TestEmail

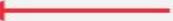
Click Configure

Post-build Actions

E-mail Notification

Recipients 

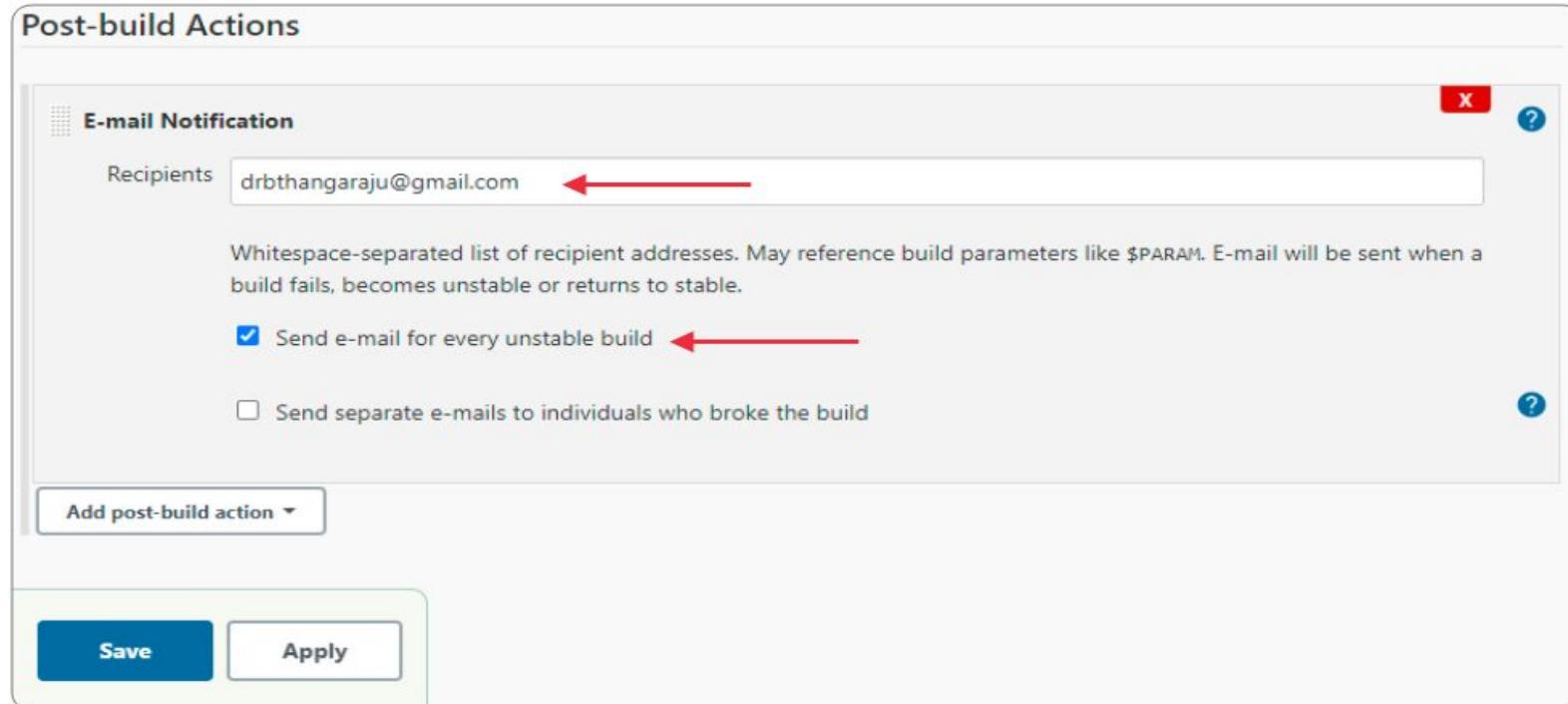
Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

Send e-mail for every unstable build 

Send separate e-mails to individuals who broke the build 

Add post-build action ▾

Save **Apply**



Email Notification

upGrad

The screenshot shows a Gmail inbox with the following details:

- Compose** button
- Inbox** tab (highlighted, 26 messages)
- Primary** tab
- Social** tab
- Promotions** tab (4 new messages: Medical Benefits, MOTILAL OS...)
- Search mail** bar
- Filter** dropdown
- Compose** button
- Starred** section (0 items)
- Message Preview 1:** Jenkins-Master - Build failed in Jenkins: TestEmail #8 - See <<http://3.95.7.132:8080/job/TestEmail/8/display/redirect>> 4:22 PM
- Message Preview 2:** Jenkins-Master - Jenkins build is back to normal : TestEmail #5 - See <<http://3.95.7.132:8080/job/TestEmail/5/display/redirect>> 3:49 PM
- Message Preview 3:** Jenkins-Master - Build failed in Jenkins: TestEmail #4 - See <<http://3.95.7.132:8080/job/TestEmail/4/display/redirect>> 3:48 PM

Build failed in Jenkins: TestEmail #8 ➔ Inbox x



Jenkins-Master <drbalat.raju@gmail.com>

to me ▾

See <<http://3.95.7.132:8080/job/TestEmail/8/display/redirect>>

Changes:

Started by user admin

Running as SYSTEM

Building in workspace <<http://3.95.7.132:8080/job/TestEmail/ws/>>

[TestEmail] \$ /bin/sh -xe /tmp/jenkins8426018425159410393.sh

+ fre -m

/tmp/jenkins8426018425159410393.sh: 2: /tmp/jenkins8426018425159410393.sh: fre: not found

Build step 'Execute shell' marked build as failure

Jenkins build is back to normal : TestEmail #5 ➔ Inbox x

Jenkins-Master <drbalat.raju@gmail.com>

to me ▾

See <<http://3.95.7.132:8080/job/TestEmail/5/display/redirect>>



Console Output

Started by user admin

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/TestEmail

[TestEmail] \$ /bin/sh -xe /tmp/jenkins11996699832483977733.sh

+ free -m

	total	used	free	shared	buff/cache	available
Mem:	978	565	70	0	342	268
Swap:	0	0	0			

Sending e-mails to: drbthangaraju@gmail.com

Finished: SUCCESS

Poll 3 (15 seconds)

Sending an email for every unstable build option enables to send notification:

- A. When any build fails
- B. When any build succeeds.
- C. When any build status changes (fail to success or success to fail)
- D. All of these

Poll 3 (15 seconds)

Sending an email for every unstable build option enables to send notification:

- A. When any build fails
- B. When any build succeeds
- C. When any build status changes (fail to success or success to fail)**
- D. All of these

Parameterized Jenkins Job

```
#!/bin/bash
# A simple calculator shell program

a=$1
b=$2

echo "Enter Choice :"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
ch=$3

case $ch in
 1) res=`expr $a + $b` ;;
 2) res=`expr $a - $b` ;;
 3) res=`expr $a \* $b` ;;
 4) res=`expr $a / $b` ;;
esac
echo "Result : $res"
```

Step 1. Create a freestyle project: **calculator parameterized**

Step 2. Develop a simple calculator shell program in the default workspace

Step 3. During execution, you can choose the a, b and choice through Jenkins 'This project is parameterized' option

Enter the First and Second Parameters

upGrad

The screenshot shows a Jenkins build configuration for a project named "calculator parameterized". The "General" tab is selected. A red arrow points to the checkbox labeled "This project is parameterized", which is checked. Below this, two string parameters are defined:

String Parameter	
Name	first
Default Value	
Description	Please enter the first number

String Parameter	
Name	second
Default Value	
Description	Please enter the second number

Choice Parameter

Name ch ?

Choices

- 1
- 2
- 3
- 4

Description

- 1 - add
- 2 - subtract
- 3 - multiply
- 4 - divide

Build

Execute shell

Command

```
chmod u+x c.sh  
./c.sh ${first} ${second} ${ch}
```

See the list of available environment variables

Advanced...

Add build step ▾

Save

Apply

The screenshot shows a 'Build Step' configuration interface. At the top, there's a 'Build' header and a 'Execute shell' section. Inside the 'Execute shell' section, there's a command input field containing two lines of shell script: 'chmod u+x c.sh' and './c.sh \${first} \${second} \${ch}'. Below this, there's a link to 'See the list of available environment variables'. On the right side of the 'Execute shell' section, there's an 'Advanced...' button. At the bottom of the page, there's a 'Save' button and an 'Apply' button. Two red arrows point from the bottom left towards the 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins interface for a project named "calculator parameterized". The left sidebar contains links: "Back to Dashboard", "Status", "Changes", "Workspace", and "Build with Parameters". A red arrow points to the "Build with Parameters" link. The main content area displays the project name "Project calculator parameterized" and two links: "Workspace" and "Recent Changes".

Jenkins

Dashboard > calculator parameterized >

Back to Dashboard

Status

Changes

Workspace

Build with Parameters

Project calculator parameterized

Workspace

Recent Changes

Project calculator parameterized

This build requires parameters:

first

Please enter the first number

second

Please enter the second number

ch

1 - add

2 - subtract

3 - multiply

4 - divide

Build

This build requires parameters:

first

255

Please enter the first number

second

5

Please enter the second number

ch

1 - add

2 - subtract

3 - multiply

4 - divide

Build





Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/calculator parameterized
[calculator parameterized] $ /bin/sh -xe /tmp/jenkins5141576588902714368.sh
+ chmod u+x c.sh
+ ./c.sh 255 5 4
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Result : 51
Finished: SUCCESS
```



Thank You!