

# CYDEO

CI / CD

---



# Contents

Terminology

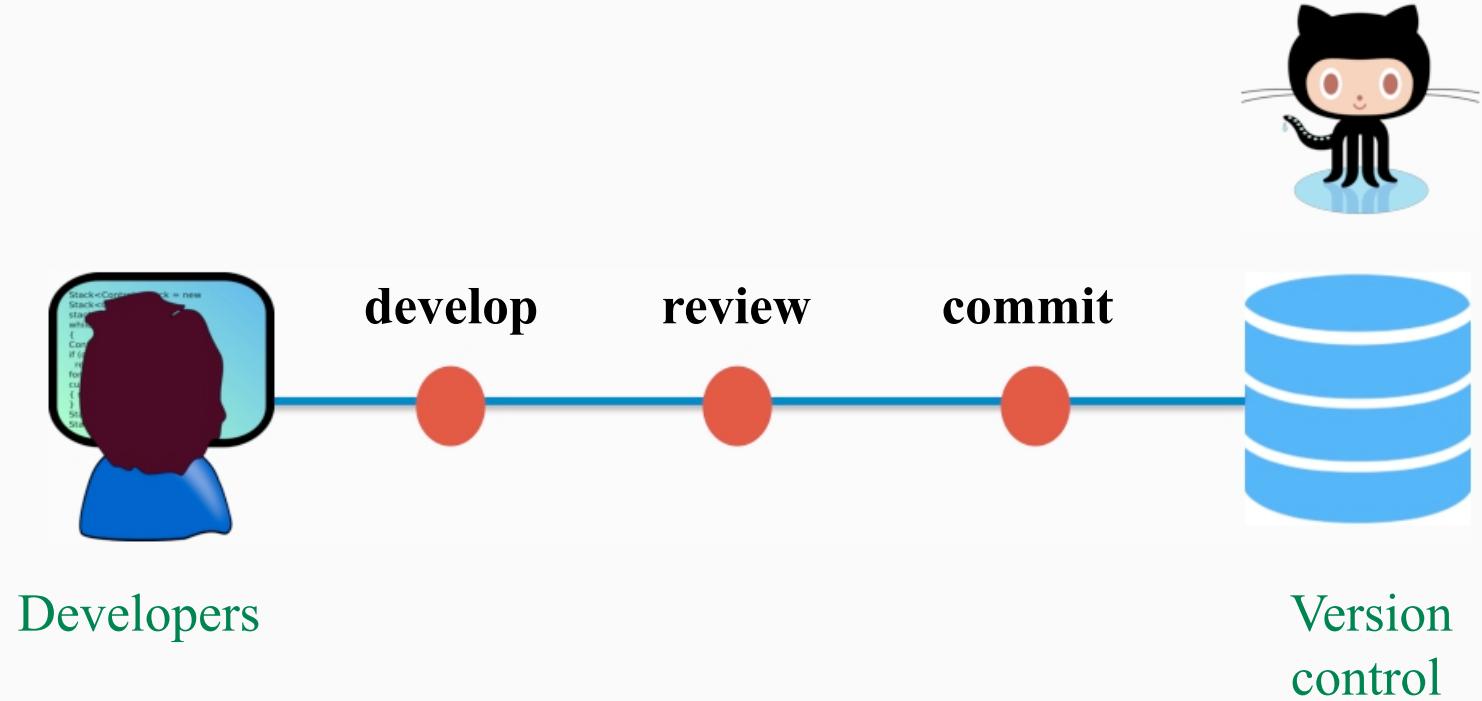
Deployment process

Pipeline example

Automation in pipeline

Jenkins

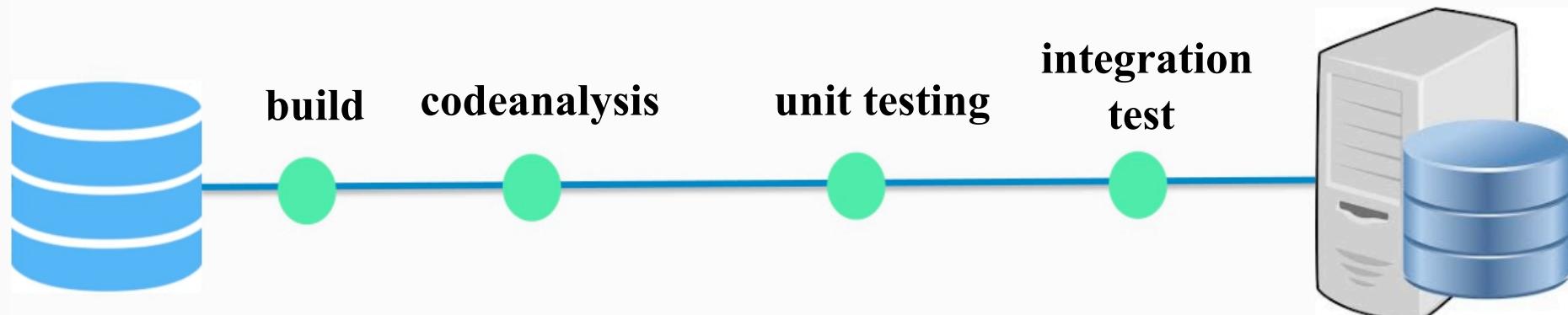






sonarqube

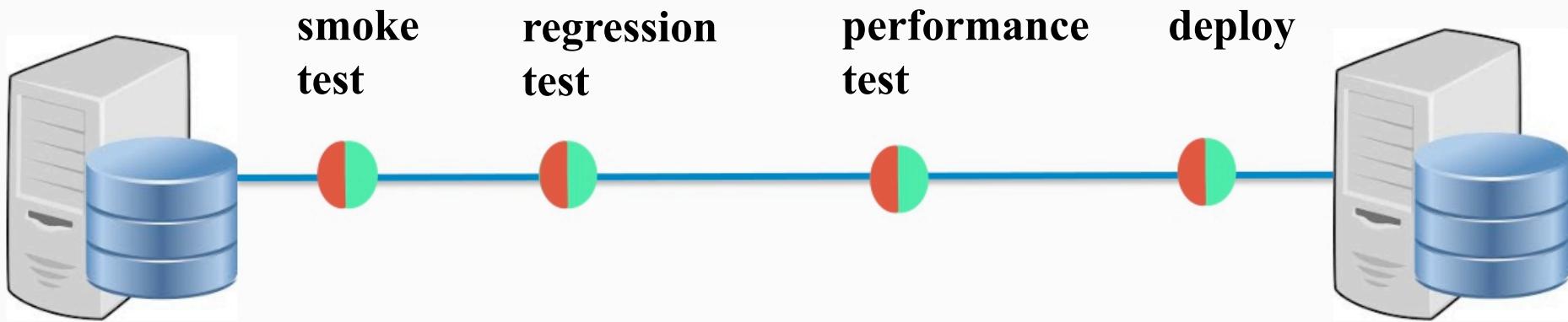
JUnit



VCS  
(github)

Test  
environment

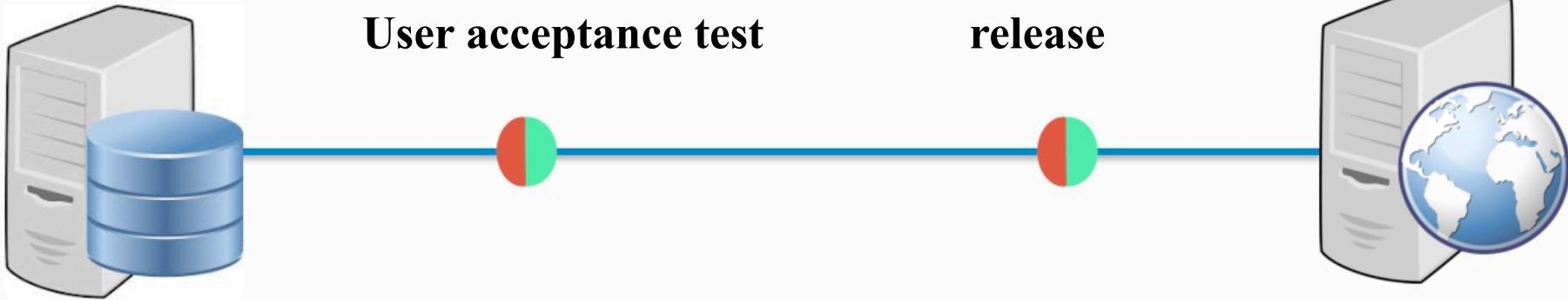




Test  
environment

Staging  
environment

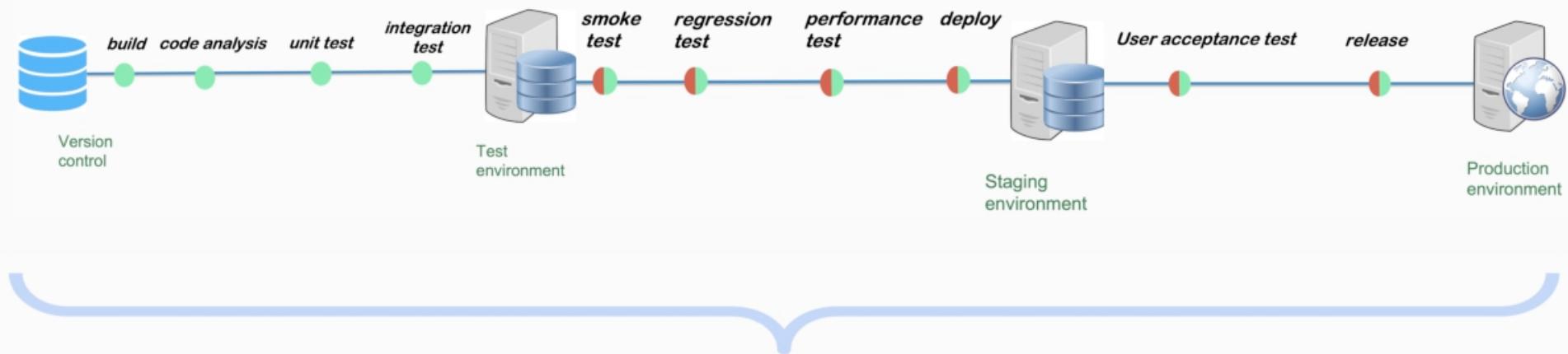




Staging  
environment

Production  
environment





# Production Pipeline



# Pipeline

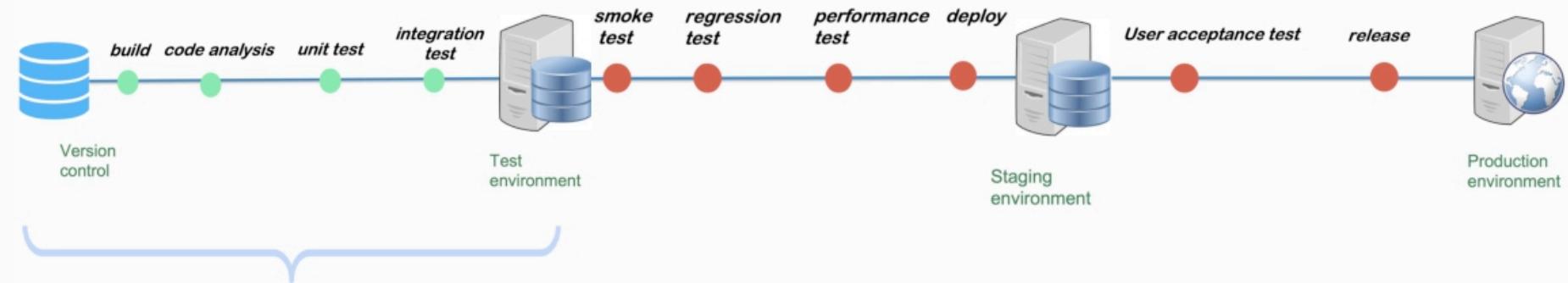
## Pipeline

Pipeline is a set of processes that take the code from version control and compile, build, test and deploy to production in automated fashion.

The pipeline breaks down the software delivery process into stages. Each stage is made of different tasks which can be carried out in parallel. When all tasks in a stage passes, next stage is triggered.



# Continuous Integration



Continuous Integration



# Continuous Integration

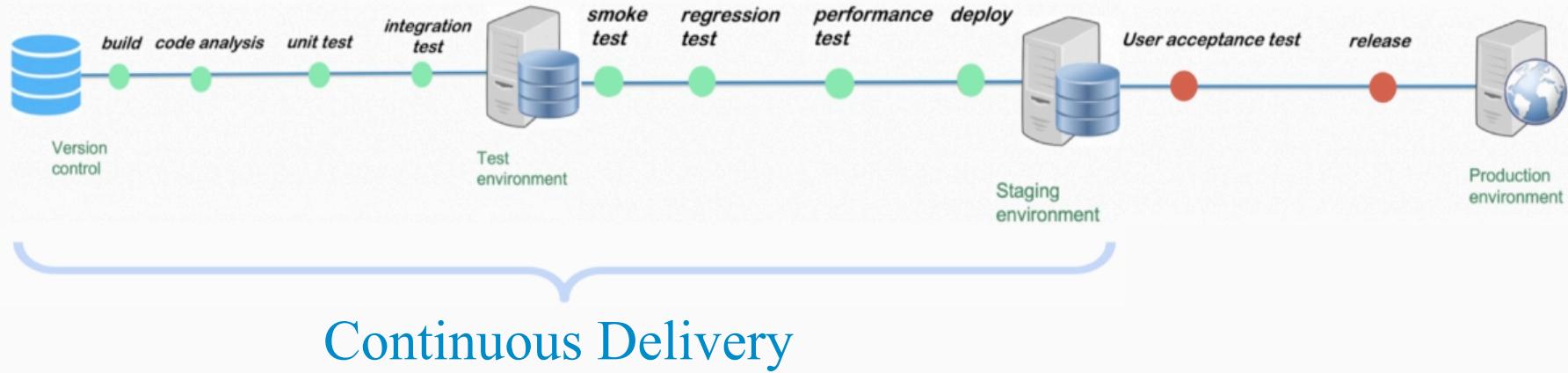
Continuous Integration is a process of automated build, unit testing, integration testing and code analysis.

The Continuous Integration process is comprised of automatic tools that assert the new code's correctness before integration. It reduces integration problems allowing to deliver software more rapidly by providing quick feedback every time new code is added to the source control. Usually Continuous Integration does not involve testing the functionality of the application.

**In a simple words:** the practice of merging all developers' working copies to a shared mainline several times a day.



# Continuous Delivery



# Continuous Delivery

Continuous Delivery is an automated build and execution of at unit and integration tests, execution of code analysis, functional tests and also deploying to any supported platform any time. Each time a build or a set of code passes the tests, it's automatically deployed out to a staging environment.

In Continuous Delivery releasing to end users is a manual process. Continuous delivery involves human decision-making when it comes to deciding when to release the software to the customers.



# Continuous Delivery

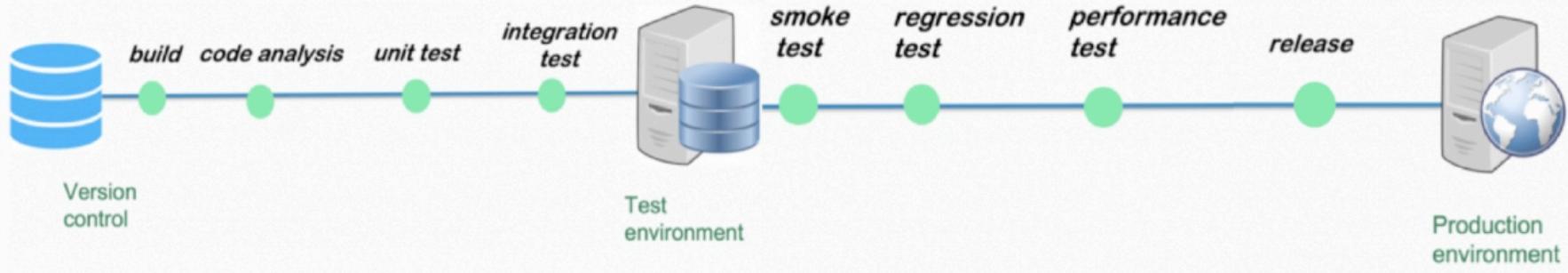
## Continuous Delivery

In a simple words: the practice in which teams produce and release value in short cycles.  
8 Principles of Continuous Delivery:

1. Repeatable Reliable Process
2. Automate Everything
3. Version Control Everything
4. Bring the Pain Forward
5. Build-in Quality
6. "Done" Means Released
7. Everyone is Responsible
8. Continuous Improvement



# Continuous Deployment



Continuous Deployment



# Continuous Deployment

Continuous deployment means that every change that you make, goes through the pipeline, and if it passes all the tests, it automatically gets deployed into production.

When a developer checks in code, the automated processes take the code and move it through the entire lifecycle and if it passes each gate, it gets deployed directly to production. The delivery speeds are notably faster due to elimination of manual steps.

**In a simple words:** the practice in which the value is delivered frequently through automated deployments.



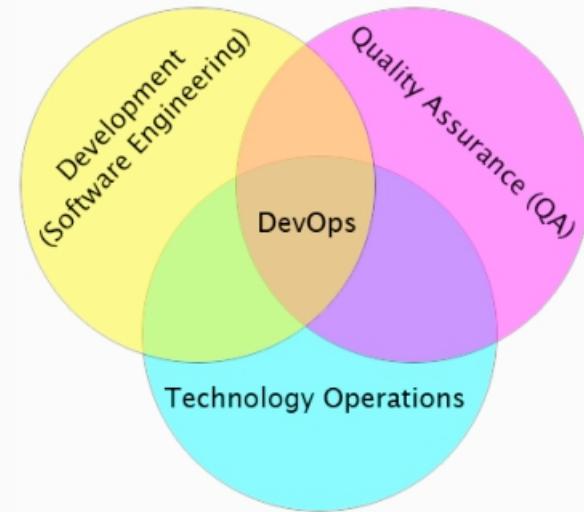
# DevOps

Methodology (like agile, waterfall)

DevOps merges developer, tester and operations roles together. whoever writes the software is also responsible and deploying and maintaining it.

DevOps organizations break down the barriers between Operations and Engineering by cross - training each team in the other's skills.

DevOps focuses on culture highlighting roles that emphasize responsiveness and breaking down barriers between developers and operations teams.



# What is Server ?

## What is a server?

A server is a computer program or device that provides a service to another computer program and its user, also known as the client. In a data center, the physical computer that a server program runs on is also frequently referred to as a server. That machine might be a dedicated server or it might be used for other purposes.

In the client/server programming model, a server program awaits and fulfills requests from client programs, which might be running in the same or other computers. A given application in a computer might function as a client with requests for services from other programs and as a server of requests from other programs.



# What is Server ?



# Physical and Virtual Servers

1. A physical server is simply a computer that is used to run server software.
2. A virtual server is a virtual representation of a physical server. Like a physical server, a virtual server includes its own operating system and applications. These are kept separate from any other virtual servers that might be running on the physical server.

The process of creating virtual machines involves installing a lightweight software component called a hypervisor onto a physical server. The hypervisor's job is to enable the physical server to function as a virtualization host. The virtualization host makes the physical server's hardware resources - such as CPU time, memory, storage and network bandwidth - available to one or more virtual machines.



# Servers in the Cloud

Servers in the cloud have revolutionized the IT industry.

- Scale capacity up and down based on demands.
- Storage, more memory, and computing power can be added as needed.
- Obtain servers in minutes.
- No need for onsite hardware or capital expenses.



# Main Types of Cloud Computing

SaaS, PaaS, and IaaS are simply three ways to describe how you can use the cloud.

- IaaS: cloud-based services, pay-as-you-go for services such as storage, networking, and virtualization.
- PaaS: hardware and software tools available over the internet.
- SaaS: software that's available via a third-party over the internet.
- On-premise: software that's installed in the same building as your business.

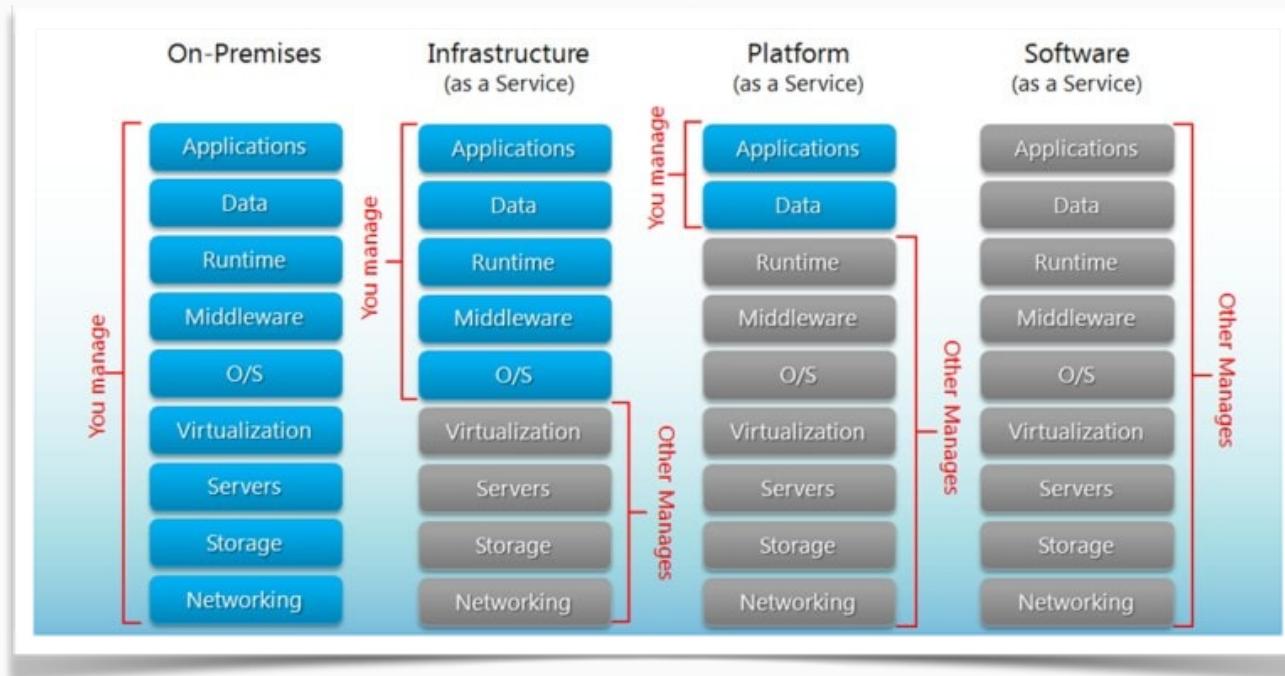
\*IaaS - Infrastructure As A Service

\*PaaS - Platform As A Service

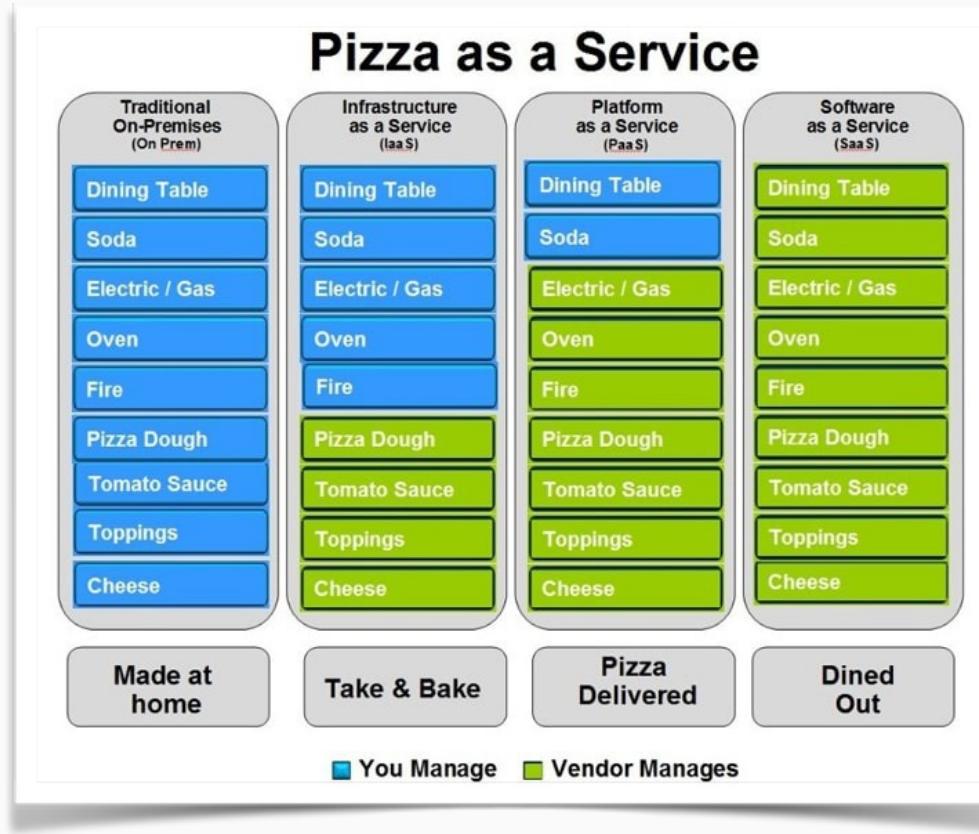
\*SaaS - Software As A Service



# Main Types of Cloud Computing



# Main Types of Cloud Computing



# Main Types of Cloud Computing

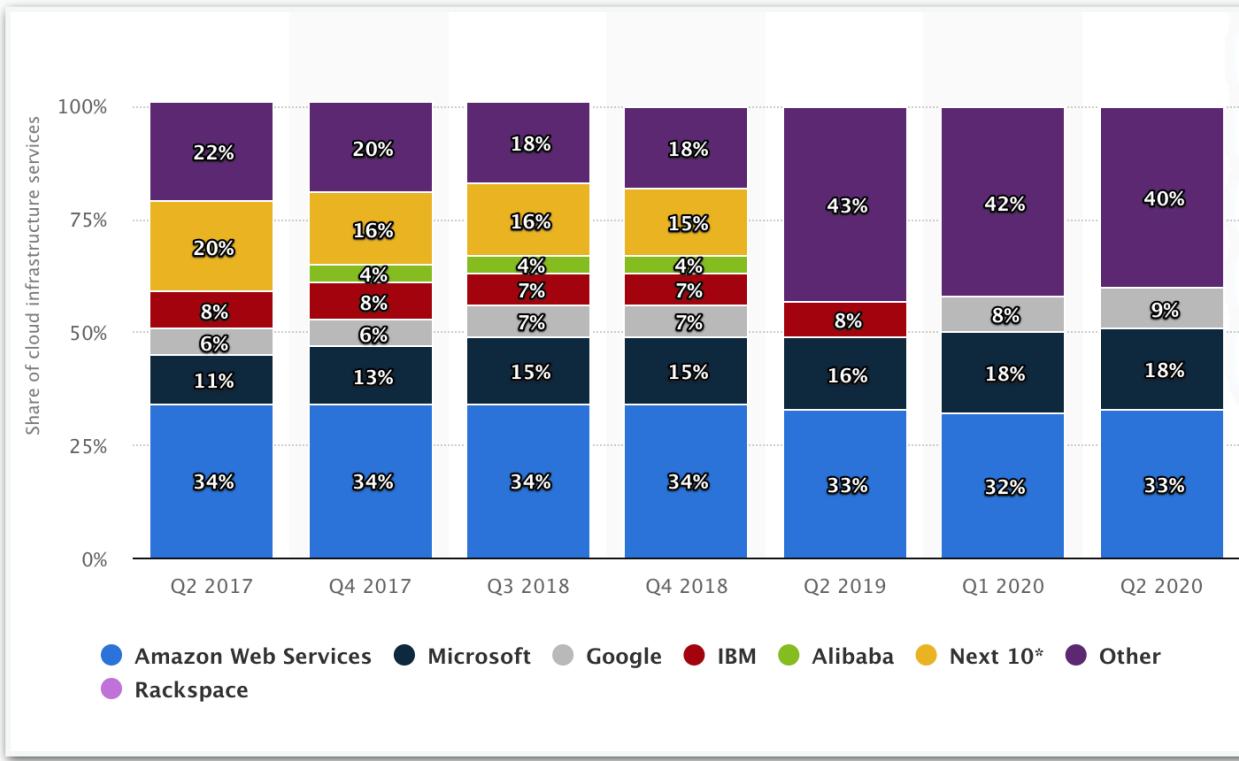
**SaaS:** BigCommerce, **Google Apps**, Salesforce, **Dropbox**, MailChimp, ZenDesk, DocuSign, **Slack**, Hubspot.

**PaaS:** AWS Elastic Beanstalk, Heroku, Windows Azure (mostly used as PaaS), Force.com, OpenShift, Apache Stratos, Magento Commerce Cloud.

**IaaS examples:** **AWS EC2**, Rackspace, Google Compute Engine (GCE), Digital Ocean, Magento 1 Enterprise Edition\*.



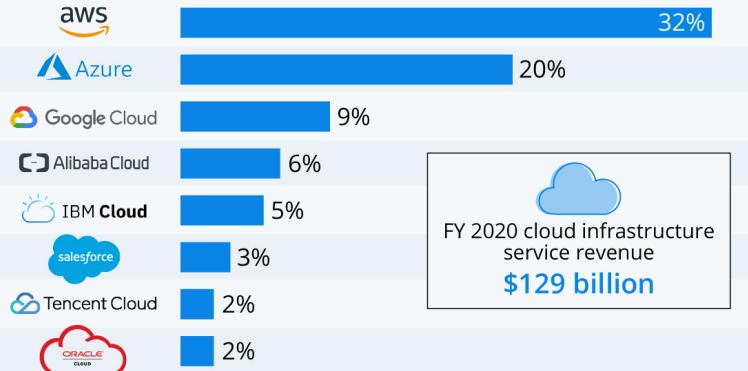
# Cloud Computing Market



# Cloud Computing Market

## Amazon Leads \$130-Billion Cloud Market

Worldwide market share of leading cloud infrastructure service providers in Q4 2020\*



\* includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

Source: Synergy Research Group



statista



# Cloud Computing with AWS

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

Cloud computing is the on-demand delivery of compute power, database storage, applications and other IT resources through a cloud services platform via the Internet with pay-as-you go pricing

AWS is made up of so many different cloud computing products and services. The highly profitable Amazon division provides servers, storage, networking, remote computing, email, mobile development, and security. AWS can be broken into three main products: EC2, Amazon's virtual machine service, Glacier, a low-cost cloud storage service, and S3, Amazon's storage system. AWS is so large and present in the computing world that it's far outpaced its competitors.



# Cloud Computing with AWS

AWS has 81 availability zones in which its servers are located. These serviced regions are divided in order to allow users to set geographical limits on their services (if they so choose), but also to provide security by diversifying the physical locations in which data is held. Overall, AWS spans 245 countries and territories.



# AWS EC2

Elastic Cloud Compute or EC2 is a foundational piece of AWS' cloud computing platform and is a service that provides servers for rent in the cloud.

## Pricing Options

There are several pricing options for EC2.

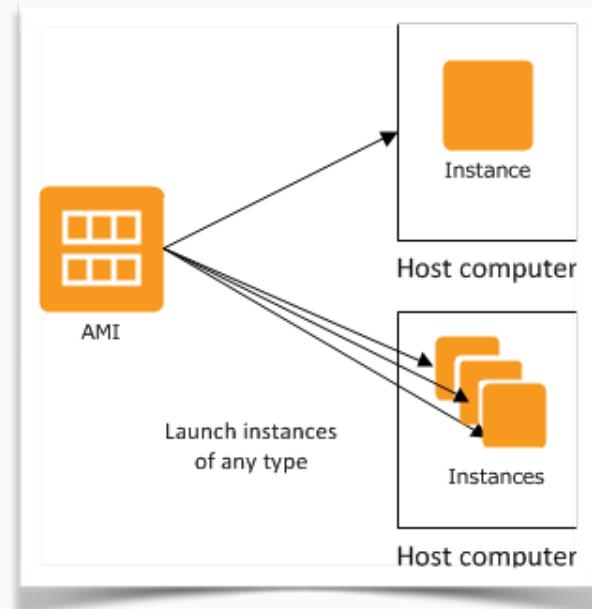
- On Demand - Pay as you go, no contract.
- Dedicated Hosts - You have your own dedicated hardware and don't share it with others.
- Spot - You place a bid on an instance price. If there is extra capacity that falls below your bid, an EC2 instance is provisioned. If the price goes above your bid while the instance is running, the instance is terminated.
- Reserved Instances - You earn huge discounts if you pay up front and sign a 1-year or 3-year contract.



# AMI

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications).

From an AMI, you launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud. You can launch multiple instances of an AMI, as shown in the following figure.



# JENKINS

1. CI/CD tool
2. Open source
3. Automates building, testing, packaging, staging, deploying the application
4. Integrates with different tools using plugins

Alternatives: Travis, TeamCity, Bamboo , CircleCI, GitHub Actions



# JENKINS

## Jenkins Job

In Jenkins everything is done by creating a job

1. a task that Jenkins performs based its schedule
2. be made of several steps
3. can have a schedule or a trigger which determines when it runs
4. reports the results of the run automatically



# CYDEO

## Jenkins Setup

---



# JENKINS

1. Open any browser
2. Type here **YOUR IP** from EC2 instance
3. Use this password to unlock Jenkins
4. Click Continue



**Example :** **54.165.159.235:8081**

This password same for ALL USERS

The screenshot shows the Jenkins 'Getting Started' page with the title 'Unlock Jenkins'. It instructs the user to copy the password from either the log or a file on the server and paste it into the 'Administrator password' field. The password '1151263c93dc4daca0b33a229b33879a' is highlighted in red. A red arrow points to the 'Continue' button at the bottom right.

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:

```
C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\secrets\initialAdmin
```

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

Administrator password

1151263c93dc4daca0b33a229b33879a

Continue



# JENKINS

Getting Started



## Customize Jenkins

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

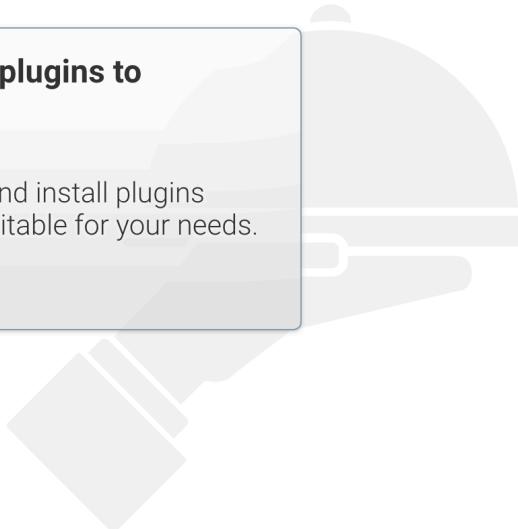
### Install suggested plugins

Install plugins the Jenkins community finds most useful.

### Select plugins to install

Select and install plugins most suitable for your needs.

- Install Suggested Plugins



# JENKINS

- Wait for all installations

## Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	... <small>✓ <a href="#">View API</a></small>
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✗ Gradle	** Bootstrap 4 API ** SnakeYAML API ** Jackson 2 API ** Popper.js 2 API ** Bootstrap 5 API ** ECharts API ** Display URL API ** Pipeline: Supporting APIs ** Checks API ** JUnit ** Matrix Project ** Resource Disposer
✗ Pipeline	✗ GitHub Branch Source	✗ Pipeline: GitHub Groovy Libraries	✗ Pipeline: Stage View	Workspace Cleanup
✗ Git	✗ SSH Build Agents	✗ Matrix Authorization Strategy	✗ PAM Authentication	Ant ** JAXB ** Durable Task ** Pipeline: Nodes and Processes ** Command Agent Launcher ** Oracle Java SE Development Kit Installer ** bouncycastle API ** JavaScript GUI Lib: ACE Editor bundle
✗ LDAP	✗ Email Extension	✗ Mailer		** - required dependency

Jenkins 2.277.1



# JENKINS

1. Create your user in JENKINS
2. These will be your information
3. After fill the form click  
***Save and Continue***

DON'T FORGET PASSWORD

Getting Started

## Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:



Jenkins 2.277.1 [Skip and continue as admin](#) [Save and Continue](#)



# JENKINS

## Getting Started

# Instance Configuration

Jenkins URL:

http://54.165.159.235:8081/

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.

- Click Save and Finish



# JENKINS

## Getting Started

# Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

- Click Start Using JENKINS



# JENKINS

 Jenkins

Dashboard >

- + New Item
- People
- Build History
- Manage Jenkins
- My Views
- New View

**Build Queue** ▾

No builds in the queue.

**Build Executor Status** ▾

1 Idle
2 Idle

**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](#)

 Search 

 1  mehmet  log out



# JENKINS

- Dashboard —> Manage JENKINS —> Global Tool Configuration

The image shows three screenshots of the Jenkins interface illustrating the navigation steps:

- Left Screenshot:** Shows the main Jenkins dashboard sidebar with various options like "New Item", "People", "Build History", etc. A red arrow points from the "Dashboard" link at the top of the sidebar to the "Manage Jenkins" option in the main menu.
- Middle Screenshot:** Shows the "Manage Jenkins" dropdown expanded. A red arrow points from the "Manage Jenkins" link in the sidebar to the "Manage Jenkins" link in the dropdown menu.
- Right Screenshot:** Shows the "System Configuration" page. The "Global Tool Configuration" link is highlighted with a blue background and a light blue border, indicating it is the selected item.

**System Configuration**

- Configure System
- Global Tool Configuration**
- Manage Plugins
- Manage Nodes and Clouds
- Security
- Configure Global Security
- Manage Credentials



# JENKINS

## Global Tool Configuration



### Maven Configuration

Default settings provider

Use default maven settings

Default global settings provider

Use default maven global settings

### JDK

JDK installations

Add JDK

List of JDK installations on this system

- Click Add JDK



# JENKINS

- Uncheck it 

JDK

Name

 Required

Install automatically 

Install Oracle Java SE Development Kit from the website 

Version

Java SE Development Kit 9.0.4 

I agree to the Java SE Development Kit License Agreement

 **Installing JDK requires Oracle account. Please enter your username/password**

Oracle Java SE 11+ is not available for business, commercial or production use without a commercial license.  
Public updates for Oracle Java SE 8 released after January 2019 will not be available for business, commercial or production use without a commercial license.  
[Oracle Java SE Licensing FAQ](#)

 Delete Installer

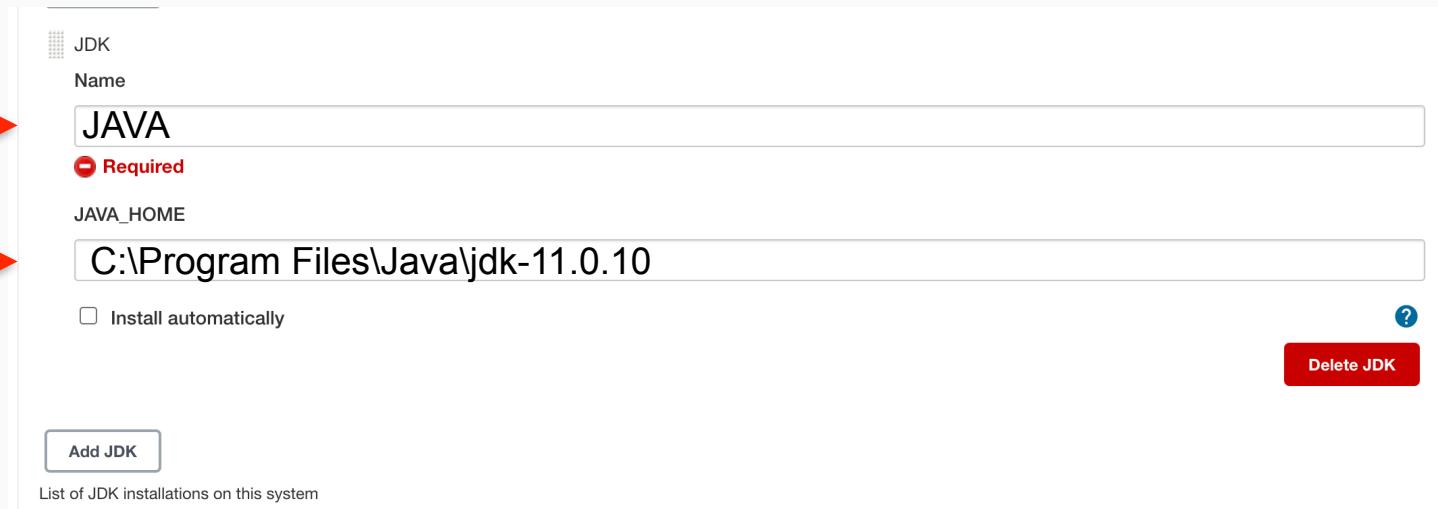
Add Installer 

 Delete JDK



# JENKINS

- Put this information in required fields



- Make sure there is no extra spaces



# JENKINS

- Click here to remove



# JENKINS

- Select JGit from list

**Git**

Git installations

Add Git ▾

- Git
- JGit**
- JGit with Apache HTTP client

**Git**

Git installations

≡ **JGit** ? ✖

Add Git ▾



# JENKINS

- Click Add Maven

## Maven

### Maven installations

Add Maven

List of Maven installations on this system



# JENKINS

- Type here M3

Maven installations

Add Maven

Maven

Name

M3 

Install automatically 

Install from Apache

Version

3.8.5 

 Delete Installer

Add Installer 

 Delete Maven

Add Maven

List of Maven installations on this system

Save  Apply  • Apply and SAVE



# How to Install Cucumber Plugin to Jenkins

- Go to Jenkins Plugin Management to install
- Select Available
- Search for **cucumber**
- Select **Cucumber Reports**
- **Install without Restart**

Search: cucumber

Updates Available Installed Advanced

Install ↑ Name Version

**Cucumber reports**  
Build Reports  
 Provides pretty html reports for Cucumber. Can be used anywhere a json report is generated (Java, Ruby, JavaScript and other implementations).  
5.7.0

Install without restart   Download now and install after restart   Update information obtained: 7 min 42 sec ago   Check now

Jenkins

Dashboard >

+ New Item  
People  
Build History  
Manage Jenkins ▾

System Configuration  
Configure System  
Global Tool Configuration  
Manage Plugins  
Manage Nodes and Clouds  
Security

Cucumber reports  
Loading plugin extensions  
Success Success

[Go back to the top page](#)  
(you can start using the installed plugins right away)

Restart Jenkins when installation is complete and no jobs are running

**Restart Jenkins**



# CYDEO

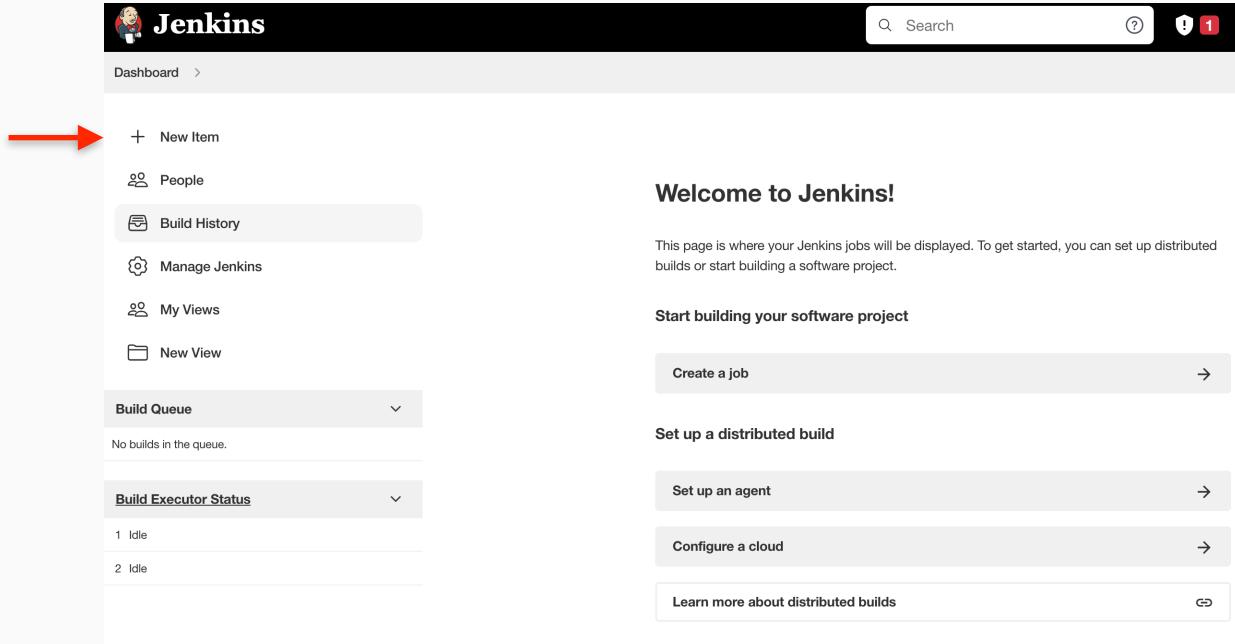
## My First Job

---



# JENKINS

- Click on Dashboard
- Click New Item



The screenshot shows the Jenkins dashboard. On the left, there is a sidebar with several options: 'Dashboard' (selected), '+ New Item' (highlighted with a red arrow), 'People', 'Build History' (selected), 'Manage Jenkins', 'My Views', and 'New View'. Below this is a 'Build Queue' section with a dropdown menu showing 'No builds in the queue.' At the bottom of the sidebar is a 'Build Executor Status' section with a dropdown menu showing '1 Idle' and '2 Idle'. On the right side of the dashboard, there is a main content area with the heading 'Welcome to Jenkins!'. It contains a message: '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.' Below this are several buttons: 'Start building your software project', 'Create a job', 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.



# JENKINS

- Give a name

- Choose Freestyle Project 

Enter an item name

» Required field

 **Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM used for something other than software build.

 **Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building 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 builds, etc.

 **Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. It provides a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

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

 **Organization Folder**

- Click OK



# Source Code Management

1. Select Git
2. Enter your repository url information
3. Select the branch to build

## NOTE :

For private repositories you should put Credentials

## REPOSITORY URL :

Check CI/CD Notes Module **demo-project**

The screenshot shows the Jenkins 'Source Code Management' configuration page. A red box highlights the 'Source Code Management' tab. Below it, the 'Git' option is selected, indicated by a blue radio button, with a red arrow pointing to it. Another red arrow points to the 'Repository URL' field, which contains the value `https://github.com/vasyafomiuk/demo-project`. Under 'Credentials', there is a dropdown set to '- none -' and a 'Add' button. At the bottom, a 'Branches to build' section shows a 'Branch Specifier' field containing `*/master`, with a red arrow pointing to it. The top navigation bar includes tabs for General, Build Triggers, Build Environment, Build, and Post-build Actions.



# Build Triggers

1. Click Build Triggers
2. Select Build Periodically
3. Enter schedule as [cron expression](#)

The screenshot shows the 'Build Triggers' tab selected in a Jenkins configuration interface. The 'Build Triggers' section is displayed, containing several trigger options:

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ? (with a red arrow pointing to it)

Under the 'Build periodically' option, there is a 'Schedule' field containing the cron expression "H 5 \* \* \*". A red arrow points to this field. Below the schedule, a note states: "Would last have run at Thursday, May 26, 2022 at 5:17:34 AM Coordinated Universal Time; would next run at Friday, May 27, 2022 at 5:17:34 AM Coordinated Universal Time." At the bottom of the 'Build Triggers' section, there are two additional options:

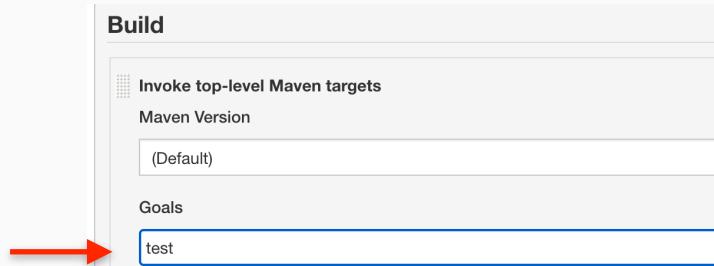
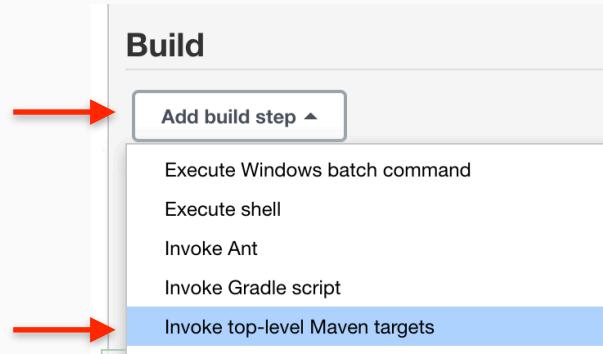
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?



# Build

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

1. Click on Build
2. Click on **Add build step**
3. Select **Invoke top-level Maven targets**
4. Enter maven goal that executes your tests



# Post-Build Actions

The screenshot shows the Jenkins build configuration interface. The top navigation bar includes tabs: General, Source Code Management, Build Triggers, Build Environment, Build, and Post-build Actions. The 'Post-build Actions' tab is highlighted with a red box and has a red arrow pointing to it from the list of actions. A dropdown menu titled 'Add post-build action ▾' lists several actions: Aggregate downstream test results, Archive the artifacts, Build other projects, Cucumber reports (which is selected and highlighted with a blue background), Publish JUnit test result report, Record fingerprints of files to track usage, Git Publisher, E-mail Notification, Editable Email Notification, Set GitHub commit status (universal), Set build status on GitHub commit [deprecated], and Delete workspace when build is done.

1. Click on Post-Build Actions
2. Click on **Add post-build action**
3. Select **Cucumber Reports**
4. **Save**

Save Apply



# JENKINS

[↑ Back to Dashboard](#)

 Status

 Changes

 Workspace

 Build Now

Click on Build Now  


 Configure

 Delete Project

## Project my-first-project



Workspace



Recent Changes

## Permalinks





## How to Send Discord Notifications

---



# How to Install Discord Notifier Plugin to Jenkins

- Go to Jenkins Plugin Management to install
- Select Available
- Search for **discord**
- Select **Discord Notifier**
- **Download now and install after restart**

Updates Available Installed Advanced

Q. discord

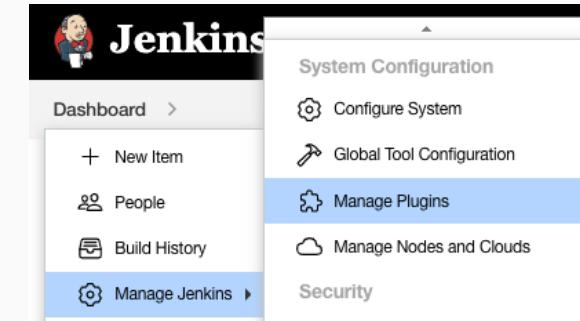
Install Name ↓ Released

**Discord Notifier** 206.vee9f4569ee63  
Discord Notifier allows you to send Discord embeds about your builds via Discord's webhooks.

4 mo 22 days ago

Install without restart Download now and install after restart

Update information obtained: 1 hr 14 min ago Check now



Cucumber reports Success

Loading plugin extensions Success

[Go back to the top page](#)  
(you can start using the installed plugins right away)

[Restart Jenkins](#) when installation is complete and no jobs are running



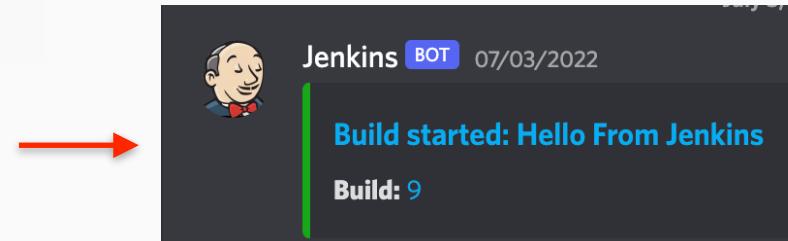
# How to Add Discord Notifications to Job

The screenshot shows the Jenkins interface with a sidebar on the left containing various build actions. A red arrow points from the bottom of this sidebar up towards the 'Discord Notifier' option, which is highlighted with a blue background. Below the sidebar is a blue button labeled 'Add post-build action ▾'.

- Filter
- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Cucumber reports
- Publish JUnit test result report
- Record fingerprints of files to track usage
- Git Publisher
- Discord Notifier**
- E-mail Notification
- Editable Email Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Delete workspace when build is done

Add post-build action ▾

- Click Add post-build action
- Choose Discord Notifier
- Take Webhook from Discord Channel
- Paste **Webhook URL** field
- Save and Apply
- Run Jenkins job to get notifications





## How to Send Email Notifications

---



# Gmail Settings

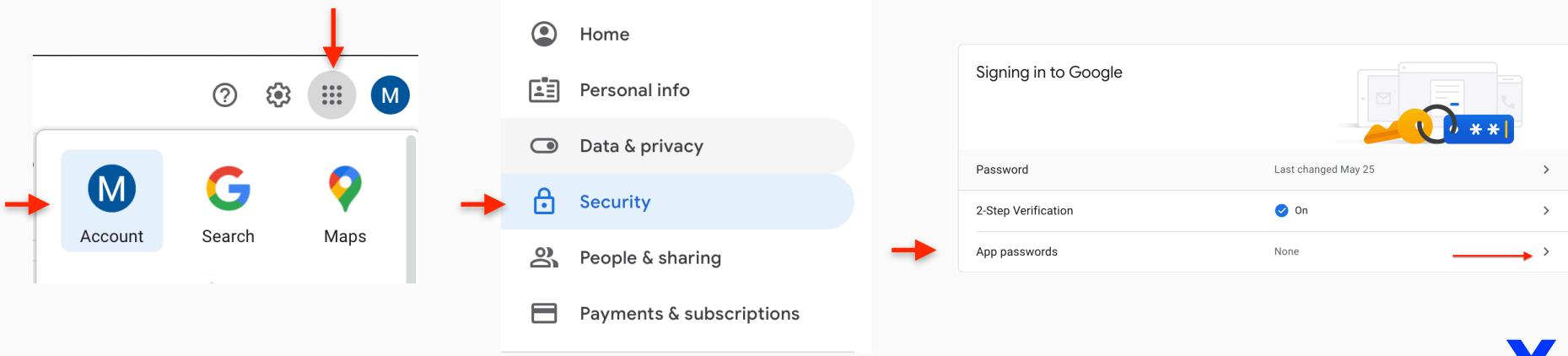
1. Log in Gmail Account
2. Click Account
3. Click Security
4. Click App Passwords

## WARNING :

Don't use your personal email account.

Since we will give access and share password with Jenkins  
Create a new email to use only practice purpose.

If you can't see App Passwords, enable 2-step verification



# Gmail Settings

1. Click select app
2. Give a name ( Jenkins )
3. Click **Generate**



← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

You don't have any app passwords.

Select the app and device you want to generate the app password for.

Jenkins X

**GENERATE**

← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

You don't have any app passwords.

Select the app and device you want to generate the app password for.

Select app

Mail

Calendar

Contacts

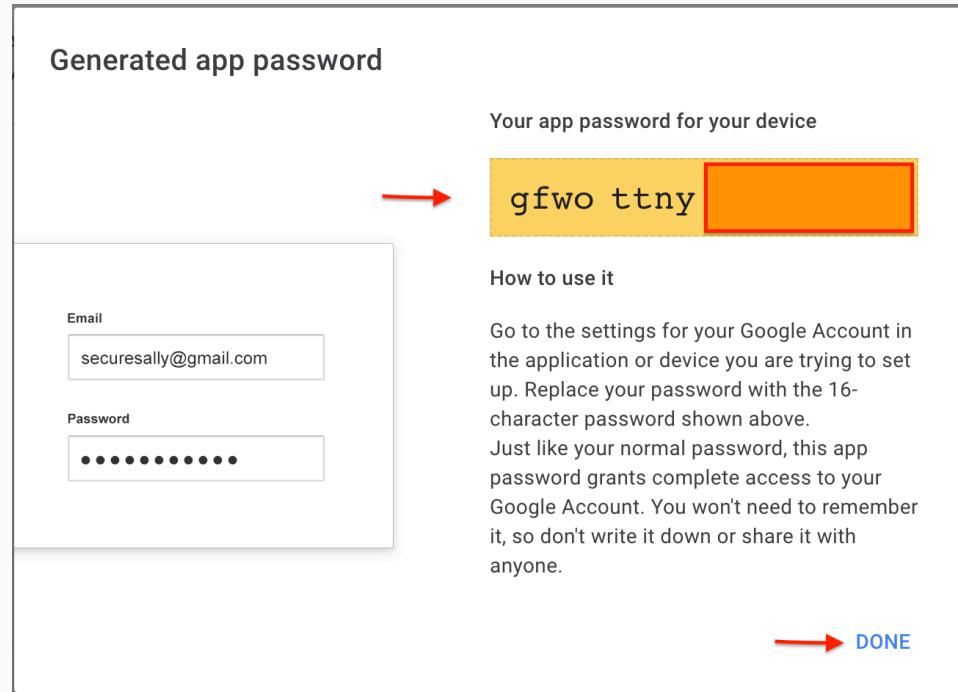
YouTube

Other (Custom name) GENERATE



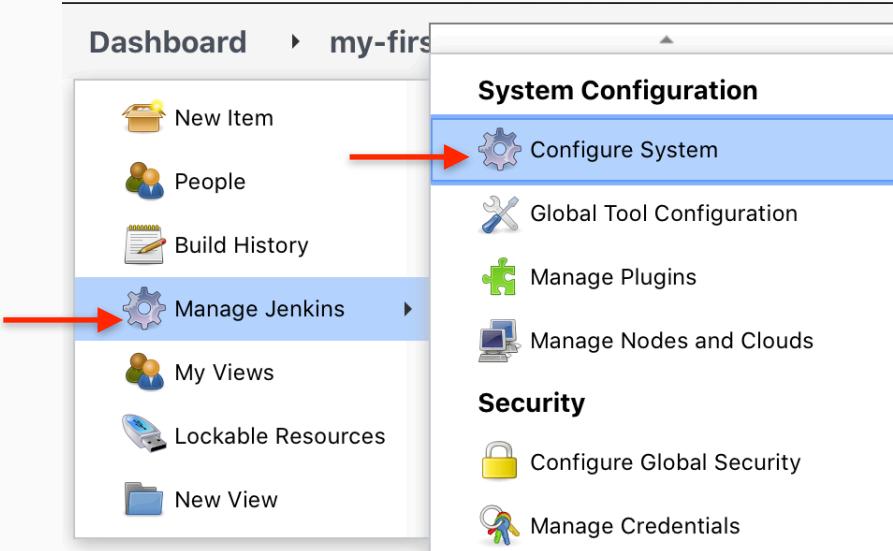
# Gmail Settings

1. Copy generated password from here to use int Jenkins
2. Click DONE



# Jenkins Email Settings

1. Click dropdown from **Dashboard**
2. Click **Configure System**



# Jenkins Email Settings

1. Scroll down to **Jenkins Location**
2. IP should match with your IP from url

IP from URL

The screenshot shows the Jenkins Location configuration page. At the top, it says "Jenkins Location". Below that, there is a "Jenkins URL" field containing "http://54.165.159.235:8081/". To the right of this field is a status bar that says "Not Secure — 54.165.159.235". A red arrow points to the left of the URL field, and another red arrow points down to the status bar. The status bar itself is also highlighted with a red rectangle.

Jenkins Location

Jenkins URL ?

http://54.165.159.235:8081/

Not Secure — 54.165.159.235

System Admin e-mail address ?

address not configured yet <nobody@nowhere>



# Jenkins Email Settings

1. Scroll down to ***Extended E-mail Notification***
2. **Server** : smtp.gmail.com
3. **Port** : 465
4. Click Advanced

Extended E-mail Notification

SMTP server

smtp.gmail.com 

SMTP Port

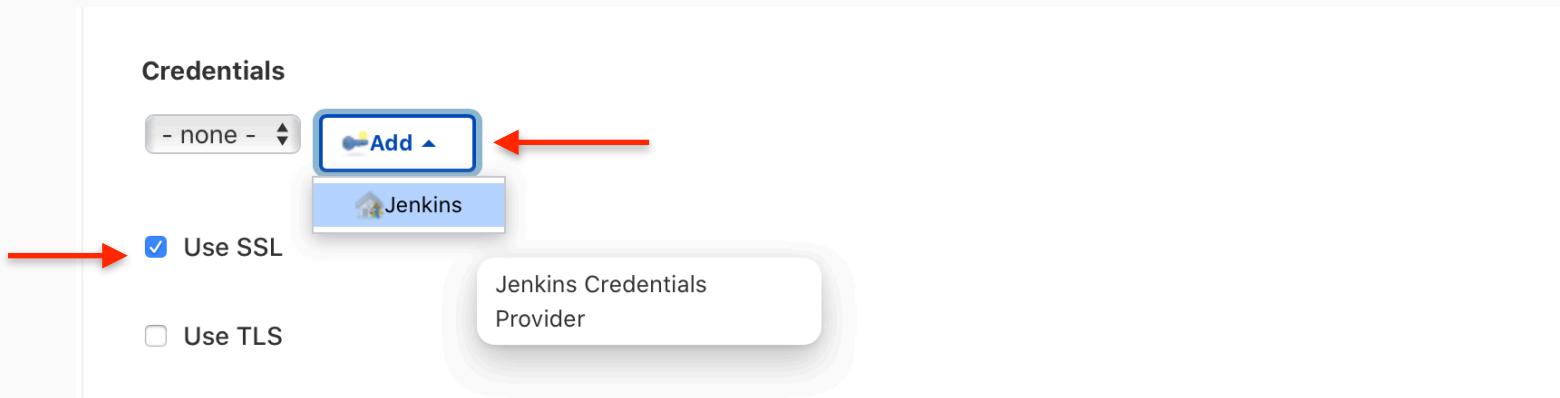
465 

 Advanced...



# Jenkins Email Settings

1. Select **Use SSL**
2. Click Add Jenkins

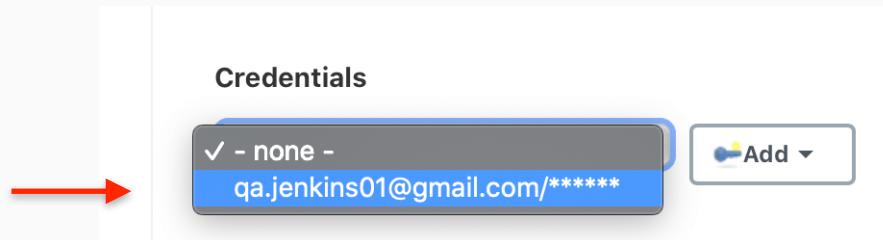
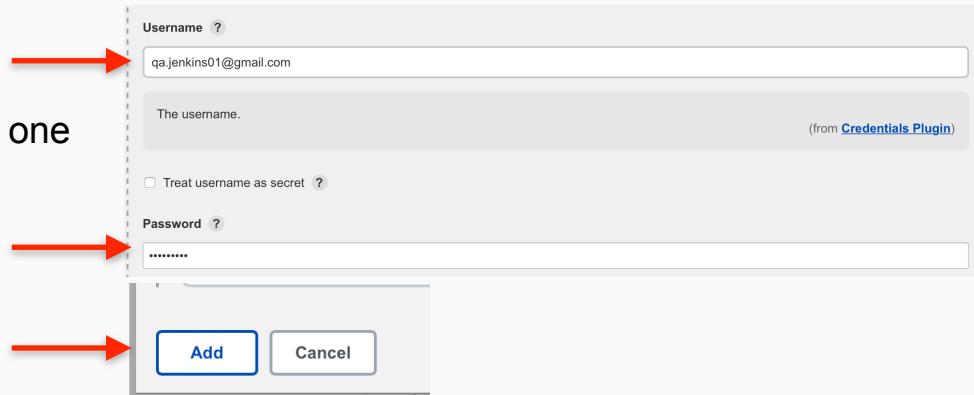
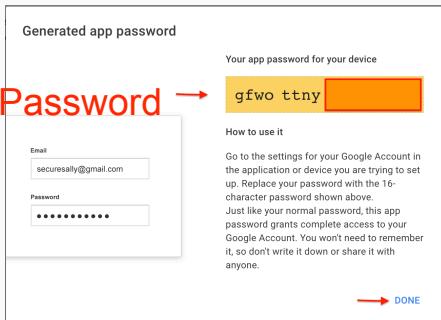


# Jenkins Email Settings

1. Put your credentials here
2. Password same with Google's generated one
3. Click Add
4. Select credentials from dropdown

## NOTE :

Create a new email to use only practice purpose.



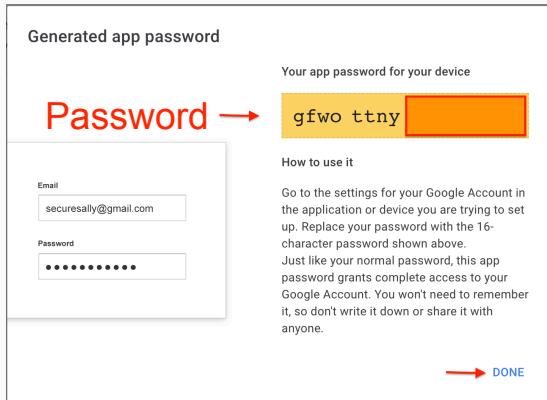
# Jenkins Email Settings

1. Scroll down to Default Content Type
2. Select HTML



# Jenkins Email Settings

1. Scroll down to Email Notification
2. **Server** : smtp.gmail.com
3. **Username** : yourGmail
4. **Password** : Take it from Google
5. Select Use SSL
6. **Port** : 465



E-mail Notification

SMTP server

Default user e-mail suffix ?

Use SMTP Authentication ?

User Name

Password

Use SSL ?

SMTP Port ?



# Jenkins Email Settings

1. Select **Test Configuration**
2. **Email** : noreply@gmail.com
3. Click **Test Configuration**
4. Email was successfully sent !

Test configuration by sending test e-mail

Test e-mail recipient

noreply@gmail.com

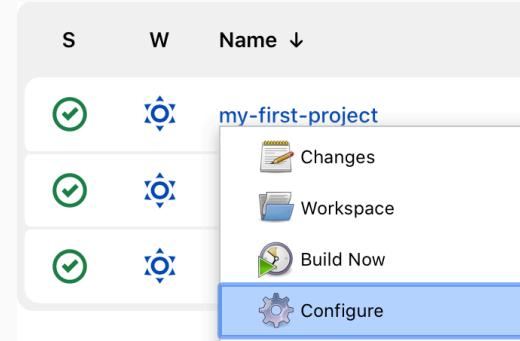
Email was successfully sent

**Test configuration**



# Jenkins Email Settings

1. Click **Configure from any Project**
2. **Scroll down to post-build actions**
3. **Make sure you added Cucumber Report**



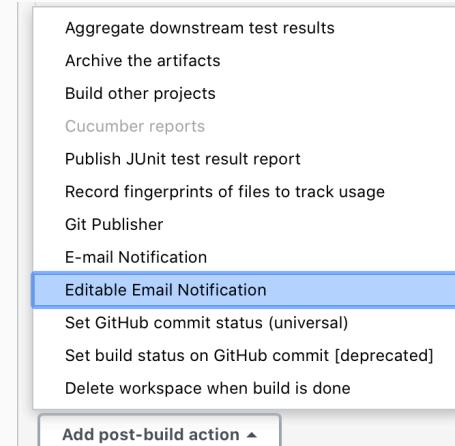
Post-build Actions

Cucumber reports	X
?	Advanced...



# Jenkins Email Settings

1. Click **Add post-build action**
2. Select **Editable Email Notification**
3. Put emails into **Recipient List**
4. Content Type needs to be HTML



A screenshot of a Jenkins input field titled 'Project Recipient List'. The field contains the placeholder text 'put email here'. Below the input field is a descriptive note: 'Comma-separated list of email address that should receive notifications for this project.' A small green circular icon with a white letter 'G' is located in the bottom right corner of the input area.

A screenshot of a Jenkins dropdown menu titled 'Content Type'. The menu is currently showing a single option: 'HTML (text/html)'. A small double-headed vertical arrow icon is located at the bottom right corner of the dropdown menu.

# Jenkins Email Settings

1. Click **Advance Settings**
2. Select Trigger as **Always**
3. **Save**

The screenshot shows the Jenkins Email Settings configuration page. At the top, there are settings for "Attach Build Log" (set to "Do Not Attach Build Log") and "Content Token Reference". A "Advanced Settings..." button is located in the top right corner of this section. Below this, a dropdown menu titled "Add Trigger" is open, listing various triggers: Aborted, Always, Before Build, Failure - 1st, Failure - 2nd, Failure - Any, Failure - Still, Failure - X, Failure -> Unstable (Test Failures), and Fixed. The "Always" option is highlighted with a blue selection bar. At the bottom of the configuration area, there are two buttons: a dark blue "Save" button and a light gray "Apply" button.

