



Week 12 : SOFTWARE DEVELOPMENT TOOLS AND ENVIRONMENTS

Traditional Deployment



Developer



Operations

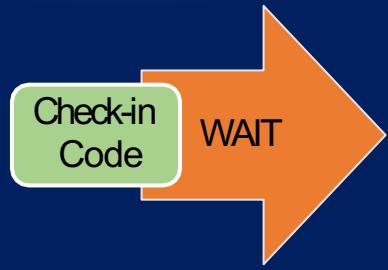
Traditional Deployment



Developer



Operations



Hours/Days + Lot of Grief for Developer & Operations

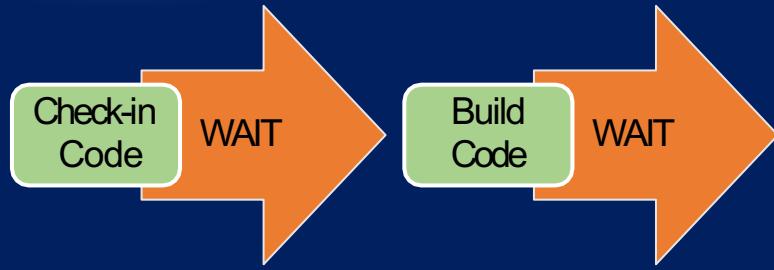
Traditional Deployment



Developer



Operations



Hours/Days + Lot of Grief for Developer & Operations

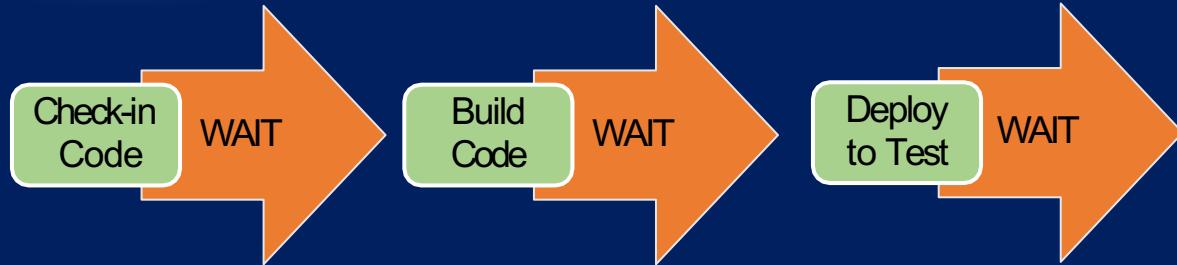
Traditional Deployment



Developer



Operations



Hours/Days + Lot of Grief for Developer & Operations

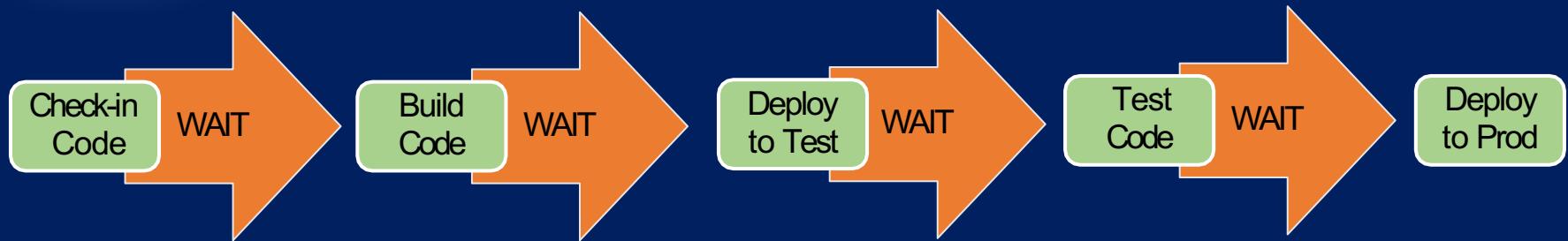
Traditional Deployment



Developer



Operations



Hours/Days + Lot of Grief for Developer & Operations

Traditional Deployment

When are
you gonna
deploy my
code?

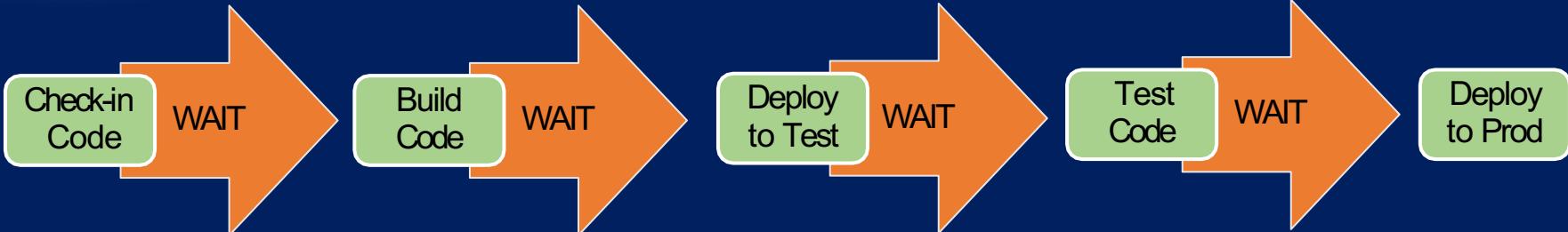


Developer

When you
stop breaking
my servers



Operations





I just
wanna do
cool stuff

SYSTEM

What is DevOps?

- Word “DevOps” coined in 2009 by Patrick Debois
- Combination of cultural philosophies, practices, and tools
 - Job market is based on tools!
- Development and Operations teams are no longer “siloed”



Traditional Deployment

When are
you gonna
deploy my
code?

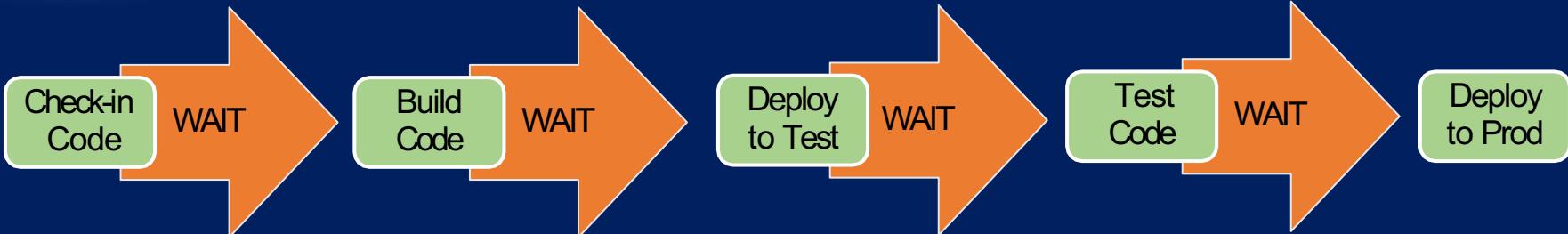


Developer

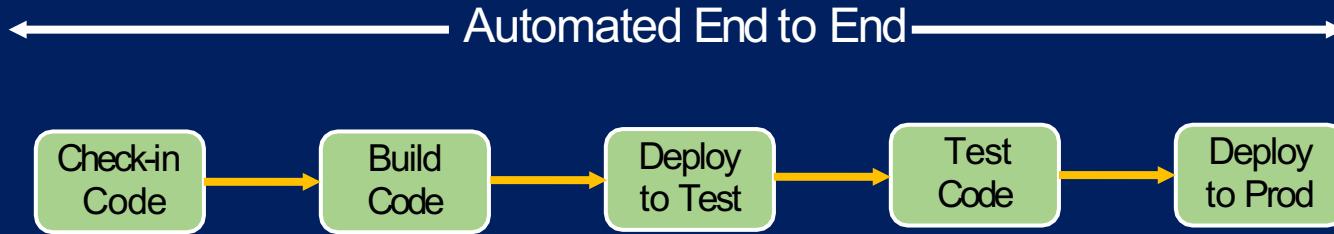
When you
stop breaking
my servers



Operations

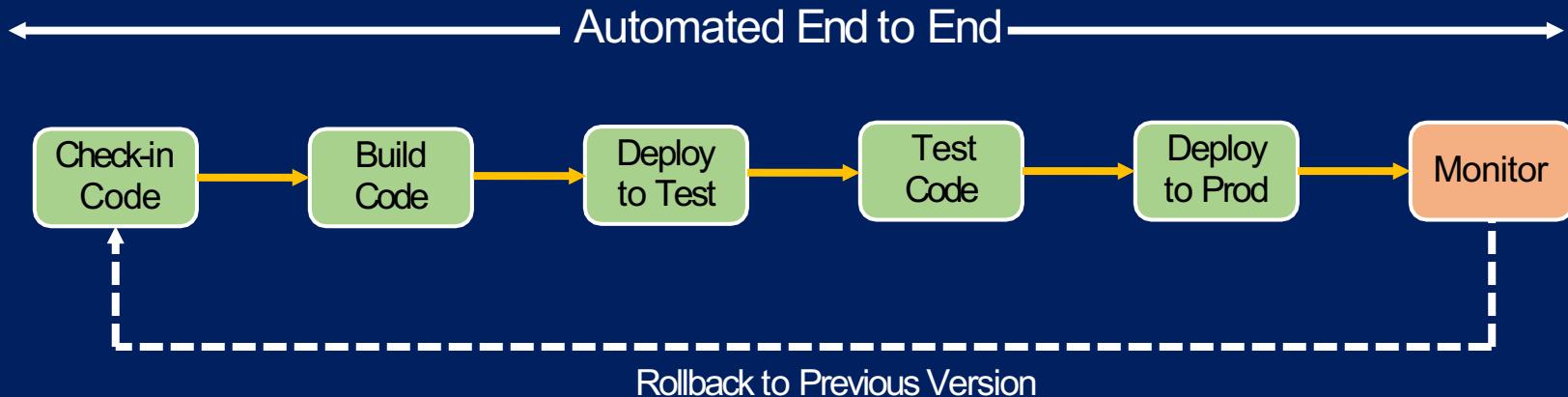


DevOps



- Whole flow done in seconds!
- Easy to rollback in case of errors

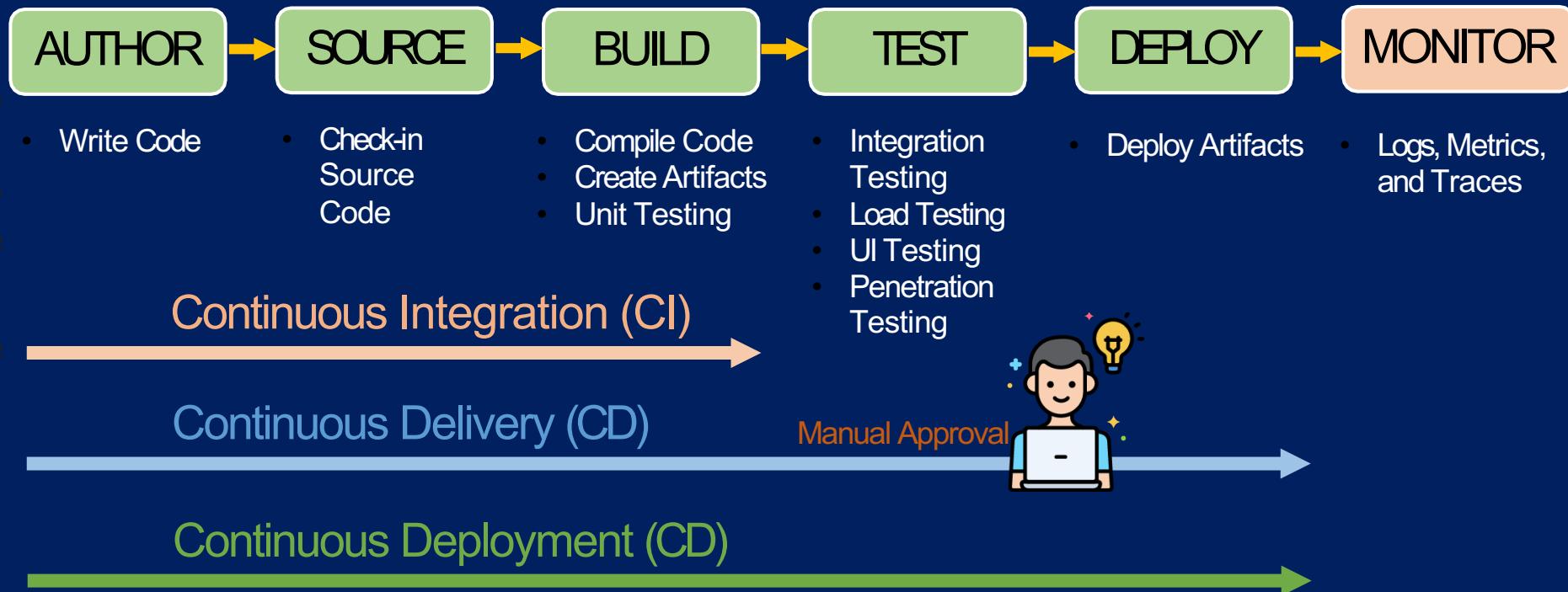
DevOps



General DevOps Practices

- Automate everything!
- Deploy frequently rather than one mega deployment in months
- Codify every step - infrastructure, application and more
- Rome was not built in a day!

DevOps Phases



DevOps Tools

JUnit



pytest



LOCUST



AWS Cloud9



AWS CodeCommit



AWS CodeBuild



AWS CodeBuild



AWS CodeDeploy



CloudWatch



AWS X-Ray

AUTHOR

SOURCE

BUILD

TEST

DEPLOY

MONITOR

- Write Code
- Check-in Source Code
- Compile Code
- Create Artifacts
- Unit Testing
- Integration Testing
- Load Testing
- UI Testing
- Penetration Testing
- Deploy Artifacts
- Logs, Metrics, and Traces



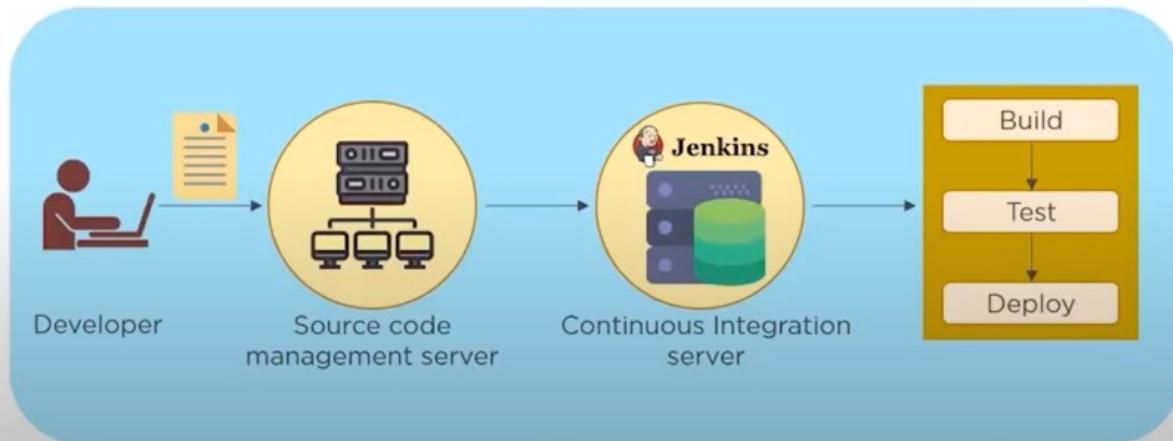
AWS CodePipeline



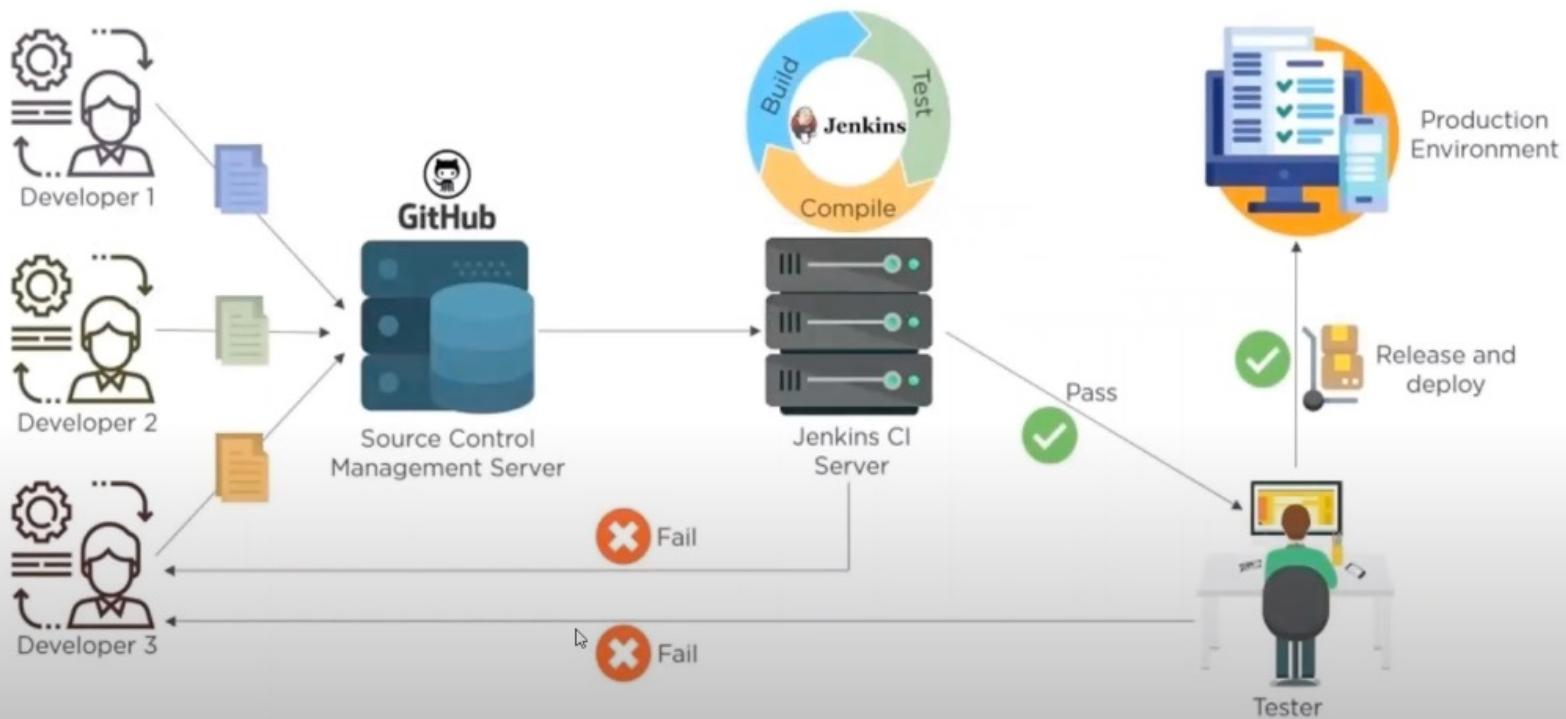
What is Jenkins?



Jenkins is an open source Continuous Integration server written in Java that allows continuous development, test and deployment of codes



CI & CD



Lab 01:

01_1_Jenkins Installation

01_2_Jenkins Plugin Installation

01_1_Jenkins Installation

Installation Steps ↗

1. Update System Packages Update the list of available packages and their versions.

```
sudo apt-get update
```

2. Installation of Java

```
sudo apt update
sudo apt install fontconfig openjdk-17-jre
java -version
openjdk version "17.0.8" 2023-07-18
OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)
```

3. Long Term Support release

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```

4. Start Jenkins

```
sudo systemctl enable jenkins
```

```
sudo systemctl start jenkins
```

```
sudo systemctl status jenkins
```

LAB 02 : Jenkins Add Credentials

1.Add SSH-Credentials

Jenkins

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

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

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

1

Security

2

Credentials

Configure credentials

Credential Providers

Configure the credential providers and types

Status Information

System Information

Displays various environmental information to assist troubleshooting.

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.

About Jenkins

See the version and license information.

Jenkins

Search (⌘+K)

Dashboard > Manage Jenkins > Credentials

Credentials

T	P	Store ↓	Domain	ID	Name
---	---	---------	--------	----	------

Stores scoped to Jenkins

P	Store ↓	Domains
System		(global)

Icon: S M L

3

Jenkins

Search (⌘+K)

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted)

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
This credential domain is empty. How about adding some credentials?			

Icon: S M L

4

Jenkins

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind: **SSH Username with private key** (circled)

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: **ssh-prod_instance** (circled)

Description: (empty)

Username: **tuchsanai** (circled)

Treat username as secret (circled)

PRIVATE KEY (Large red box around this section)

Enter directly:

Key:

```
-----BEGIN OPENSSH PRIVATE KEY-----  
-----END OPENSSH PRIVATE KEY-----
```

 (highlighted with a blue border)

Passphrase: (empty)

Create

For SSH

Jenkins

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
ssh-prod_instance	ssh-prod_instance	SSH Username with private key	

Icon: S M L (circled)

Compute Engine

VM instances

INSTANCES OBSERVABILITY INSTANCE SCHEDULES

VM instances

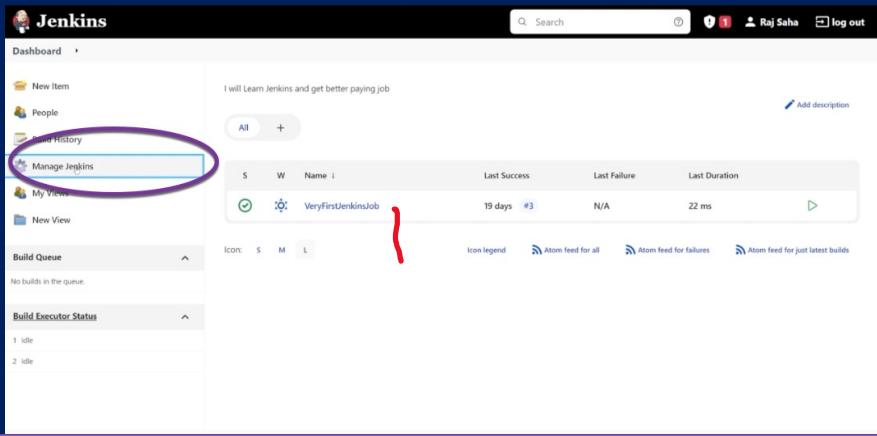
Filter: Enter primary name or value

Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	jenkins	asia-southeast1-a		10.148.0.15 (nic0)	34.126.119.107 (nic0)	SSH	⋮
□	prod-instance	asia-southeast1-b		10.148.0.16 (nic0)	34.143.151.212 (nic0)	SSH	⋮

Related actions:

- Explore Backup and DR (NEW)
- View billing report
- Monitor VMs
- Explore VM logs

2.Add Github-Credentials



Jenkins Dashboard

I will Learn Jenkins and get better paying job

Manage Jenkins

VeryFirstJenkinsJob

Build Queue

Build Executor Status

Security

Configure Global Security

Manage Users

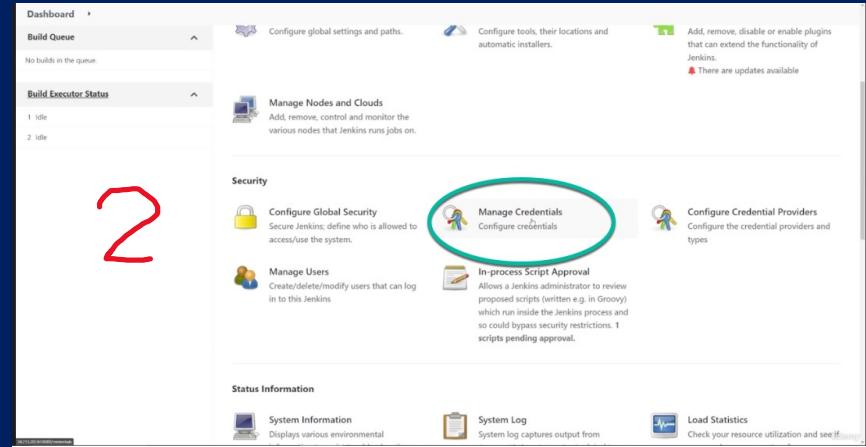
In-process Script Approval

Status Information

System Information

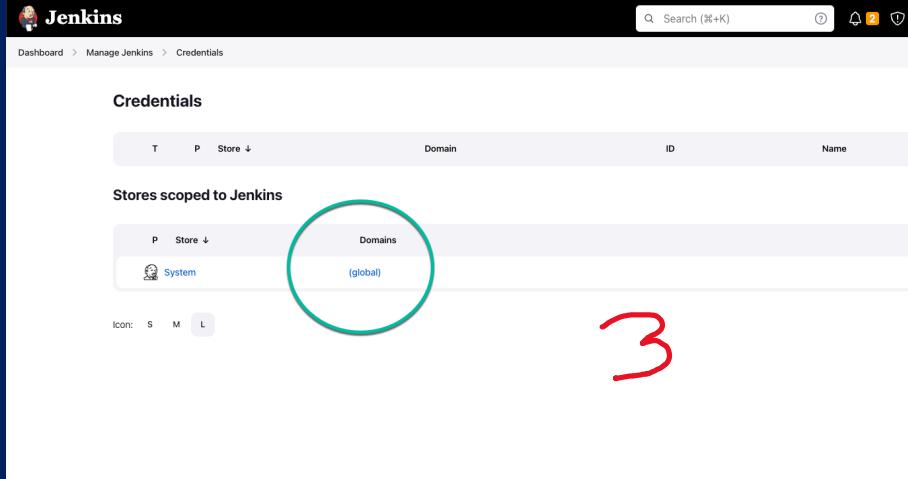
System Log

Load Statistics



2

Manage Credentials



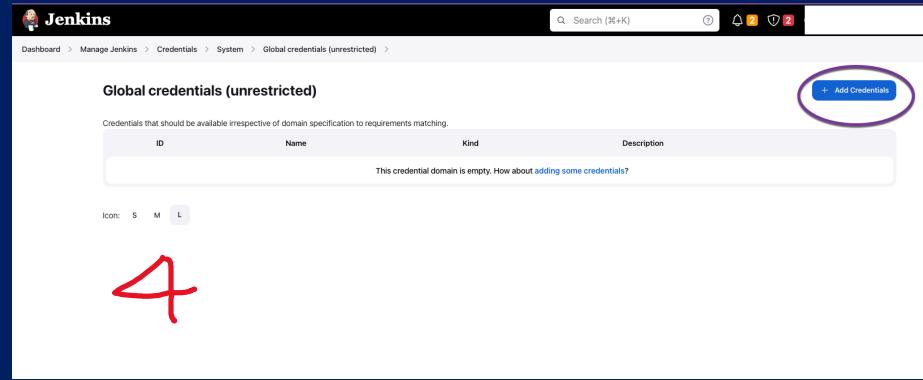
Credentials

Stores scoped to Jenkins

Domains

(global)

3



Global credentials (unrestricted)

+ Add Credentials

ID Name Kind Description

This credential domain is empty. How about adding some credentials?

4

For Github

New credentials

1. **Id**: Username with password

2. **Username**: tuchsanai

3. **Password**:

4. **ID**: github

Description: my github for test jenkins

Settings / Developer settings

Personal access tokens (classic)

Make sure to copy your personal access token now. You won't be able to see it again!

ghp_Zp7azz7td4qxUcZdLyTT8PHG1Ljri0t3Fz

Personal access tokens (classic) | Jenkins | OAuth access tokens

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

repo

- repo:status
- repo_deployment
- public_repo
- repo:invite
- security_events

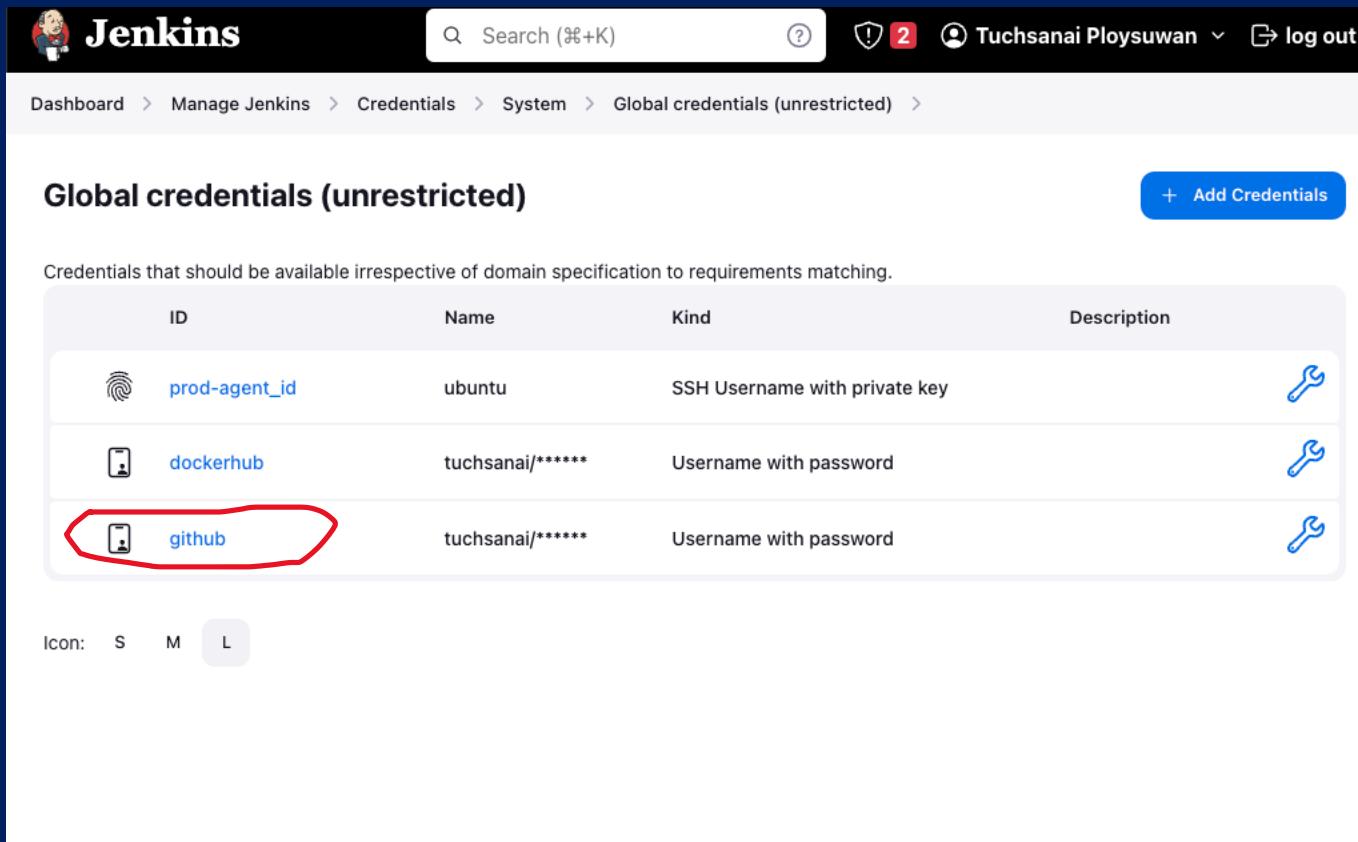
Full control of private repositories
Access commit status
Access deployment status
Access public repositories
Access repository invitations
Read and write security events

admin:repo_hook

- write:repo_hook
- read:repo_hook

Full control of repository hooks
Write repository hooks
Read repository hooks

For Github



The screenshot shows the Jenkins Global credentials (unrestricted) page. The navigation path is: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted). The page title is "Global credentials (unrestricted)". A blue button on the right says "+ Add Credentials". Below the title, a note says "Credentials that should be available irrespective of domain specification to requirements matching." A table lists three credentials:

ID	Name	Kind	Description
	prod-agent_id	ubuntu	SSH Username with private key
	dockerhub	tuchsanai/********	Username with password
	github	tuchsanai/********	Username with password

At the bottom left, there are icons for S, M, and L, with L being highlighted.

3.Add docker hub Credentials

Jenkins Dashboard

I will Learn Jenkins and get better paying job

Manage Jenkins

S	W	Name	Last Success	Last Failure	Last Duration
Green	Yellow	VeryFirstJenkinsJob	19 days #3	N/A	22 ms

Manage Jenkins

VeryFirstJenkinsJob

Build Queue

Build Executor Status

Dashboard

Build Queue

Build Executor Status

Manage Nodes and Clouds

Manage Credentials

Configure Global Security

Manage Users

In-process Script Approval

Security

Status Information

System Information

System Log

Load Statistics

Configure Credential Providers

2

Jenkins

Dashboard > Manage Jenkins > Credentials

Credentials

T	P	Store	Domain	ID	Name
System	(global)				

Stores scoped to Jenkins

Domains

(global)

Icon: S M L

3

Jenkins

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

+ Add Credentials

ID	Name	Kind	Description
			This credential domain is empty. How about adding some credentials?

4

The screenshot shows the Docker Hub homepage. A large blue banner on the left features the text "Build and Ship any Application Anywhere". Below the banner, it says "Docker Hub is the world's easiest way to create, manage, and deliver your team's container applications." A red hand-drawn number "5" is written over the blue banner area. At the bottom, it says "Docker Hub is the world's largest library and community for container images". The main navigation bar at the top includes "Explore", "Pricing", "Sign In", and "Sign up". A search bar says "Search Docker Hub".

Welcome back
Sign in to Docker

Username or email address

Sign in

Don't have an account? Sign up

The screenshot shows the Docker Hub user profile page for the user "tuchsanai". The top navigation bar includes "Explore", "Repositories", "Organizations", "Search Docker Hub", and a user icon with a red hand-drawn letter "T". A red hand-drawn letter "b" is written near the bottom left of the page. The sidebar on the right has links for "What's New", "My Profile", "My Account" (which is circled in red), and "Billing". It also shows a "Sign out" button. The main content area lists three repositories: "tuchsanai / pytorch_jupyterlab_ubuntu22.04", "tuchsanai / kubirstapp", and "tuchsanai / custom-nginx-registry".

tuchsanai / pytorch_jupyterlab_ubuntu22.04
Contains: Image | Last pushed: 2 months ago

tuchsanai / kubirstapp
Contains: Image | Last pushed: 3 months ago

tuchsanai / custom-nginx-registry
Contains: Image | Last pushed: 9 months ago

What's New

My Profile

My Account

Billing

Sign out

Community All-Hands: On-Demand

All sessions from our 6th Community All-Hands are now available on-demand! Over 35 talks cover best practices

The screenshot shows the Docker Hub account settings under the 'Security' tab. The user profile is displayed with a blue fingerprint icon and the name 'tuchsanai'. A red circle highlights the profile area. Below the profile, there's a section titled 'Access Tokens' with a message: 'It looks like you have not created any access tokens. Docker Hub lets you create tokens to authenticate access. Treat personal access tokens as alternatives to your password.' A red circle highlights the 'New Access Token' button. At the bottom, there's a 'Two-Factor Authentication' section with a message: 'Two-factor authentication is not enabled yet. Two-factor authentication adds an extra layer of security to your account by requiring more than just a password to sign in.' A red circle highlights the 'Enable Two-Factor Authentication' button.

The screenshot shows a 'Copy Access Token' dialog box. It contains the following text:
ACCESS TOKEN DESCRIPTION
a
ACCESS PERMISSIONS
Read, Write, Delete
To use the access token from your Docker CLI client:
1. Run `docker login -u [REDACTED]`
2. At the password prompt, enter the personal access token.
A text input field contains the token: `dckr_pat_tgAISGZbNsN5LuR4ckSI6Zb6c8`. A red circle highlights the input field. A warning message below says: 'WARNING: This access token will only be displayed once. It will not be stored and cannot be retrieved. Please be sure to save it now.' A red circle highlights the warning message. At the bottom right is a 'Copy and Close' button.

The screenshot shows the Docker Hub account settings under the 'Security' tab. The user profile is displayed with a blue fingerprint icon and the name 'tuchsanai'. Below the profile, there's a table titled 'Access Tokens' showing one entry: 'mydockhub' (MANUAL), Read, Write, Delete, Never, Jan 04, 2024, 20:48:55. A red circle highlights the table row. At the bottom, there's a 'Two-Factor Authentication' section with a message: 'Two-factor authentication is not enabled yet. Two-factor authentication adds an extra layer of security to your account by requiring more than just a password to sign in.' A red circle highlights the 'Enable Two-Factor Authentication' button.

Jenkins

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: tuchsanai

Treat username as secret

Password:

ID: dockerhub

Description:

Create

Jenkins

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
 prod-agent_id	ubuntu	SSH Username with private key	
 dockerhub	tuchsanai*****	Username with password	

Icon: S M L



Jenkins

Search (%+K)



Tuchsanai Ploysawan

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description	
github	tuchsanai/******** (my github for test jenkins)	Username with password	my github for test jenkins	
dockerhub	tuchsanai/******** (my docker hub)	Username with password	my docker hub	

Icon: S M L

LAB 3 : First Jenkins



Jenkins

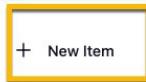
Search (⌘+K)



e

log out

Dashboard >



1



Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

(offline)

Add description

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





Enter an item name

first_project

» Required field

2

**Freestyle project**

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

**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.

3

**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.

**Multibranch Pipeline**

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

**Organization Folder**

Creates a set of multibranch project subfolders by scanning for repositories.

Configure

General

Advanced Project Options

Pipeline

Throttle builds ?

Build Triggers

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

Advanced Project Options

Advanced ▾

4

Pipeline

Definition

Pipeline script

Script ?

```
1~ pipeline {  
2~   agent any // Execute on any available Jenkins agent  
3~  
4~   stages {  
5~     stage('Hello World') {  
6~       steps {  
7~         echo 'Hello World!'  
8~       }  
9~     }  
10~   }  
11~ }  
12~  
13~
```

try sample Pipeline... ▾

Use Groovy Sandbox ?

Dashboard > first_project >

 Status

</> Changes

 Build Now

5

 Configure Delete Pipeline Full Stage View Rename Pipeline Syntax

first_project

 Add description[Disable Project](#)

Stage View

No data available. This Pipeline has not yet run.

Permalinks

 Build Historytrend ▾

No builds

 Atom feed for all  Atom feed for failures

 Status

first_project

</> Changes

 Add description

▷ Build Now

 Disable Project Configure Delete Pipeline Full Stage View Rename Pipeline Syntax Build History

trend ▾

 Filter...

/

 #1

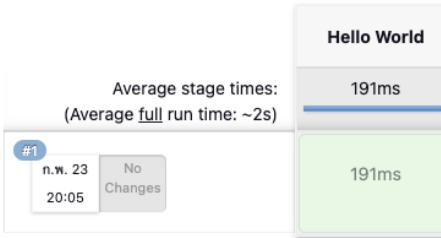
23 n.w. 2024 13:05

Permalinks

6

Stage View

Average stage times:
(Average full run time: ~2s)



 Status

7

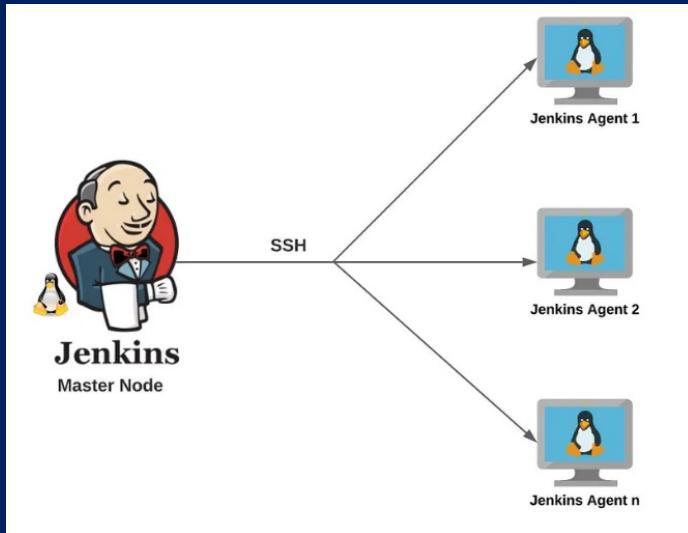
</> Changes

 Console Output View as plain text Edit Build Information Delete build '#1' Restart from Stage Replay Pipeline Steps Workspaces

Console Output

```
Started by user Tuchsanai Ploysawan
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/first_project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello World)
[Pipeline] echo
Hello World!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```


LAB x : Jenkins SSH-Agent



The screenshot shows the Google Cloud Compute Engine interface for managing virtual machines. The left sidebar menu includes "Compute Engine", "Virtual machines", "VM instances" (which is selected and highlighted in blue), "Instance templates", "Sole-tenant nodes", "Machine images", "TPUs", "Committed use discounts", "Reservations", and "Migrate to Virtual Machine...". The main content area is titled "VM instances" and displays a table of running instances:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
Green circle	jenkins	asia-southeast1-a			10.148.0.15 (nic0)	34.126.119.107 (nic0)	SSH
Green circle	prod-instance	asia-southeast1-b			10.148.0.16 (nic0)	34.143.151.212 (nic0)	SSH

Below the table, under "Related actions", are four buttons: "Explore Backup and DR", "View billing report", "Monitor VMs", and "Explore VM logs".

Jenkins

Search (%+K) [?](#) [2](#) [Tuchsanai Ploysuwan](#) [log out](#)

Dashboard > All >

Enter an item name

ssh-agent 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.

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

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Copy from
Type to autocomplete

OK

← → 🔍 🏠 ⚡ Not Secure 54.151.253.215:8080/job/ssh-agent/configure

Dashboard > ssh-agent > Configuration

Configure

General

GitHub project

Pipeline speed/durability override

Preserve stashes from completed builds

This project is parameterized

Throttle builds

Build Triggers

Build after other projects are built

Build periodically

GitHub hook trigger for GITScm polling

Poll SCM

Quiet period

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

Advanced Project Options

Advanced

Pipeline

Definition

Pipeline script

```
Script ?  
1 stage('Apply multiple Commands') {  
2   steps {  
3     sshagent(['prod-agent_id']) {  
4       sh ***  
5         ssh -o StrictHostKeyChecking=no remote_user@remote_host  
6         echo First command;  
7         echo Second command;  
8         echo Third command  
9     }  
10    ...  
11  }  
12 }  
13 // Replace remote_user, remote_host, and the commands with your own.  
14  
try sample Pipeline...  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29
```

Use Groovy Sandbox

Pipeline Syntax

Save Apply

```

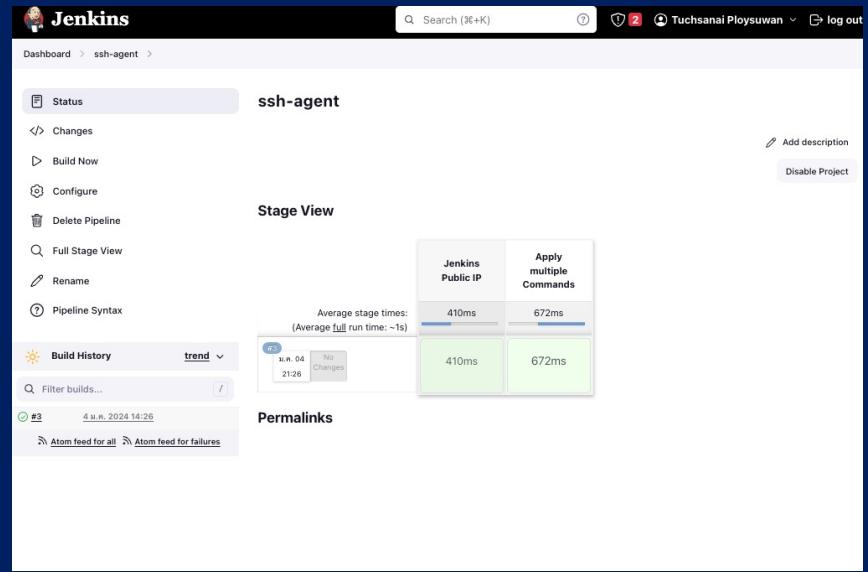
pipeline {
    agent any

    stages {
        stage('Jenkins Public IP') {
            steps {
                script {
                    // Use a shell command to get the public IP address
                    def ip = sh(script: 'curl -s ifconfig.me', returnStdout: true).trim()
                    echo "Jenkins Public IP Address: ${ip}"
                }
            }
        }

        stage('Apply multiple Commands') {
            steps {
                sshagent(['agent_id']) {
                    sh """
                        ssh -o StrictHostKeyChecking=no remote_user@remote_host
                        echo First command;
                        echo Second command;
                        echo Third command
                    """
                }
            }
        }
    }
}

```

// Replace remote_user, remote_host, and the commands with your own.



Jenkins

Dashboard > ssh-agent > #3

Status Changes Console Output View as plain text Edit Build Information Delete build #3 Restart from Stage Replay Pipeline Steps Workspaces

Started by user Tushanai Playswan
[Pipeline] Start of Pipeline
[Pipeline] stage
[Pipeline] {
[Pipeline] stage
[Pipeline] { Jenkins Public IP
[Pipeline] script
[Pipeline] {
+ curl -s lftconfig.me
[Pipeline] echo
Jenkins Public IP Address: 54.151.253.215
[Pipeline] }
[Pipeline] // script
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { Apply multiple Commands
[Pipeline] {
[sh-agent] Using credentials ubuntu
[sh-agent] Looking for ssh-agent implementation...
[sh-agent] Exec ssh-agent (binary ssh-agent on a remote machine)
\$ ssh -o StrictHostKeyChecking=ubuntu18.143.133.167
\$ SSH_AUTH_SOCK=/tmp/ssh-XXXXXXgb4dNA/agent.15229
SSH_AGENT_PID=15232
Running ssh-add (command line suppressed)
Identity added: /var/lib/jenkins/workspace/ssh-agent@tmp/private_key_1758830963968727163.key (/var/lib/jenkins/workspace/ssh-agent@tmp/private_key_1758830963968727163.key)
[sh-agent] Started.
[Pipeline] }
[Pipeline] }
+ ssh -o StrictHostKeyChecking=ubuntu18.143.133.167
Pseudo-terminal will not be allocated because stdn is not a terminal.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

* Documentation: <https://help.ubuntu.com>
* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/adantage>

System information as of Thu Jan 4 14:26:28 UTC 2024

System load: 0.0 Processes: 188
Usage of /: 19.5% of 9.51GB Users logged in: 1
Memory usage: 6% IPv4 address for eth0: 172.31.37.159
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
compliance features.

<https://ubuntu.com/ubuntu/pro>

Expanded Security Maintenance for Applications is not enabled.
29 updates can be applied immediately.
29 of these updates are standard security updates.
To get these additional updates run: apt list --upgradable

Enable ESP Apps to receive additional future security updates.
See <https://ubuntu.com/esp> or run: sudo pro status

+ echo First command
First command
+ echo Second command
Second command
+ echo Third command
Third command
[Pipeline] {
+ curl -s lftconfig.me
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 15232 killed;
[sh-agent] Stopped.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

Finished: SUCCESS

LAB : How to Change Port for Jenkins?



<http://localhost:8080>



<http://localhost:5000>

HTTP Port (default: 8080) – This is the main port used by Jenkins to serve web pages on your machine. By default, this port is set to 8080 but can be changed based on your requirements.

If Jenkins fails to start because a port is in use, run `sudo systemctl edit jenkins` and add the following:

1. Override the Jenkins Service Configuration

```
sudo systemctl edit jenkins
```



2. Add or Modify the ExecStart Command

```
[Service]  
Environment="JENKINS_PORT=5000"
```



3. Reload the Systemd Daemon

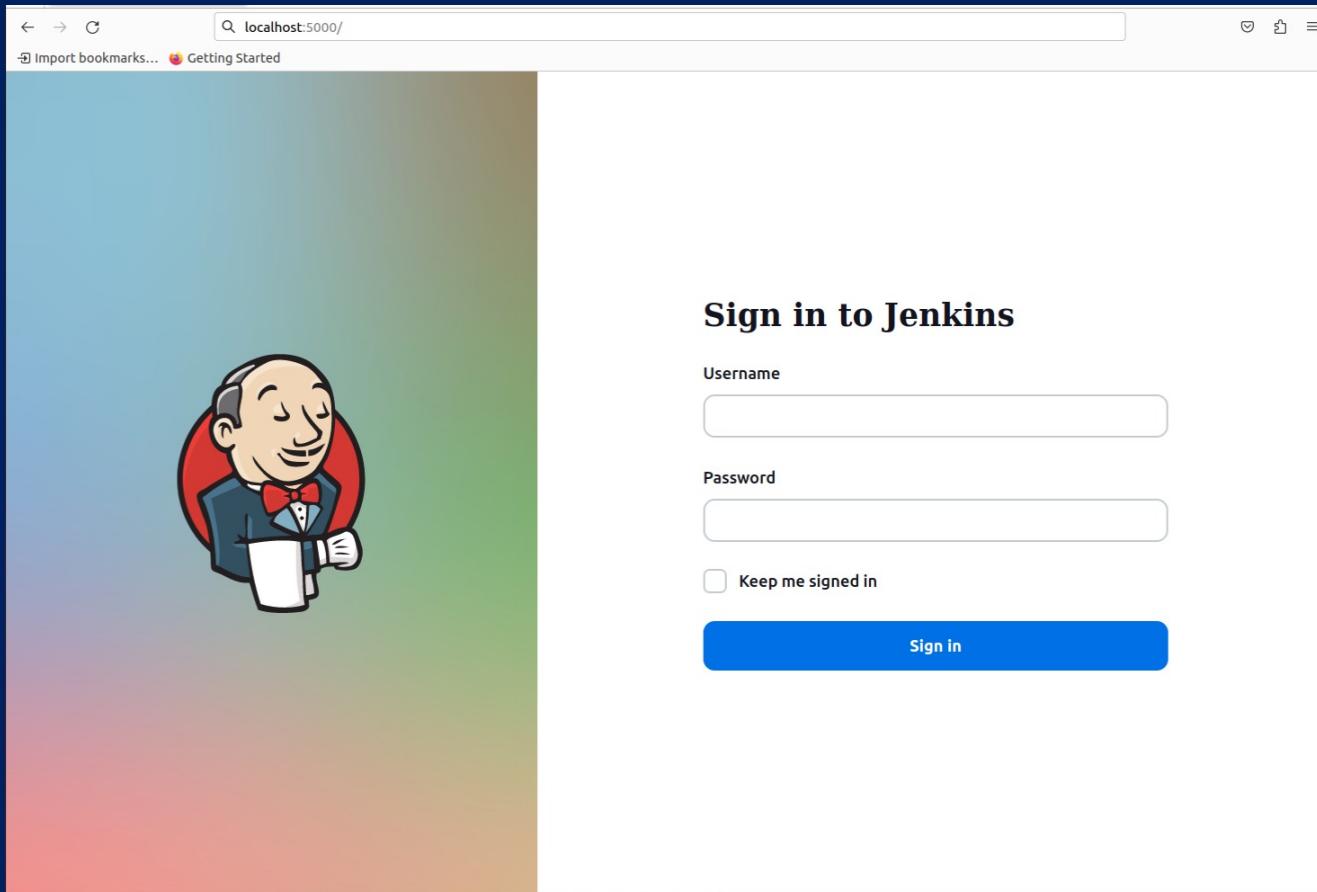
```
sudo systemctl daemon-reload
```



4. Restart the Jenkins service to apply the changes

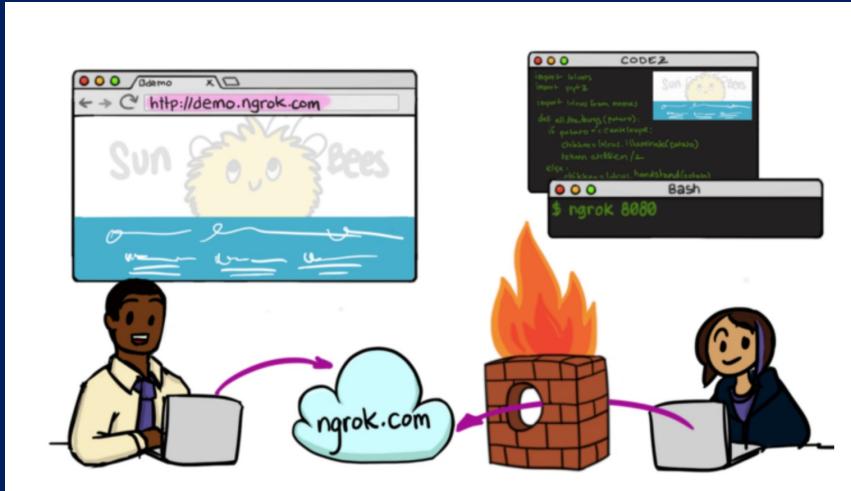
```
sudo service jenkins restart
```





LAB : Forward my local port to public using ngrok

What is ngrok?



ngrok exposes local servers behind NATs and firewalls to the public internet over secure tunnels.

Advantages

- Demoing web sites without deploying
- Building webhook consumers on your dev machine
- Testing mobile apps connected to your locally running backend
- Stable addresses for your connected devices that are deployed in the field
- Running personal cloud services from your home

The screenshot shows the official website for ngrok. At the top, there's a purple header bar with browser navigation icons (back, forward, search, etc.) and the URL "ngrok.com". Below this is a dark blue navigation bar with the "ngrok" logo on the left and links for "Product", "Solutions", "Customers", "Docs", "Pricing", and "Download". On the right side of the navigation bar are two buttons: "Login" (white background) and "Sign up" (blue background). A red rectangular box highlights these two buttons. The main content area features a large, bold heading: "Enrich Requests with Geo Location with one command". Below the heading is a descriptive text: "ngrok is a simplified API-first ingress-as-a-service that adds connectivity, security, and observability to your apps in one line". Underneath this text is a blue button labeled "Sign up for free". At the bottom of the page is a dark gray terminal-style box containing a command-line snippet: "[user@localhost]\$ | ngrok http 80 --request-header-add "country: \${ngrok.geo.country_code}"]. To the right of this box is a "Copy!" button.

Introducing always-on global server load balancing ->

ngrok

Product Solutions Customers Docs Pricing Download

Login Sign up

Enrich Requests with Geo Location with one command

ngrok is a simplified API-first ingress-as-a-service that adds connectivity, security, and observability to your apps in one line

Sign up for free

```
[user@localhost]$ | ngrok http 80 --request-header-add "country: ${ngrok.geo.country_code}"]
```

Copy!

Getting Started

Setup & Installation

Your Authtoken

Cloud Edge

Tunnels

Events

API

Security

Users

Billing

Settings

Your Authtoken

This is your personal Authtoken. Use this to authenticate the ngrok agent that you downloaded.

Pyzz2017v
Copy

Command Line

Authenticate your ngrok agent. You only have to do this once. The Authtoken is saved in the default configuration file.

```
$ ngrok config add-authtoken 
```

Configuration File

Alternatively, you can directly add the Authtoken to your `ngrok.yml` configuration file. Use `ngrok config edit` to open the file.

```
# in ngrok.yml
authtoken: 
```

Reset Your Authtoken

To setup ngrok on Ubuntu 22.04, there are two primary methods you can follow. [♂](#)

1. Create an Ngrok Account:

- If you don't have an ngrok account, create one on the [official ngrok website](#).

2. Download ngrok:

- You can download ngrok directly from the Setup & Installation page on the ngrok dashboard or use the following command in the terminal to download the binary file:

```
wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
```



3. Decompress & Install Ngrok Tunnel:

- Extract the downloaded file with the following command:

```
tar zxvf ngrok-v3-stable-linux-amd64.tgz
```



4. Connect Ubuntu to the ngrok account:

- Run the following command to add the auth token to the default Ngrok.yml configuration file (replace the token with your own):

```
./ngrok config add-authtoken <YourAuthToken>
```



5. Usage:

- To create a tunnel to your local server or port forwarding using ngrok, use the following command (replace 8080 with your desired port):

```
./ngrok http 8080
```



- This will provide you with a URL accessible on any device having internet connectivity. The URL of your local Web Interface will be like "http://127.0.0.1:8080," and the URL provided by ngrok to access this web server worldwide will be something like "<https://in.ngrok.io>" .

```
ngrok
Introducing Always-On Global Server Load Balancer: https://ngrok.com/r/gslb

Session Status          online
Account                lucasmai@gmail.com (Plan: Free)
Version                3.3.5
Region                 Asia Pacific (ap)
Latency                39ms
Web Interface          http://127.0.0.1:4040
Forwarding             https://8bc7-171-96-174-205.ngrok-free.app -> http://localhost:8080

Connections            ttl     opn     rti1    rt5      p50      p90
                        2       0       0.01   0.00   30.64   30.92

HTTP Requests
-----
GET /static/e6b141b3/favicon.ico           200 OK
GET /login                                200 OK
GET /static/e6b141b3/images/svg/logo.svg   200 OK
GET /static/e6b141b3/jsbundles/simple-page.css 200 OK
GET /                                         403 Forbidden
```

