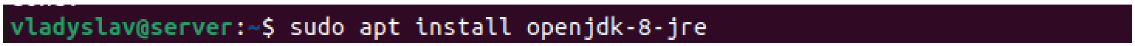
Jenkins

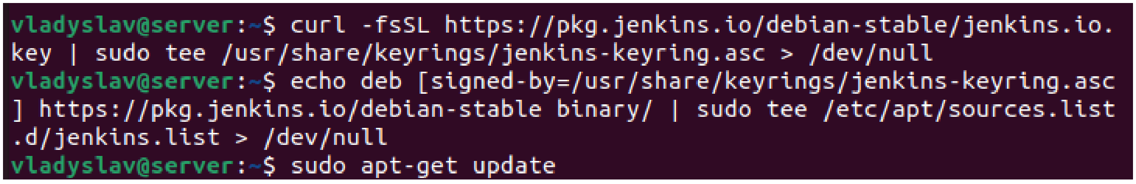
1) I install Jenkins on a virtual machine (Ubuntu distribution).

Installation commands:

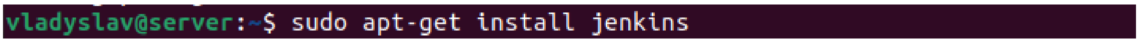
```
sudo apt update
sudo apt install openjdk-11-jre
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```



```
vladyslav@server:~$ sudo apt install openjdk-8-jre
```

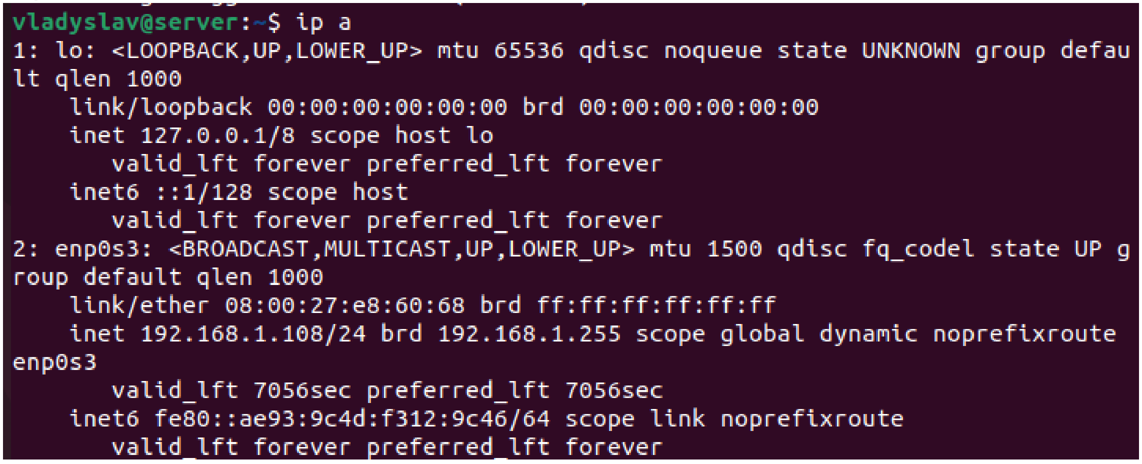


```
vladyslav@server:~$ curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.
key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
vladyslav@server:~$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc
] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list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

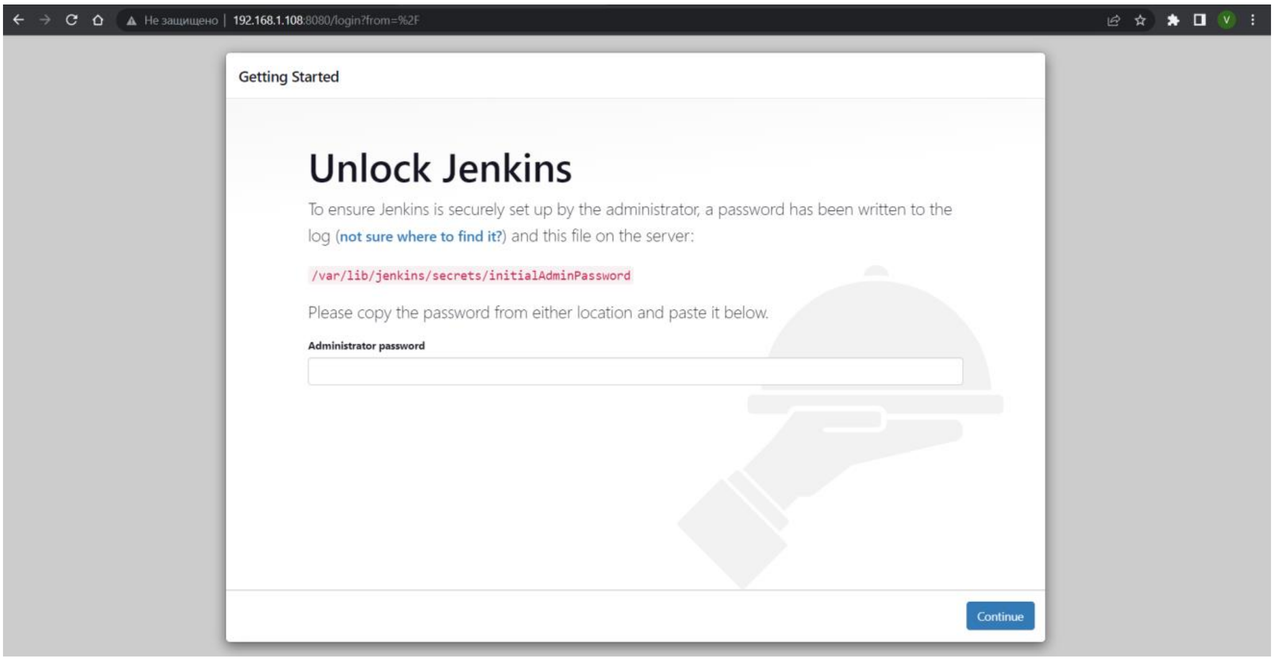


```
vladyslav@server:~$ sudo apt-get install jenkins
```

2) I get the IP address of the machine and check the correctness of the installation.



```
vladyslav@server:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:e8:60:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.108/24 brd 192.168.1.255 scope global dynamic noprefixroute
enp0s3
       valid_lft 7056sec preferred_lft 7056sec
    inet6 fe80::ae93:9c4d:f312:9c46/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```



Getting Started

Unlock Jenkins

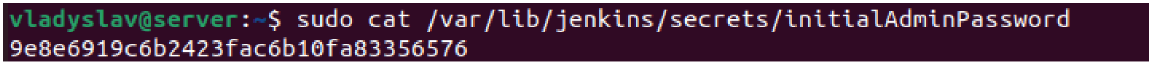
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

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

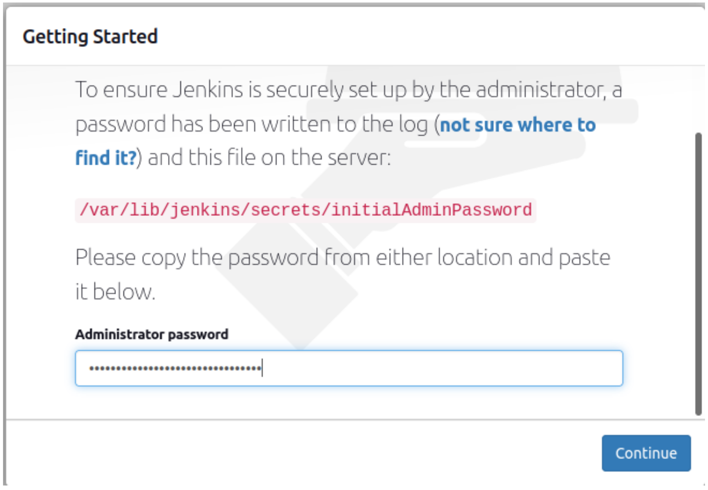
Please copy the password from either location and paste it below.

Administrator password

3) I find out a password and unlock Jenkins.



```
vladyslav@server:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
9e8e6919c6b2423fac6b10fa83356576
```



Getting Started

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:

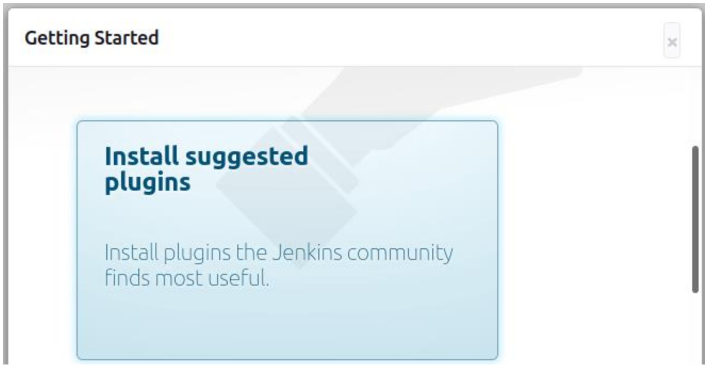
`/var/lib/jenkins/secrets/initialAdminPassword`

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

Administrator password

••••••••••••••••••••••••••••••••

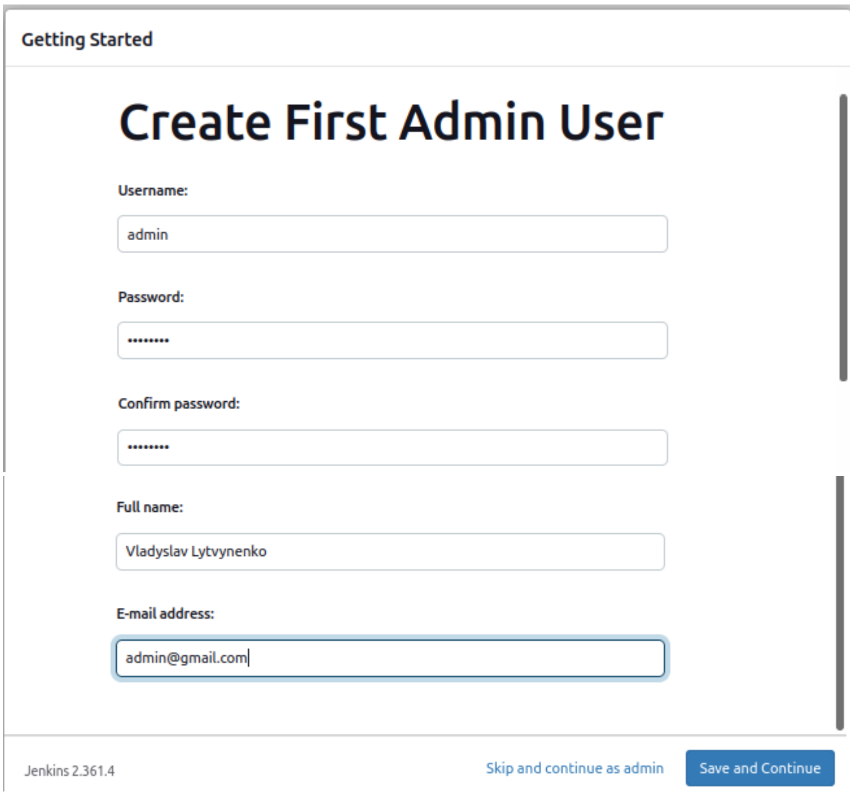
4) I execute basic setup.



Getting Started

Install suggested plugins

Install plugins the Jenkins community finds most useful.



Getting Started

Create First Admin User

Username: admin

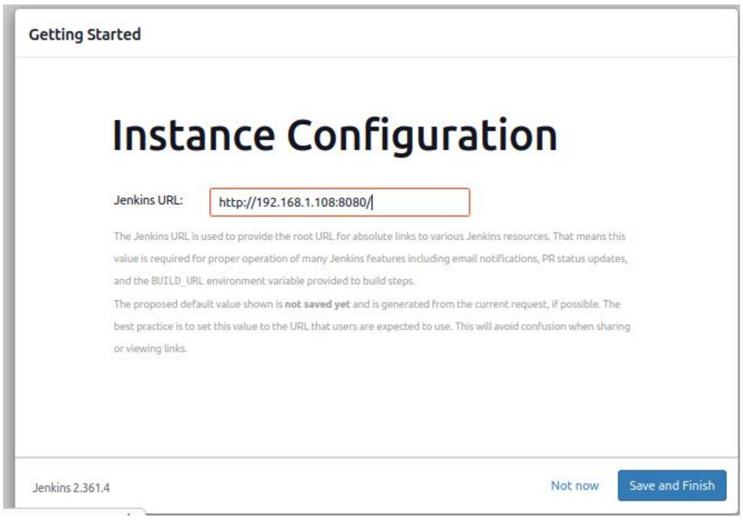
Password: ••••••••

Confirm password: ••••••••

Full name: Vladyslav Lytvynenko

E-mail address: admin@gmail.com

Jenkins 2.361.4



Getting Started

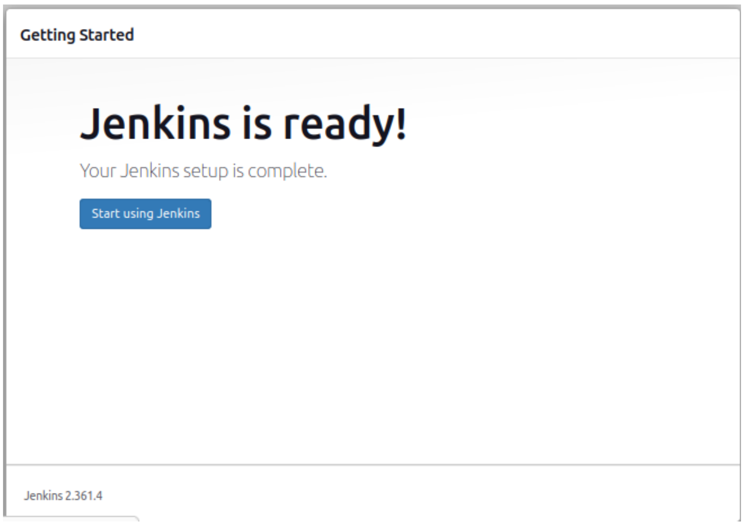
Instance Configuration

Jenkins URL: http://192.168.1.108:8080/

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.

Jenkins 2.361.4



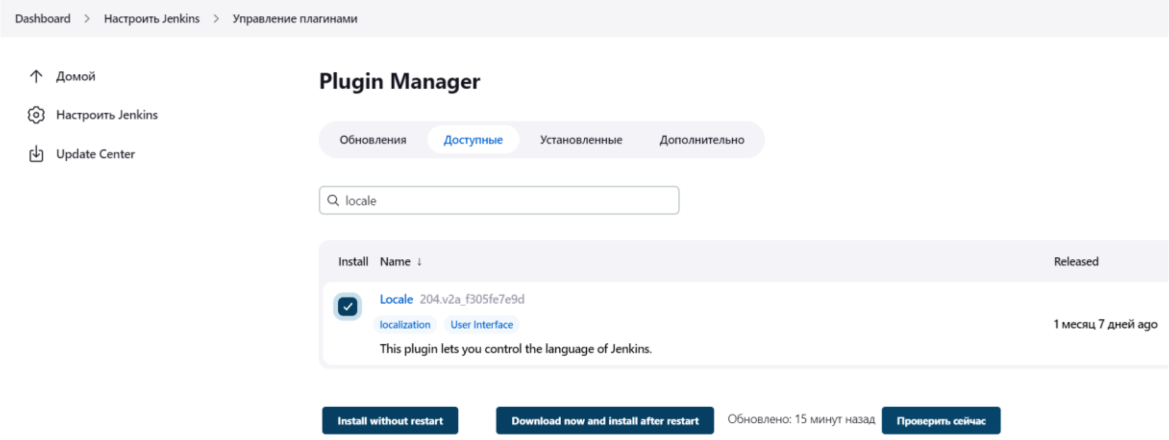
Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Jenkins 2.361.4

5) I install and configure a plugin on Jenkins.



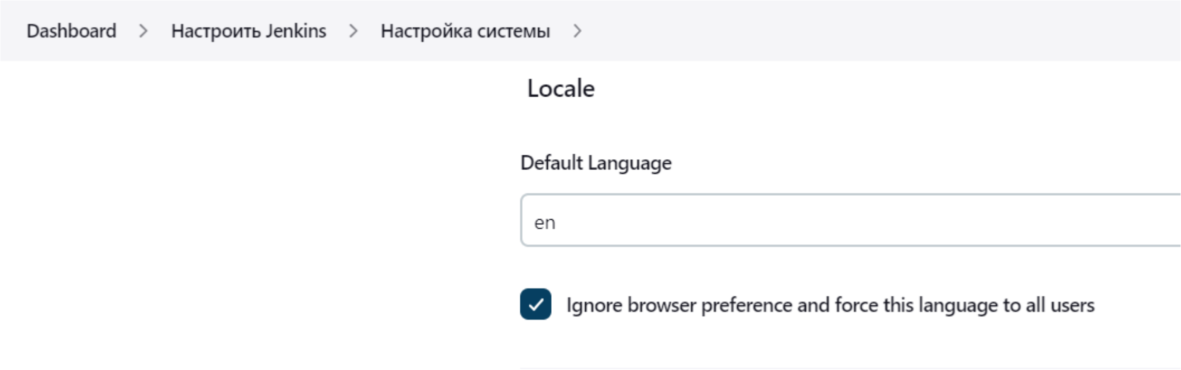
Dashboard > Настроить Jenkins > Управление плагинами

Plugin Manager

locale

Install	Name ↓	Released
☑	Locale 204.v2a_f305fe7e9d localization User Interface This plugin lets you control the language of Jenkins.	1 месяц 7 дней ago

Обновлено: 15 минут назад



Dashboard > Настроить Jenkins > Настройка системы >

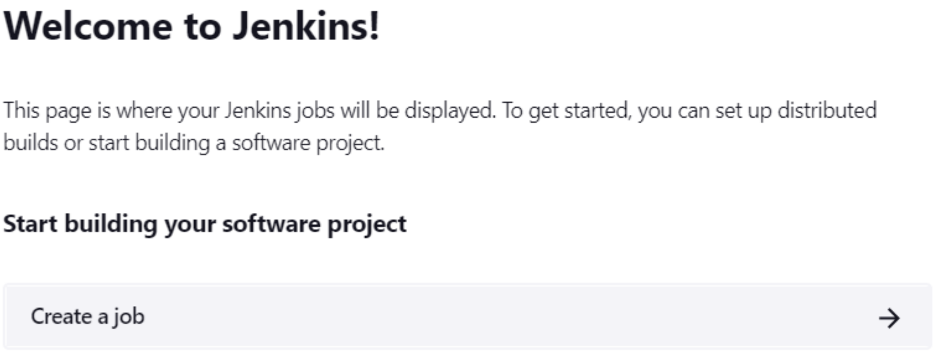
Locale

Default Language

en

☑ Ignore browser preference and force this language to all users

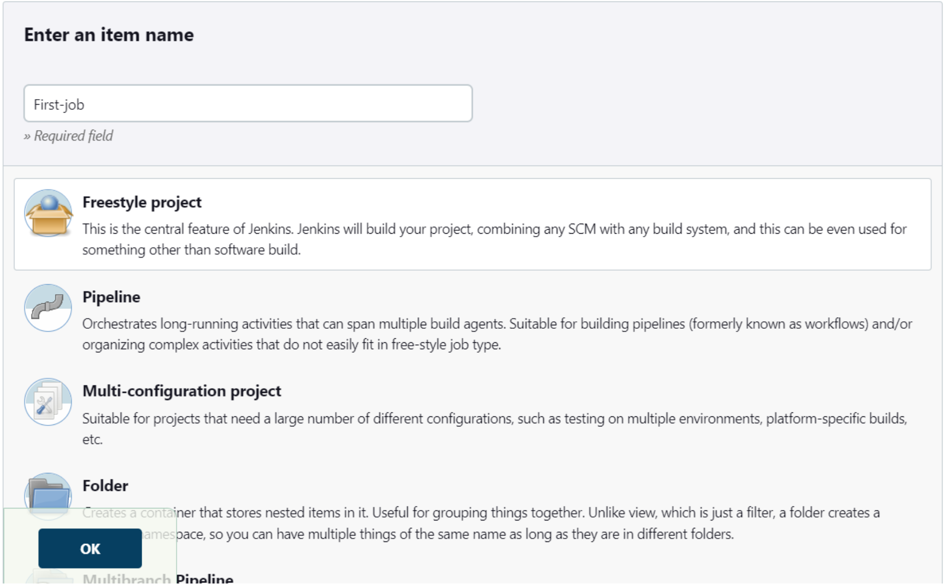
6) I create a job.



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



Enter an item name

First-job

» Required field

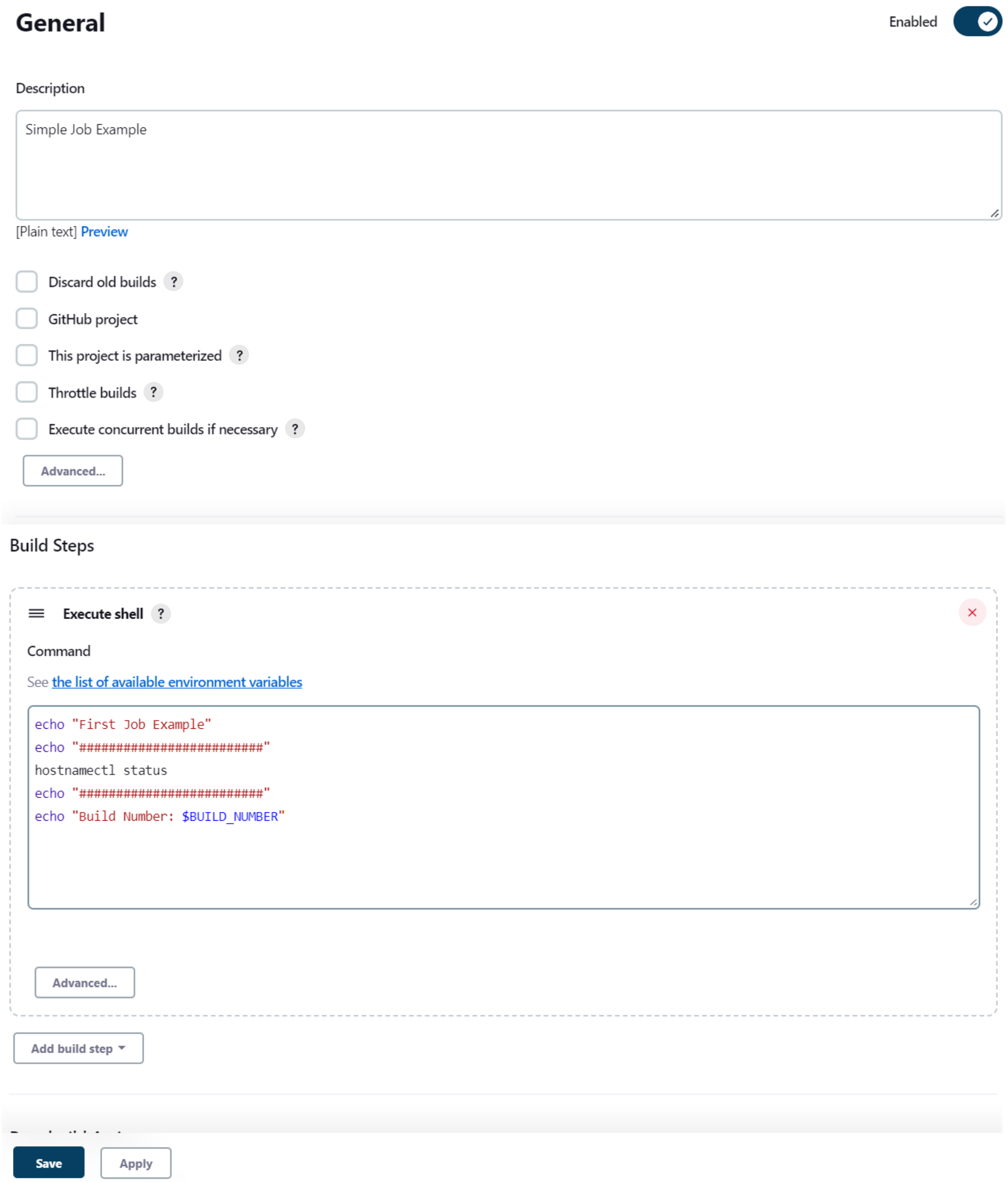
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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 namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline



General

Enabled

Description

Simple Job Example

[Plain text] Preview

□ Discard old builds ?

□ GitHub project

□ This project is parameterized ?

□ Throttle builds ?

□ Execute concurrent builds if necessary ?

Build Steps

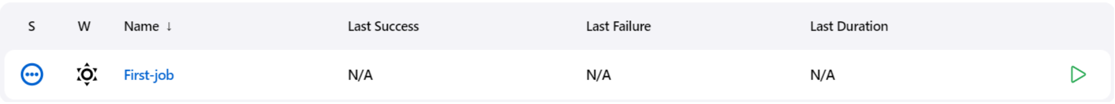
Execute shell ?

Command

See the list of available environment variables

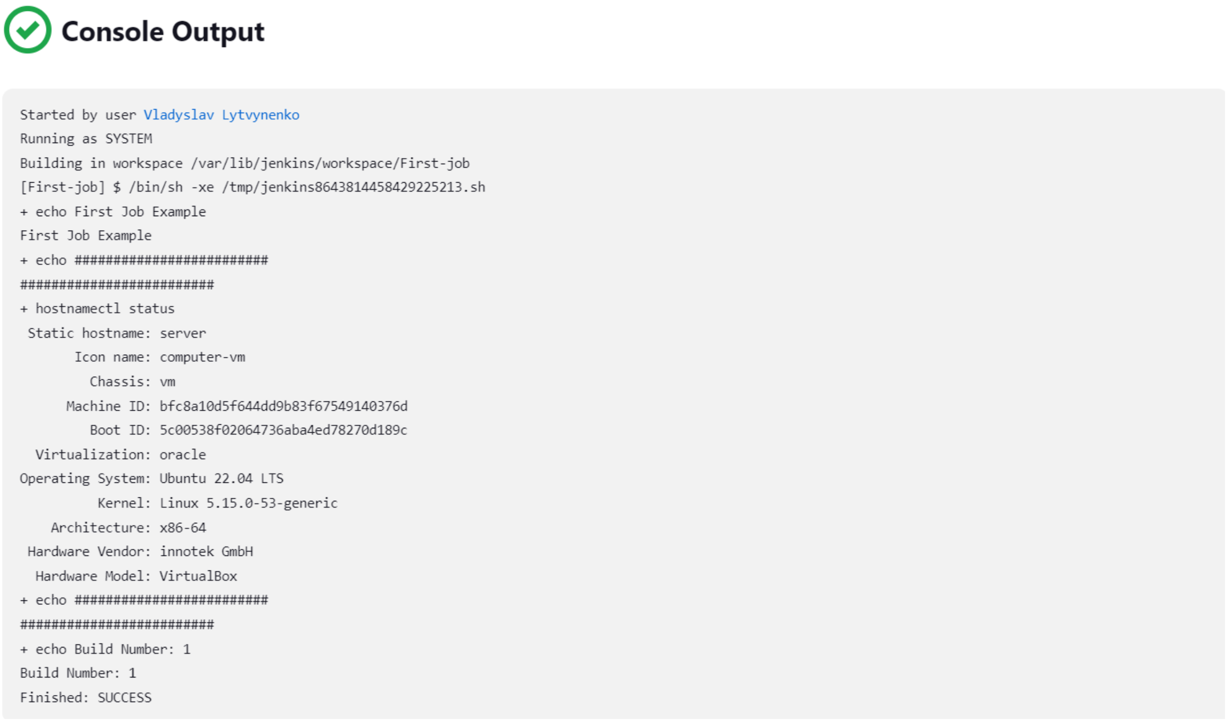
```
echo "First Job Example"
echo "########################"
hostnamectl status
echo "########################"
echo "Build Number: $BUILD_NUMBER"
```

7) I start the job.



S	W	Name ↓	Last Success	Last Failure	Last Duration	
⊙	☼	First-job	N/A	N/A	N/A	▷

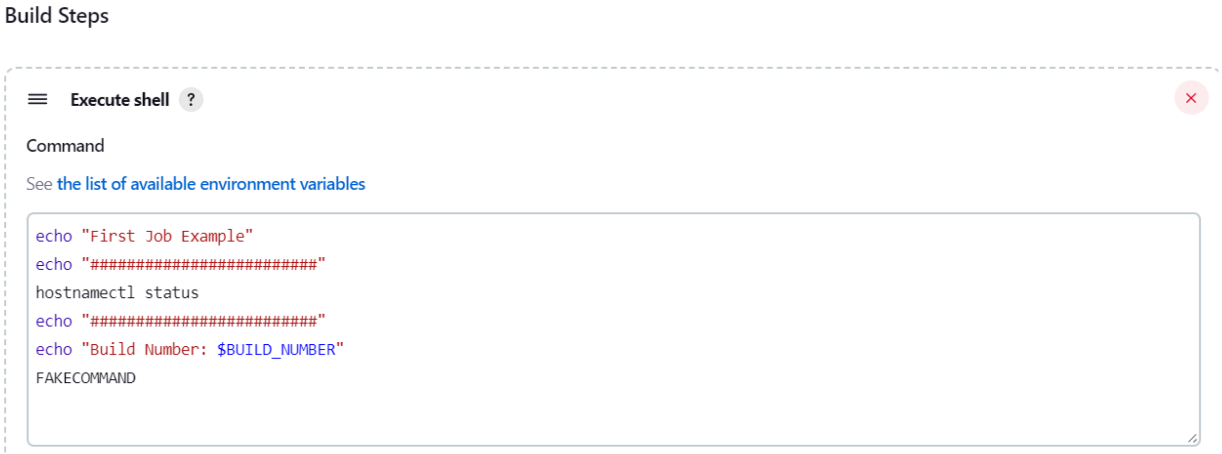
Execution Result:



Console Output

```
Started by user Vladyslav Lytvynenko
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/First-job
[First-job] $ /bin/sh -xe /tmp/jenkins8643814458429225213.sh
+ echo First Job Example
First Job Example
+ echo #########################
#########################
+ hostnamectl status
 Static hostname: server
       Icon name: computer-vm
         Chassis: vm
      Machine ID: bfc8a10d5f644dd9b83f67549140376d
         Boot ID: 5c00538f02064736aba4ed78270d189c
  Virtualization: oracle
Operating System: Ubuntu 22.04 LTS
          Kernel: Linux 5.15.0-53-generic
    Architecture: x86-64
 Hardware Vendor: innotek GmbH
  Hardware Model: VirtualBox
+ echo #########################
#########################
+ echo Build Number: 1
Build Number: 1
Finished: SUCCESS
```

8) I configure the job so that the build result is failed and check a result.



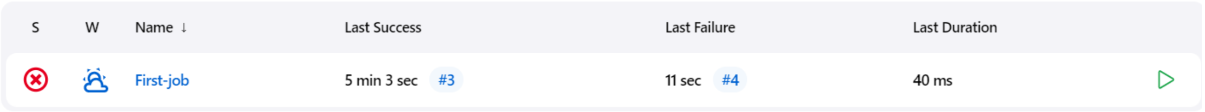
Build Steps

Execute shell

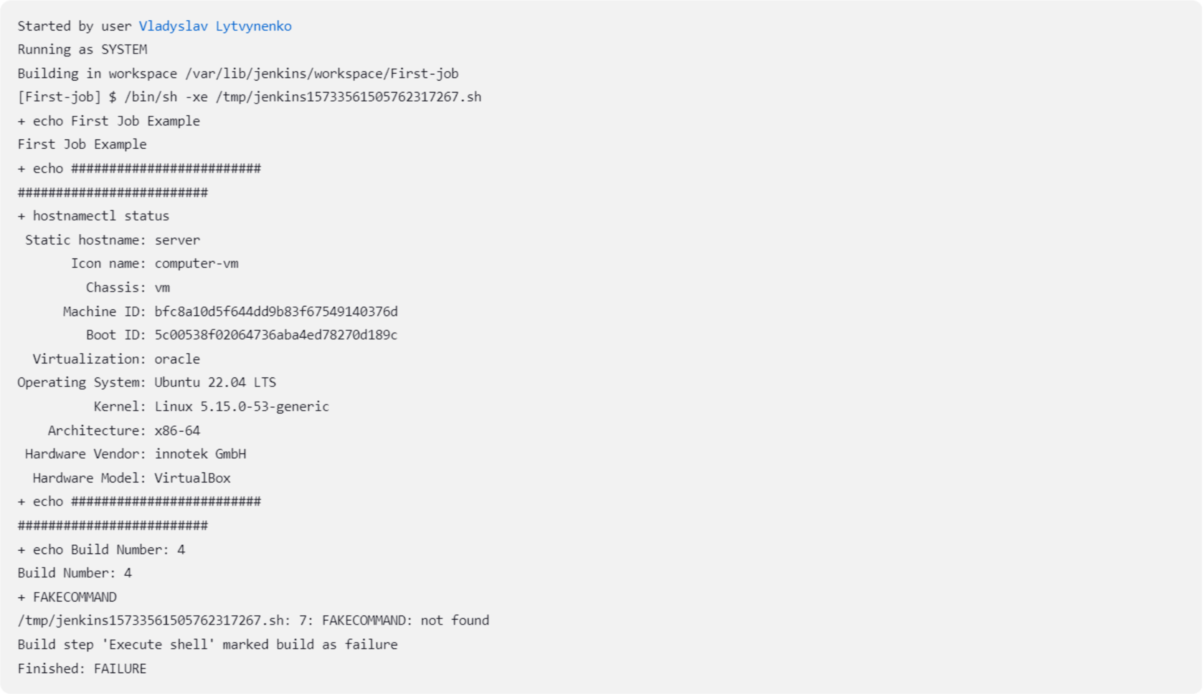
Command

See the list of available environment variables

```
echo "First Job Example"
echo "#########################"
hostnamectl status
echo "#########################"
echo "Build Number: $BUILD_NUMBER"
FAKECOMMAND
```

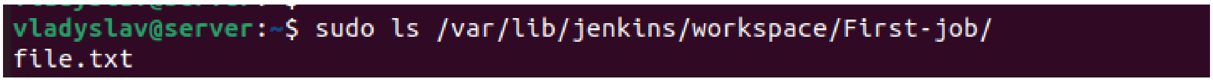


S	W	Name ↓	Last Success	Last Failure	Last Duration
⊗		First-job	5 min 3 sec #3	11 sec #4	40 ms



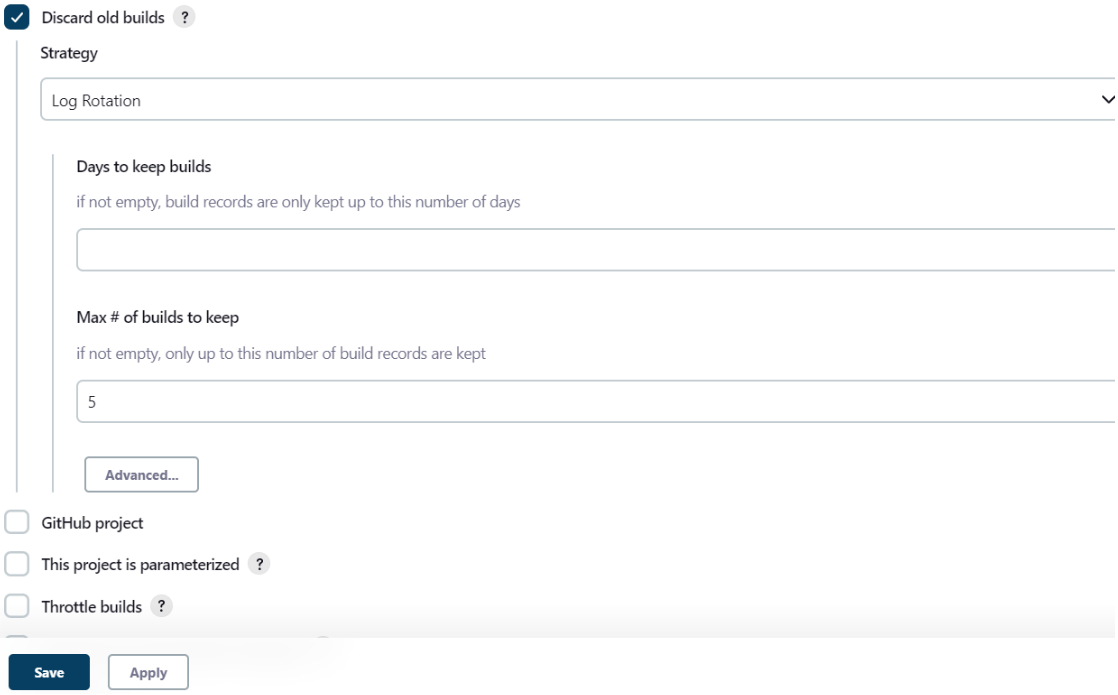
```
Started by user Vladyslav Lytvynenko
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/First-job
[First-job] $ /bin/sh -xe /tmp/jenkins15733561505762317267.sh
+ echo First Job Example
First Job Example
+ echo #########################
#########################
+ hostnamectl status
 Static hostname: server
       Icon name: computer-vm
         Chassis: vm
      Machine ID: bfc8a10d5f644dd9b83f67549140376d
         Boot ID: 5c00538f02064736aba4ed78270d189c
  Virtualization: oracle
Operating System: Ubuntu 22.04 LTS
          Kernel: Linux 5.15.0-53-generic
    Architecture: x86-64
 Hardware Vendor: innotek GmbH
  Hardware Model: VirtualBox
+ echo #########################
#########################
+ echo Build Number: 4
Build Number: 4
+ FAKECOMMAND
/tmp/jenkins15733561505762317267.sh: 7: FAKECOMMAND: not found
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

9) I check the folder where the builds are stored.



```
vladyslav@server:~$ sudo ls /var/lib/jenkins/workspace/First-job/
file.txt
```

10) I configure the job to discard old builds.



☑ Discard old builds ?

Strategy

Log Rotation

Days to keep builds

if not empty, build records are only kept up to this number of days

Max # of builds to keep

if not empty, only up to this number of build records are kept

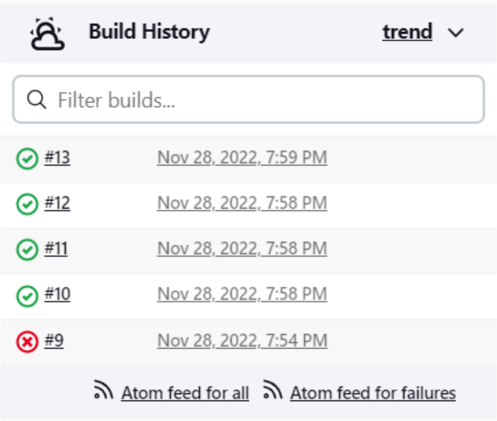
5

☐ GitHub project

☐ This project is parameterized ?

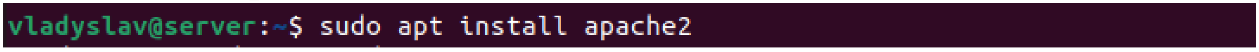
☐ Throttle builds ?

Records about old builds are removed:

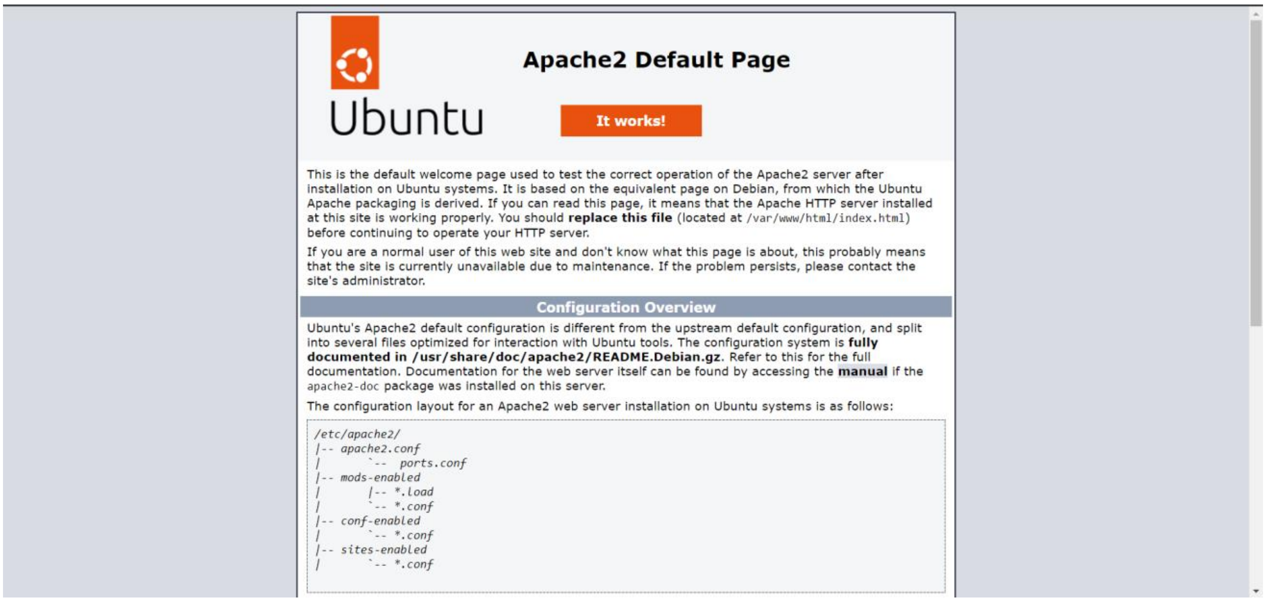


✓ #13	Nov 28, 2022, 7:59 PM
✓ #12	Nov 28, 2022, 7:58 PM
✓ #11	Nov 28, 2022, 7:58 PM
✓ #10	Nov 28, 2022, 7:58 PM
✗ #9	Nov 28, 2022, 7:54 PM

11) I install apache on the virtual machine.



```
vladyslav@server:~$ sudo apt install apache2
```



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

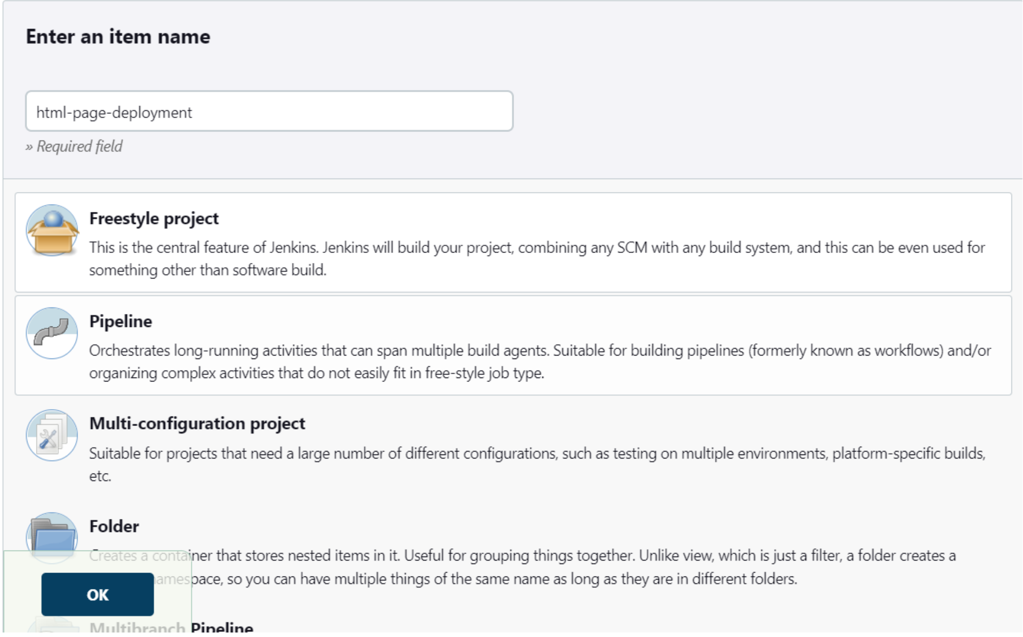
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

12) I create an html page with Jenkins job and deploy it on the apache server.

I create a new job:



Enter an item name

html-page-deployment

» 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.

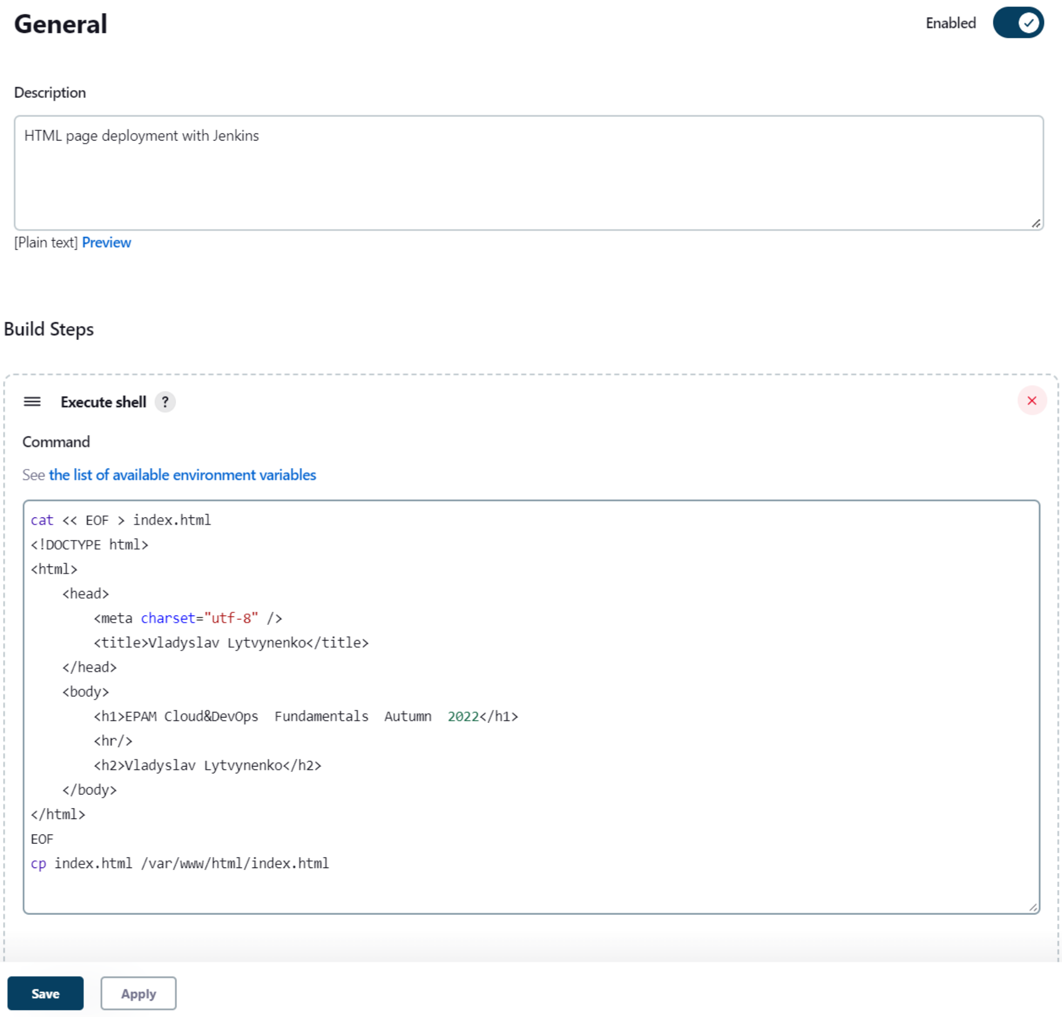
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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 namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Multibranch Pipeline

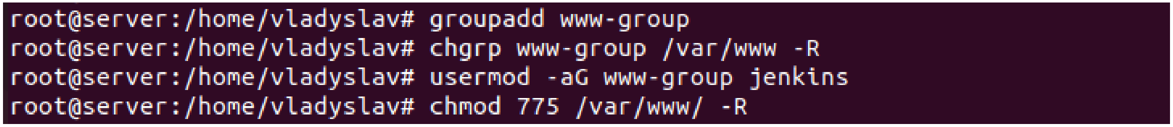


Description

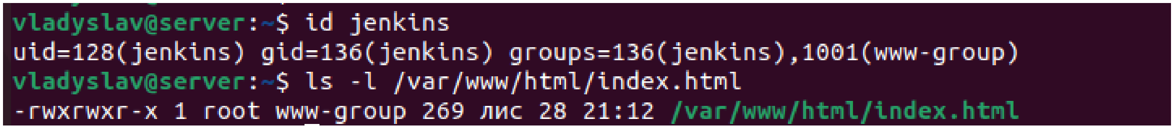
HTML page deployment with Jenkins

```
cat << EOF > index.html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>Vladyslav Lytvynenko</title>
    </head>
    <body>
        <h1>EPAM Cloud&DevOps  Fundamentals  Autumn  2022</h1>
        <hr/>
        <h2>Vladyslav Lytvynenko</h2>
    </body>
</html>
EOF
cp index.html /var/www/html/index.html
```

I grant Jenkins permissions to modify web-server files:

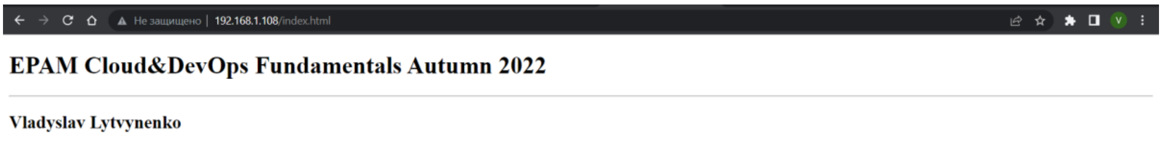


```
root@server:/home/vladyslav# groupadd www-group
root@server:/home/vladyslav# chgrp www-group /var/www -R
root@server:/home/vladyslav# usermod -aG www-group jenkins
root@server:/home/vladyslav# chmod 775 /var/www/ -R
```



```
vladyslav@server:~$ id jenkins
uid=128(jenkins) gid=136(jenkins) groups=136(jenkins),1001(www-group)
vladyslav@server:~$ ls -l /var/www/html/index.html
-rwxrwxr-x 1 root www-group 269 лис 28 21:12 /var/www/html/index.html
```

Build Result:

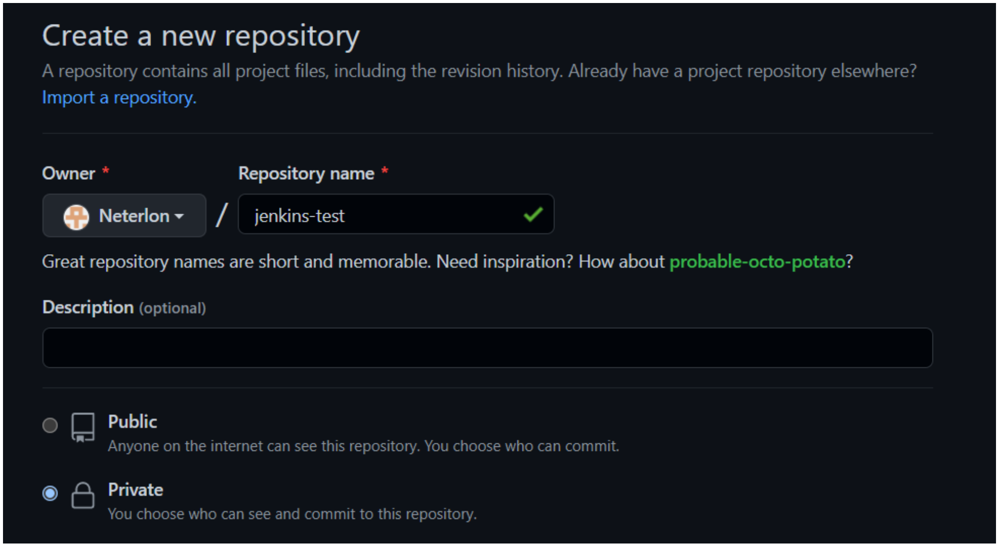


EPAM Cloud&DevOps Fundamentals Autumn 2022

Vladyslav Lytvynenko

Simple CI/CD pipeline

13) I create a new repository on GitHub:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

Neterlon / jenkins-test ✓

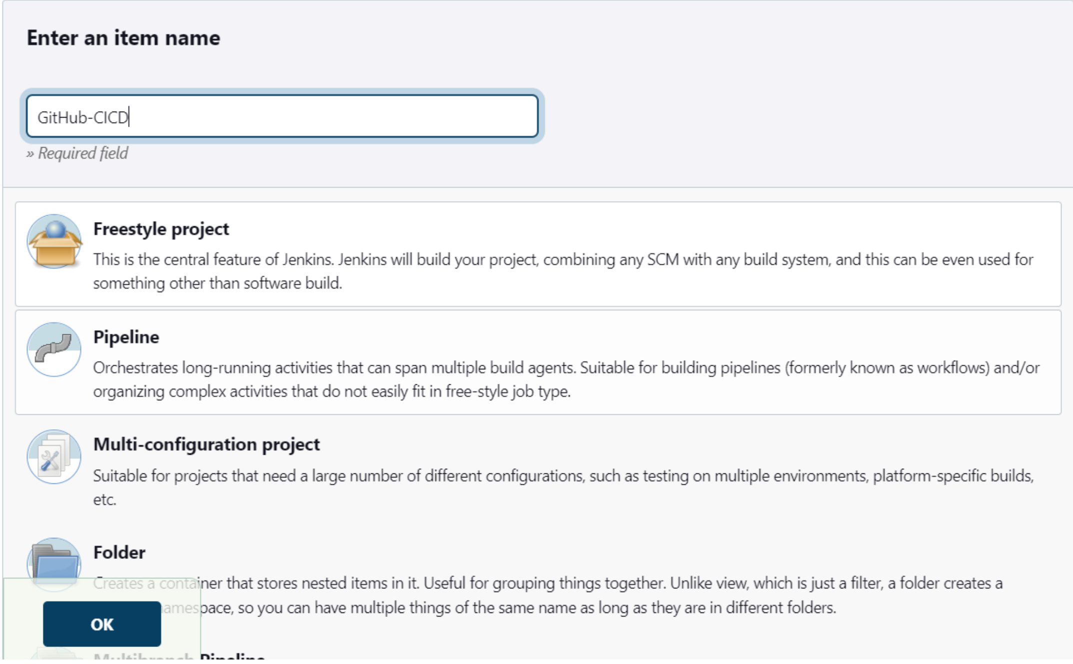
Great repository names are short and memorable. Need inspiration? How about probable-octo-potato?

Description (optional)

○ Public
Anyone on the internet can see this repository. You choose who can commit.

● Private
You choose who can see and commit to this repository.

14) I create a new job to deploy html pages on a web server from a GitHub repository:



Enter an item name

GitHub-CICD

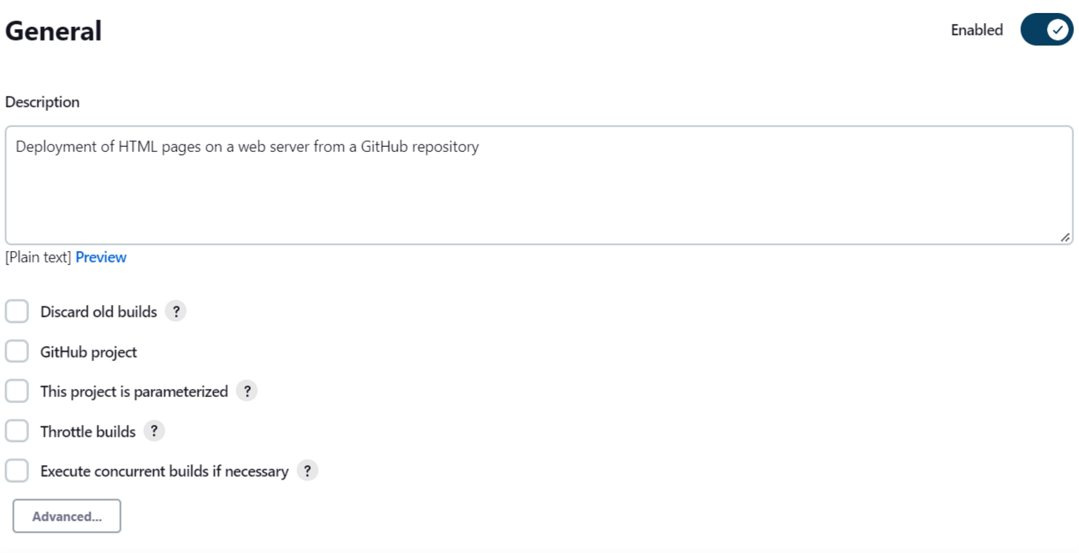
» Required field

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 namespace, so you can have multiple things of the same name as long as they are in different folders.



General Enabled

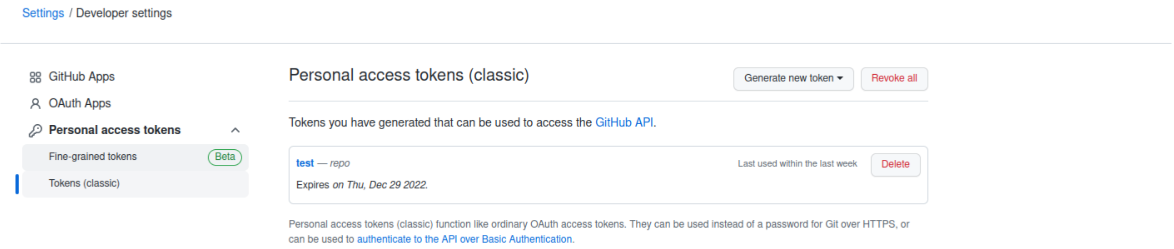
Description

Deployment of HTML pages on a web server from a GitHub repository

[Plain text] Preview

□ Discard old builds ?
□ GitHub project
□ This project is parameterized ?
□ Throttle builds ?
□ Execute concurrent builds if necessary ?

I create a token on GitHub:



Settings / Developer settings

Personal access tokens (classic)

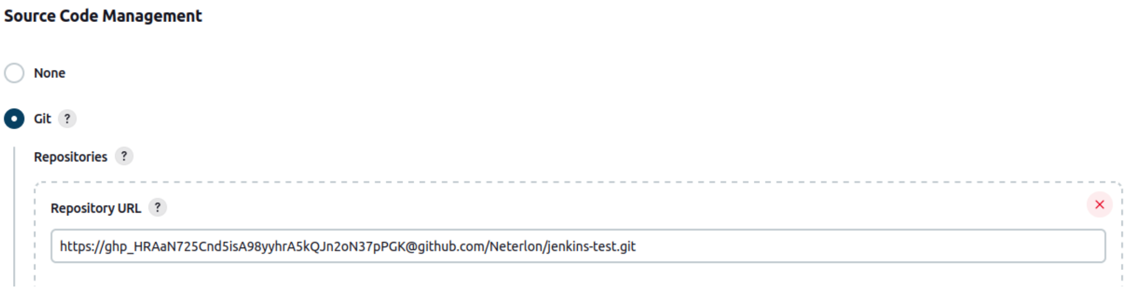
Tokens you have generated that can be used to access the GitHub API.

test — *repo* Last used within the last week Delete

Expires *on Thu, Dec 29 2022.*

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

In jenkins, I point to the repository with the generated token:



Source Code Management

○ None

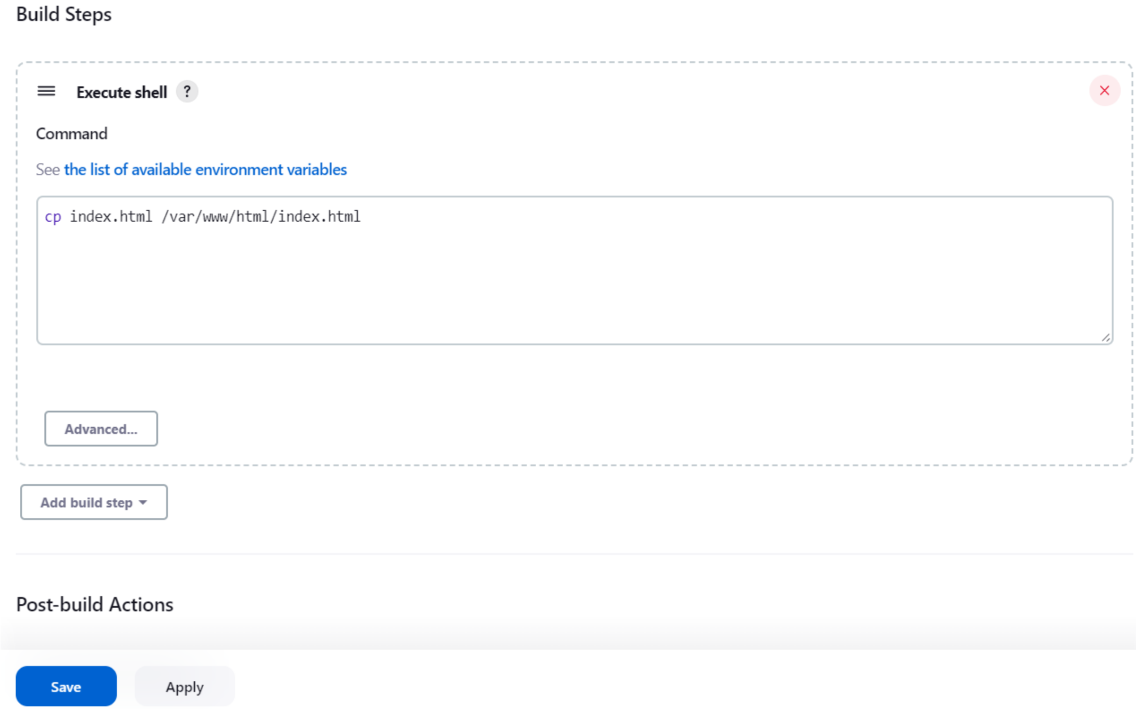
● Git ?

Repositories ?

Repository URL ?

https://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@github.com/Neterlon/jenkins-test.git

I copy a file from the repository to the web-server directory:



Build Steps

Execute shell ?

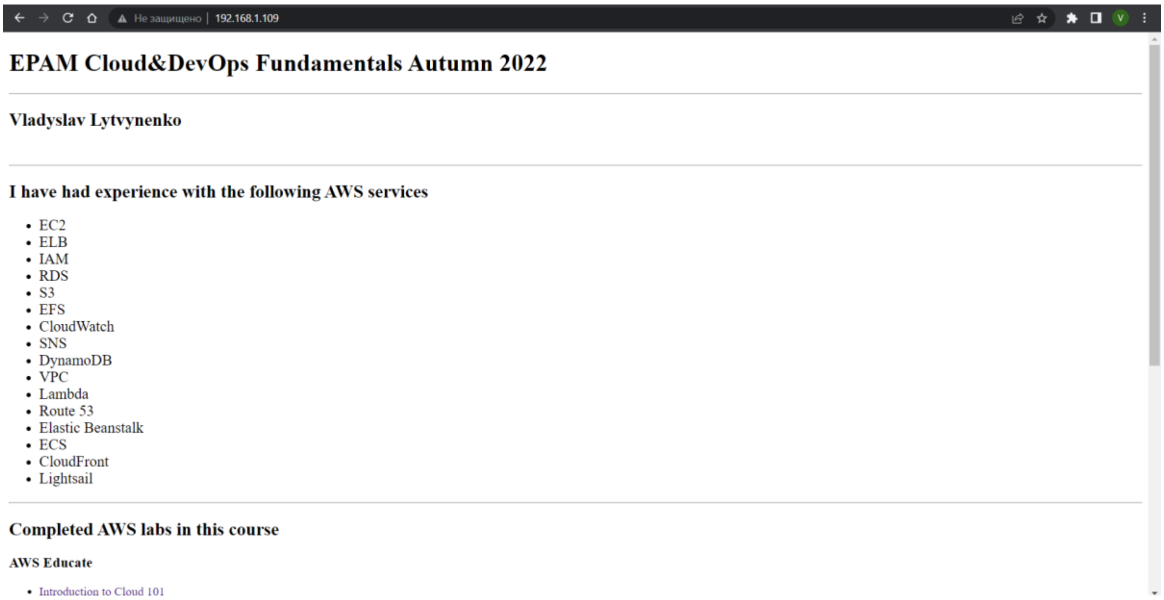
Command

See the list of available environment variables

```
cp index.html /var/www/html/index.html
```

Post-build Actions

15) I build the project and check the execution result:



EPAM Cloud&DevOps Fundamentals Autumn 2022

Vladyslav Lytvynenko

I have had experience with the following AWS services

- EC2
- ELB
- IAM
- RDS
- S3
- EFS
- CloudWatch
- SNS
- DynamoDB
- VPC
- Lambda
- Route 53
- Elastic Beanstalk
- ECS
- CloudFront
- Lightsail

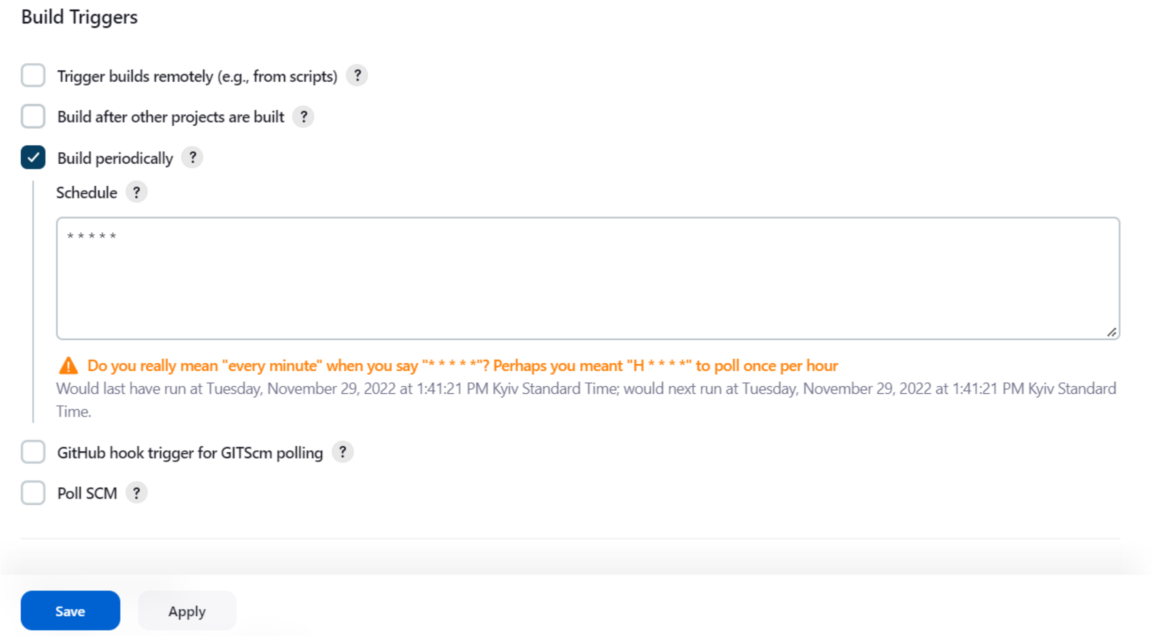
Completed AWS labs in this course

AWS Educate

- Introduction to Cloud 101

16) I try some types of triggers.

1) "Build periodically" Trigger:



Build Triggers

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

□ Build after other projects are built ?

☑ Build periodically ?

Schedule ?

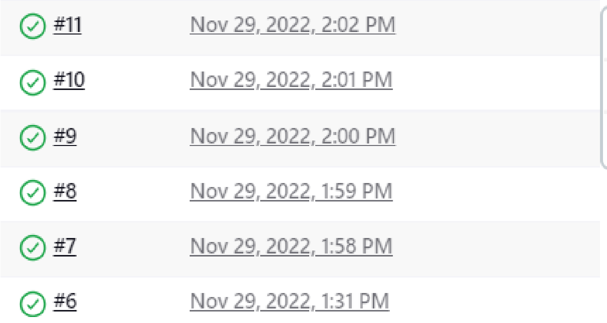
```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Tuesday, November 29, 2022 at 1:41:21 PM Kyiv Standard Time; would next run at Tuesday, November 29, 2022 at 1:41:21 PM Kyiv Standard Time.

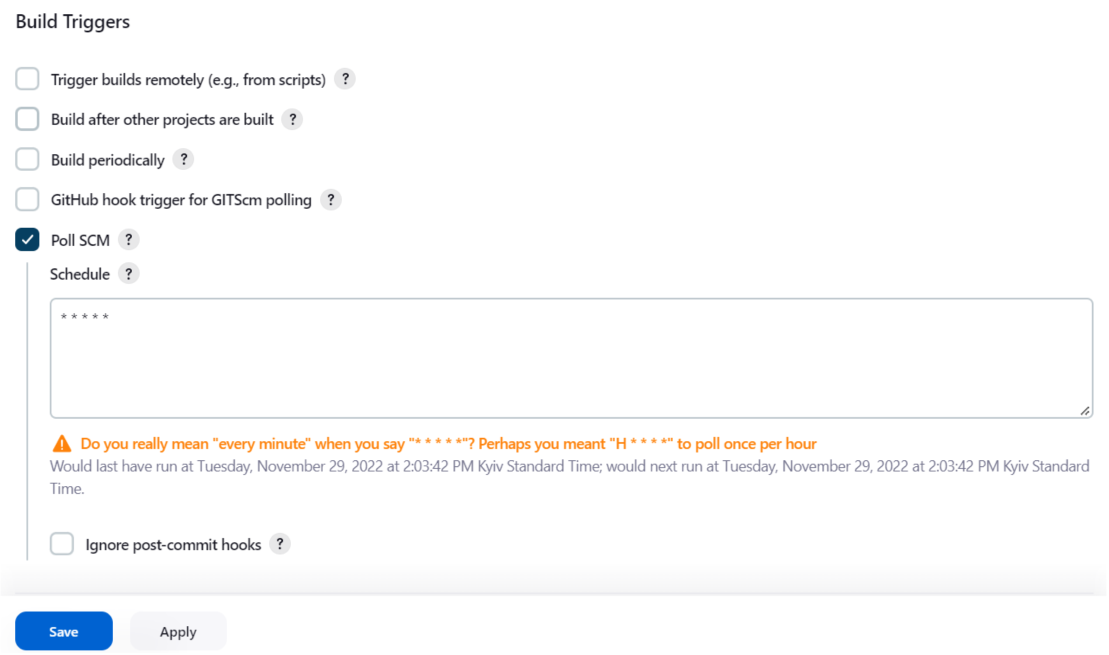
□ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?



✓ #11	Nov 29, 2022, 2:02 PM
✓ #10	Nov 29, 2022, 2:01 PM
✓ #9	Nov 29, 2022, 2:00 PM
✓ #8	Nov 29, 2022, 1:59 PM
✓ #7	Nov 29, 2022, 1:58 PM
✓ #6	Nov 29, 2022, 1:31 PM

2) "Poll SCM" Trigger:



Build Triggers

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

□ Build after other projects are built ?

□ Build periodically ?

□ GitHub hook trigger for GITScm polling ?

☑ Poll SCM ?

Schedule ?

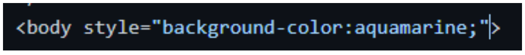
```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Tuesday, November 29, 2022 at 2:03:42 PM Kyiv Standard Time; would next run at Tuesday, November 29, 2022 at 2:03:42 PM Kyiv Standard Time.

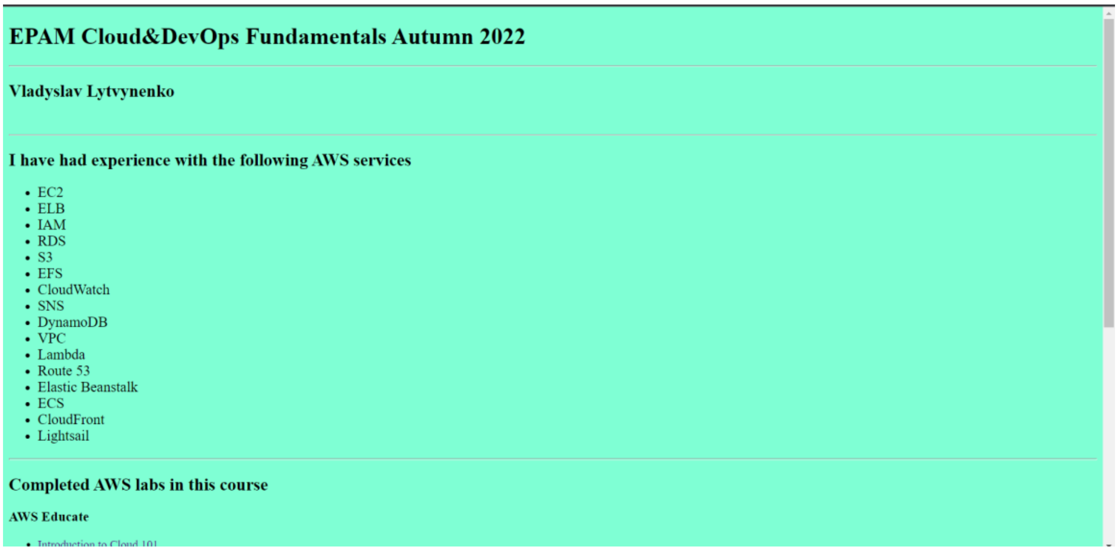
□ Ignore post-commit hooks ?

Change some code in the repository:



```
<body style="background-color:aquamarine;">
```

Build result:



EPAM Cloud&DevOps Fundamentals Autumn 2022

Vladyslav Lytvynenko

I have had experience with the following AWS services

- EC2
- ELB
- IAM
- RDS
- S3
- EFS
- CloudWatch
- SNS
- DynamoDB
- VPC
- Lambda
- Route 53
- Elastic Beanstalk
- ECS
- CloudFront
- Lightsail

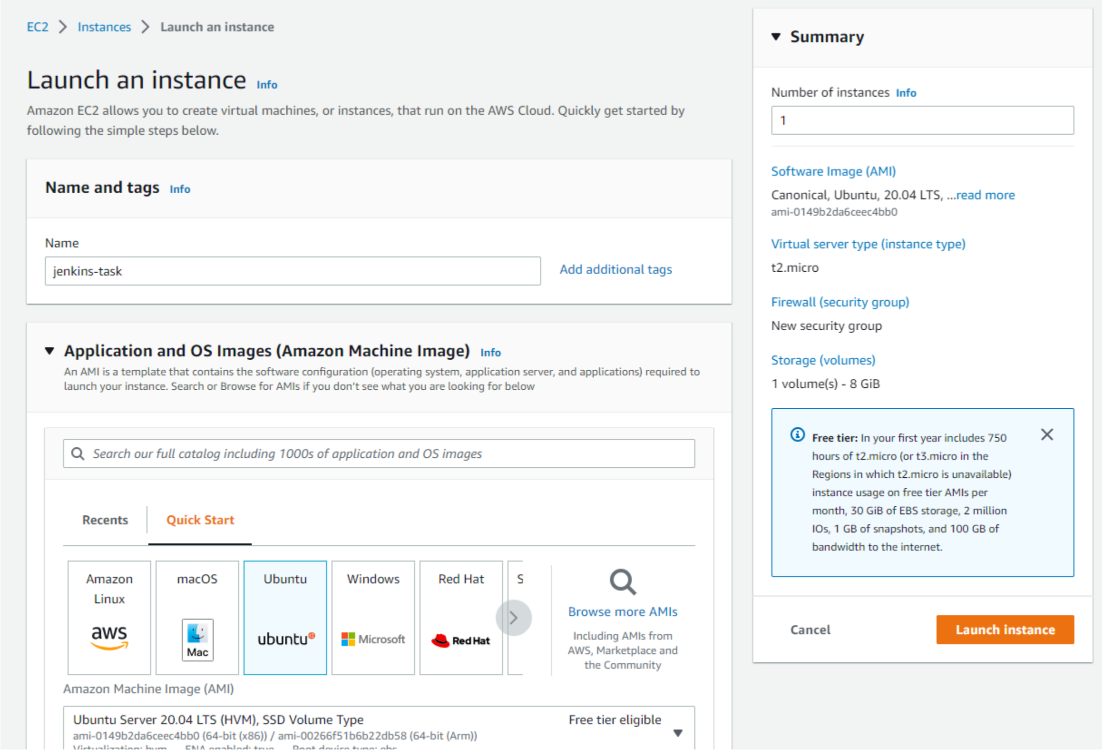
Completed AWS labs in this course

AWS Educate

- Introduction to Cloud 101

Simple CI/CD pipeline using AWS

17) I create an EC2 instance on AWS.



EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: jenkins-task

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | S

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type — Free tier eligible

ami-0149b2da6ceec4bb0 (64-bit (x86)) / ami-00266f51b6b22db58 (64-bit (Arm))

▼ Summary

Number of instances Info: 1

Software Image (AMI): Canonical, Ubuntu, 20.04 LTS, ...read more

ami-0149b2da6ceec4bb0

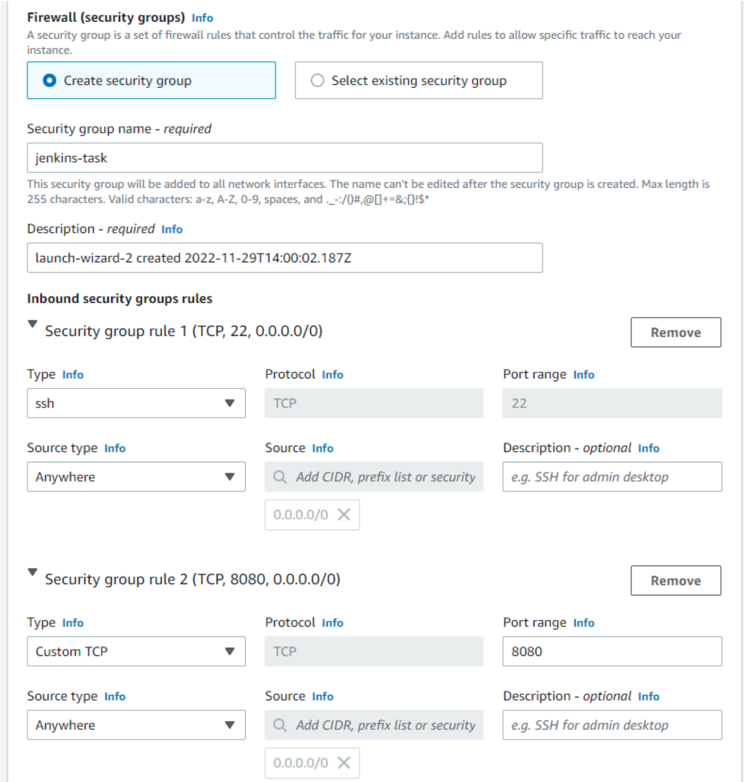
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance



Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - *required*: jenkins-task

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info: launch-wizard-2 created 2022-11-29T14:00:02.187Z

Inbound security groups rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) — Remove

Type	Protocol	Port range
ssh	TCP	22

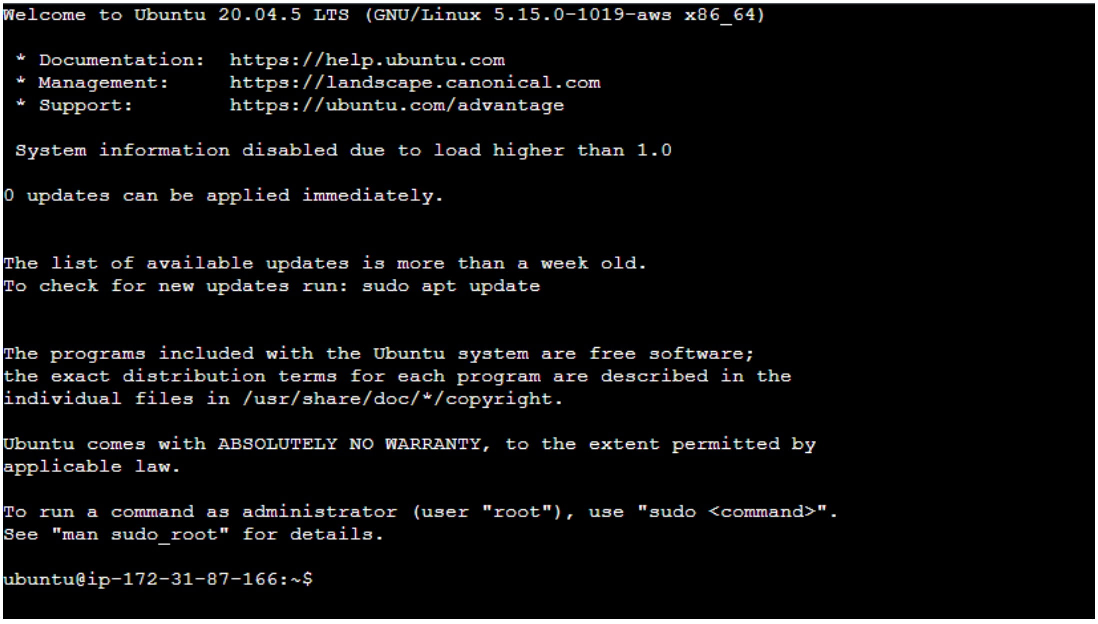
Source type	Source	Description - *optional*
Anywhere	0.0.0.0/0	*e.g. SSH for admin desktop*

▼ Security group rule 2 (TCP, 8080, 0.0.0.0/0) — Remove

Type	Protocol	Port range
Custom TCP	TCP	8080

Source type	Source	Description - *optional*
Anywhere	0.0.0.0/0	*e.g. SSH for admin desktop*

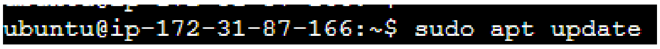
18) I connect to the instance through SSH and install Jenkins, Apache.



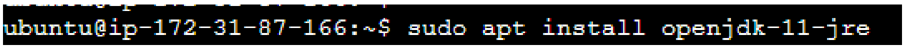
```
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information disabled due to load higher than 1.0

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-87-166:~$
```

Installation commands:

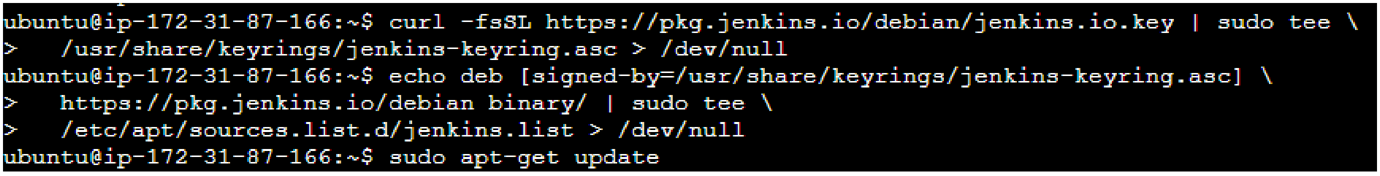
```
sudo apt-get update
sudo apt-get install openjdk-11-jre
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
sudo apt-get install apache2
```



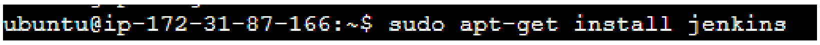
```
ubuntu@ip-172-31-87-166:~$ sudo apt update
```



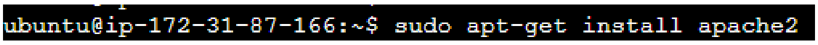
```
ubuntu@ip-172-31-87-166:~$ sudo apt install openjdk-11-jre
```



```
ubuntu@ip-172-31-87-166:~$ curl -fsSL https://pkg.jenkins.io/debian/jenkins.io.key | sudo tee \
>   /usr/share/keyrings/jenkins-keyring.asc > /dev/null
ubuntu@ip-172-31-87-166:~$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
>   https://pkg.jenkins.io/debian binary/ | sudo tee \
>   /etc/apt/sources.list.d/jenkins.list > /dev/null
ubuntu@ip-172-31-87-166:~$ sudo apt-get update
```

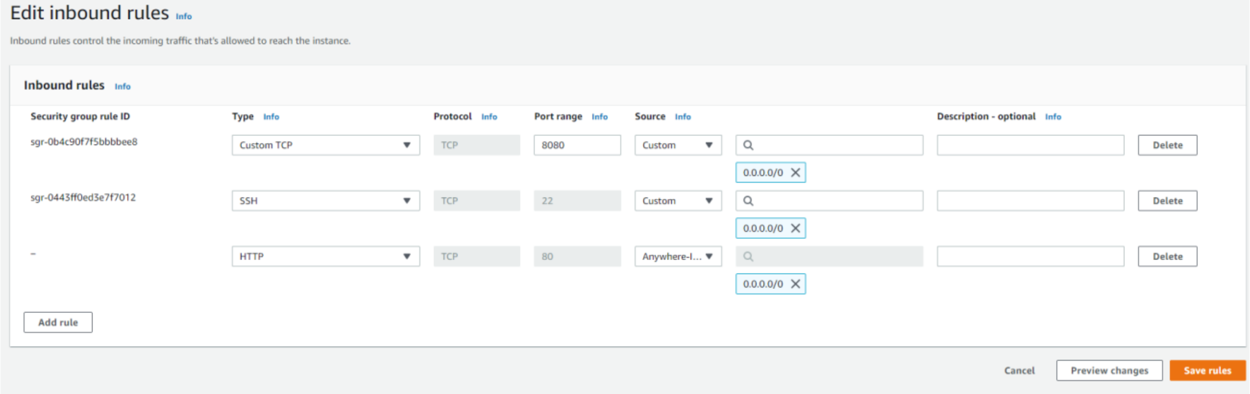


```
ubuntu@ip-172-31-87-166:~$ sudo apt-get install jenkins
```



```
ubuntu@ip-172-31-87-166:~$ sudo apt-get install apache2
```

19) I configure security group inbound rules.



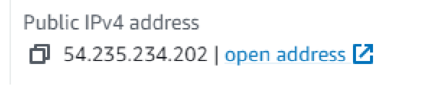
Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

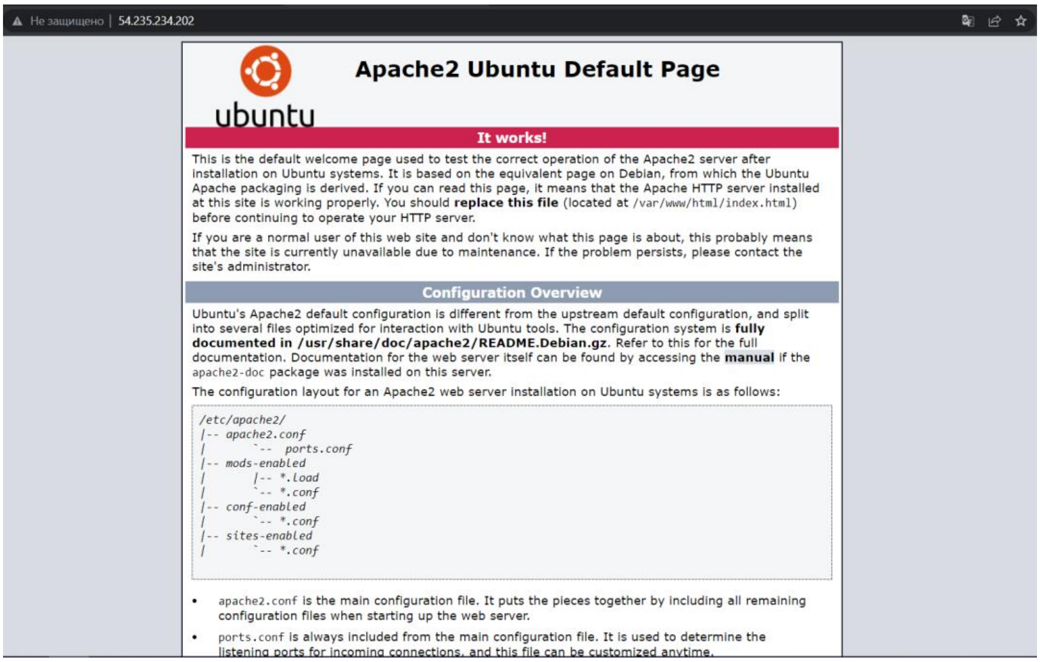
Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-0b4c90f7f5bbbbee8	Custom TCP	TCP	8080	Custom	0.0.0.0/0		Delete
sgr-0443ff0ed3e7f7012	SSH	TCP	22	Custom	0.0.0.0/0		Delete
–	HTTP	TCP	80	Anywhere-I...	0.0.0.0/0		Delete

20) I find out a public IP address of the instance and check installed services.



Public IPv4 address

54.235.234.202 | open address



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

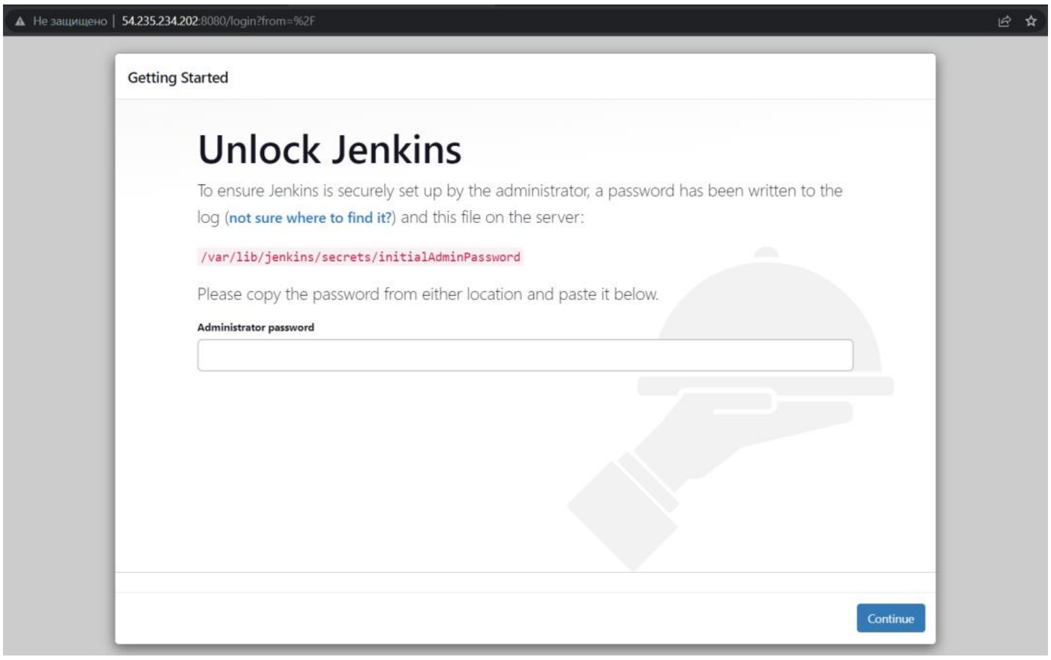
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.



Unlock Jenkins

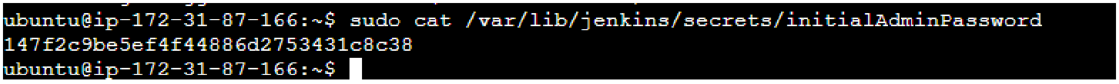
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

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

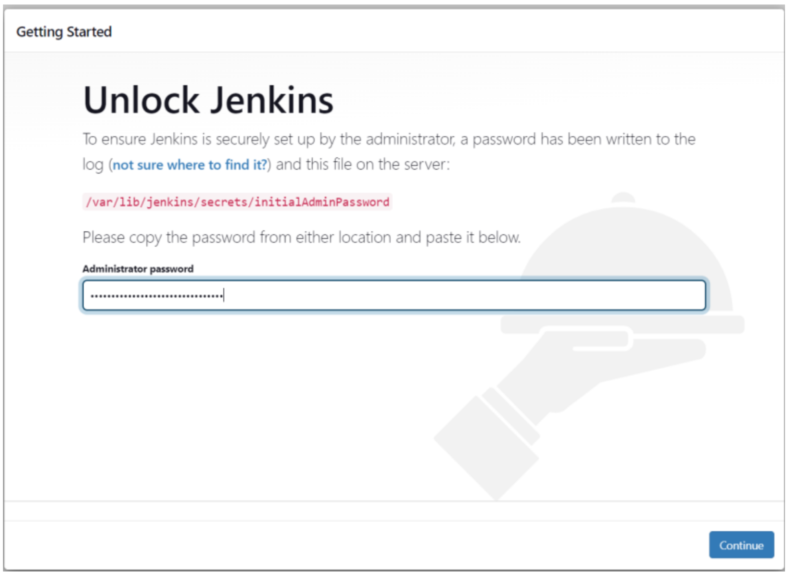
Please copy the password from either location and paste it below.

Administrator password

21) I find out a password and unlock Jenkins, execute basic setup.



```
ubuntu@ip-172-31-87-166:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
147f2c9be5ef4f44886d2753431c8c38
ubuntu@ip-172-31-87-166:~$
```



Getting Started

Unlock Jenkins

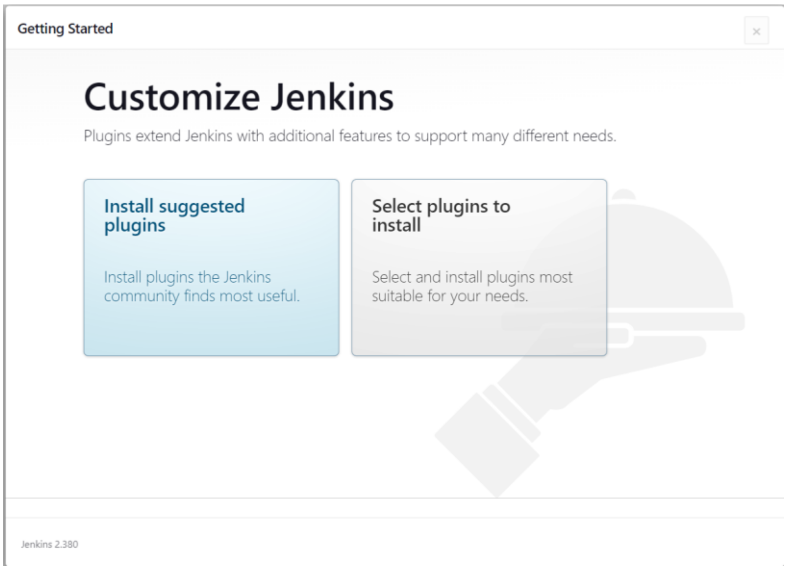
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

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

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

Administrator password

································



Getting Started

Customize Jenkins

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

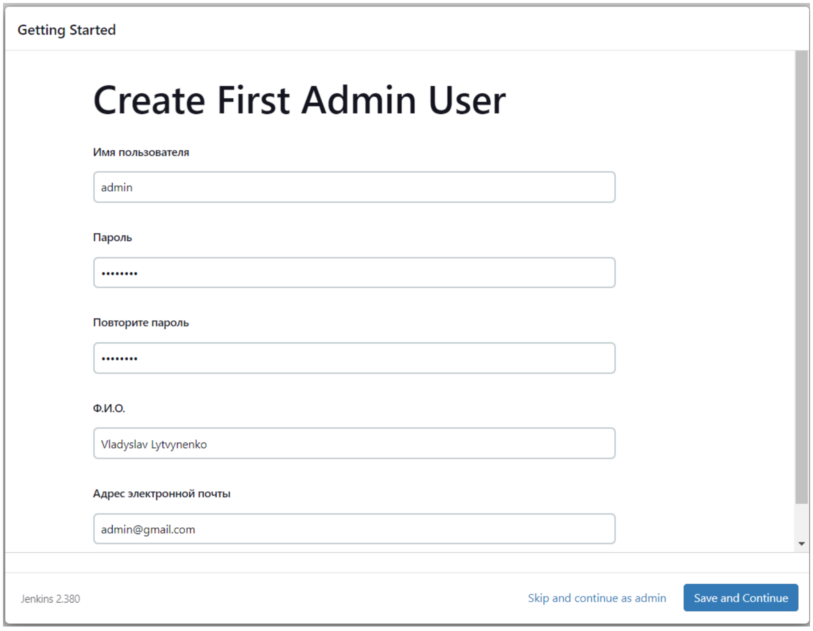
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.380



Getting Started

Create First Admin User

Имя пользователя: admin

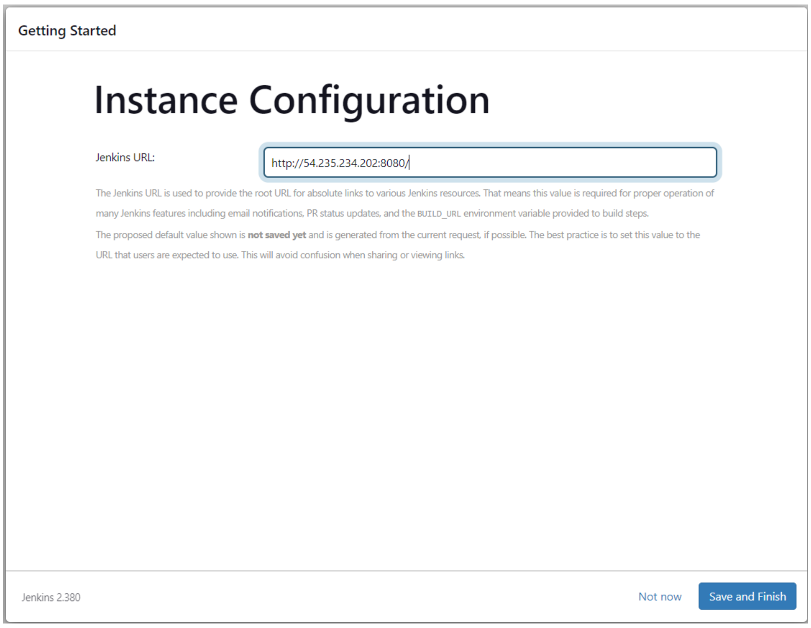
Пароль: ········

Повторите пароль: ········

Ф.И.О.: Vladyslav Lytvynenko

Адрес электронной почты: admin@gmail.com

Jenkins 2.380



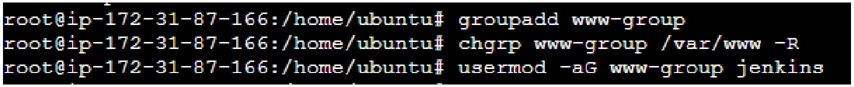
Instance Configuration

Jenkins URL: http://54.235.234.202:8080/

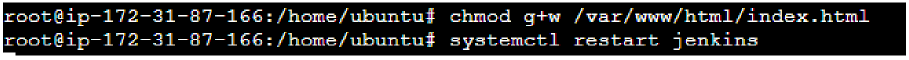
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

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.

22) I grant Jenkins permissions to modify web-server files.

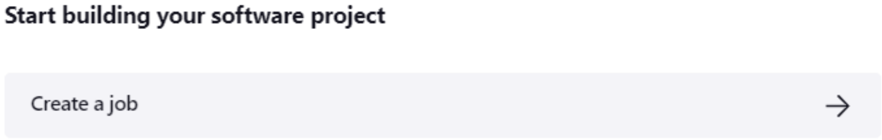


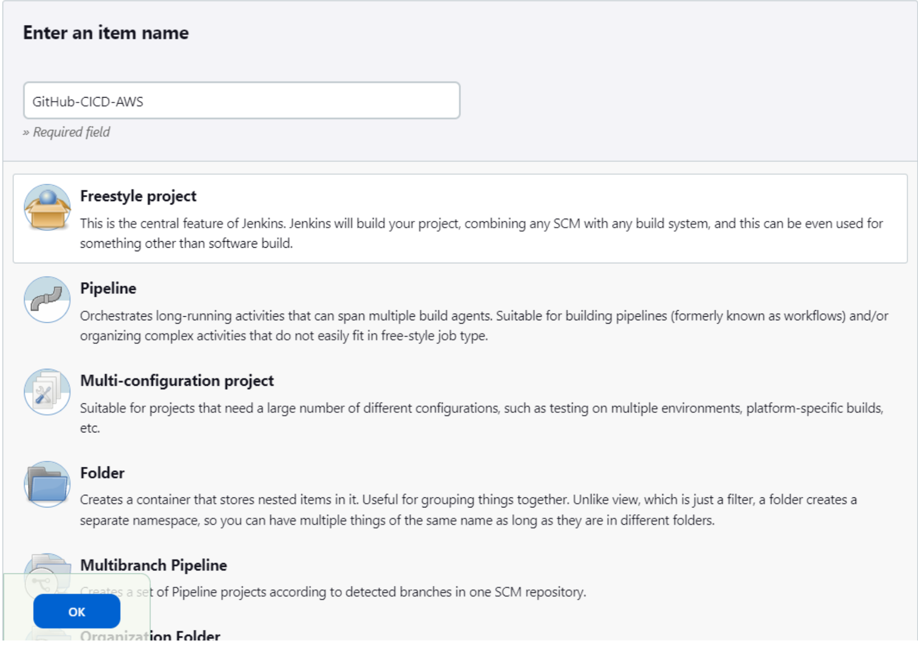
```
root@ip-172-31-87-166:/home/ubuntu# groupadd www-group
root@ip-172-31-87-166:/home/ubuntu# chgrp www-group /var/www -R
root@ip-172-31-87-166:/home/ubuntu# usermod -aG www-group jenkins
```



```
root@ip-172-31-87-166:/home/ubuntu# chmod g+w /var/www/html/index.html
root@ip-172-31-87-166:/home/ubuntu# systemctl restart jenkins
```

23) I create a new job to deploy html pages on a web server from a GitHub repository.





Enter an item name

GitHub-CICD-AWS

» *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.

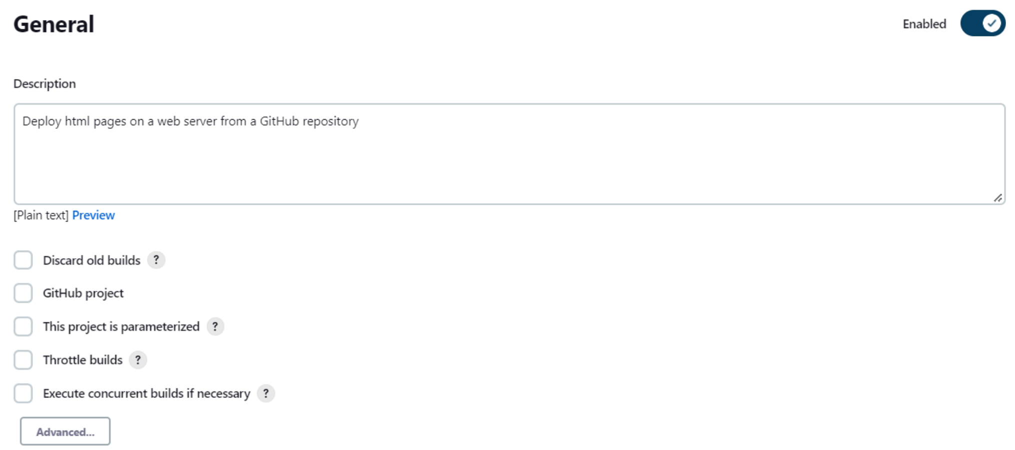
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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



General

Enabled

Description

Deploy html pages on a web server from a GitHub repository

[Plain text] Preview

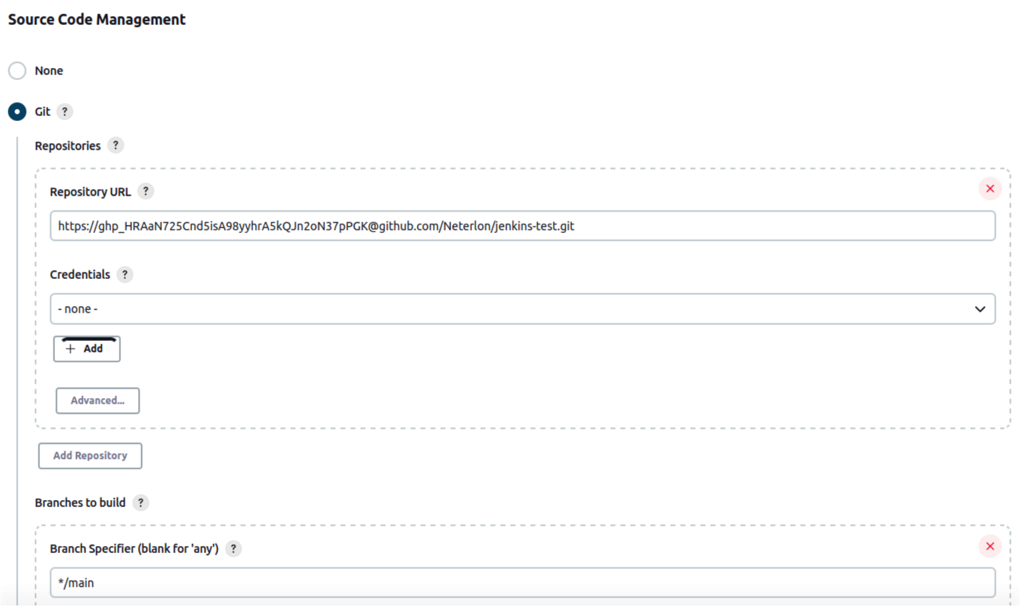
□ Discard old builds ?

□ GitHub project

□ This project is parameterized ?

□ Throttle builds ?

□ Execute concurrent builds if necessary ?



Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?

https://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@github.com/Neterlon/jenkins-test.git

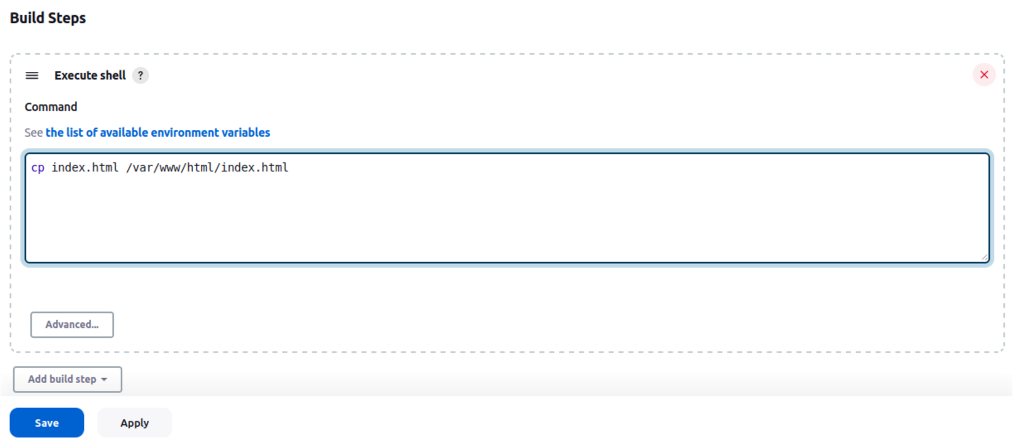
Credentials ?

- none -

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main



Build Steps

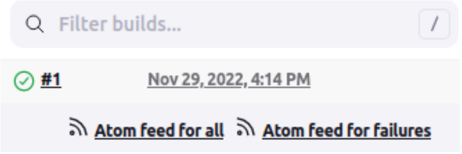
Execute shell ?

Command

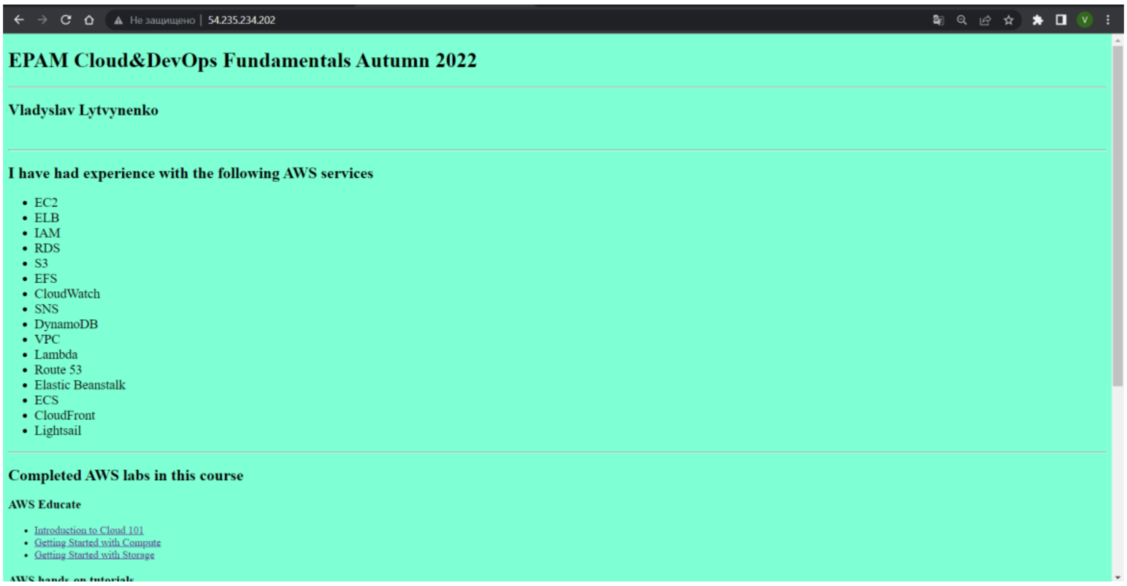
See the list of available environment variables

```
cp index.html /var/www/html/index.html
```

24) I run the job and check the result.



✓ #1 Nov 29, 2022, 4:14 PM



Не защищено | 54.235.234.202

EPAM Cloud&DevOps Fundamentals Autumn 2022

Vladyslav Lytvynenko

I have had experience with the following AWS services

- EC2
- ELB
- IAM
- RDS
- S3
- EFS
- CloudWatch
- SNS
- DynamoDB
- VPC
- Lambda
- Route 53
- Elastic Beanstalk
- ECS
- CloudFront
- Lightsail

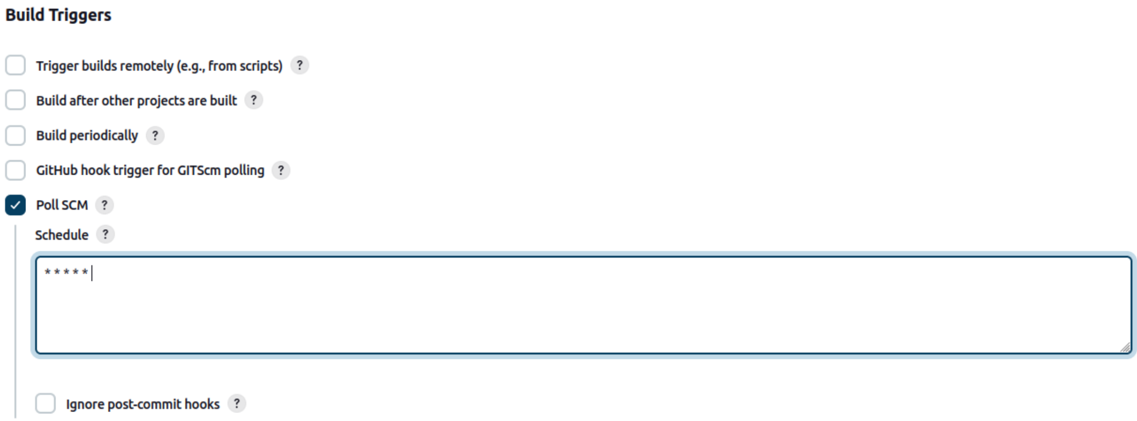
Completed AWS labs in this course

AWS Educate

- Introduction to Cloud 101
- Getting Started with Compute
- Getting Started with Storage

AWS hands-on tutorials

25) I add I a trigger to the job.



Build Triggers

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

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

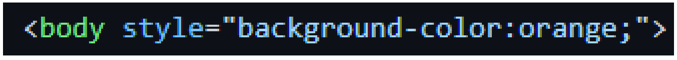
☑ Poll SCM ?

Schedule ?

```
* * * * *
```

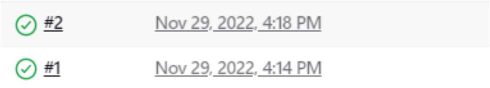
☐ Ignore post-commit hooks ?

26) I change some code in the repository.



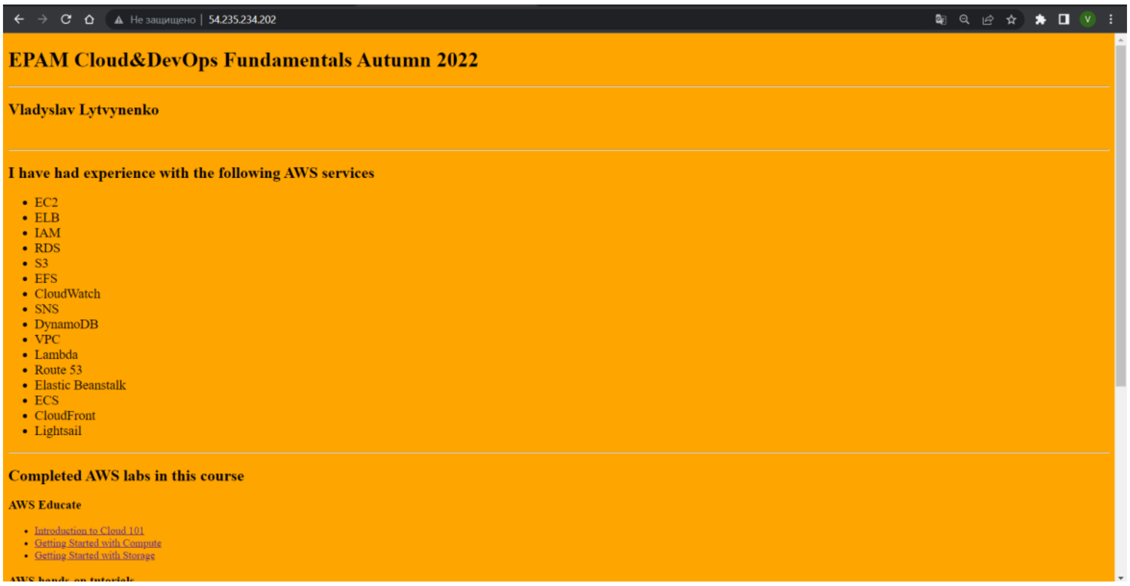
```
<body style="background-color:orange;">
```

Job execution result:



#2 Nov 29, 2022, 4:18 PM

#1 Nov 29, 2022, 4:14 PM



Не защищено | 54.235.234.202

EPAM Cloud&DevOps Fundamentals Autumn 2022

Vladyslav Lytvynenko

I have had experience with the following AWS services

- EC2
- ELB
- IAM
- RDS
- S3
- EFS
- CloudWatch
- SNS
- DynamoDB
- VPC
- Lambda
- Route 53
- Elastic Beanstalk
- ECS
- CloudFront
- Lightsail

Completed AWS labs in this course

AWS Educate

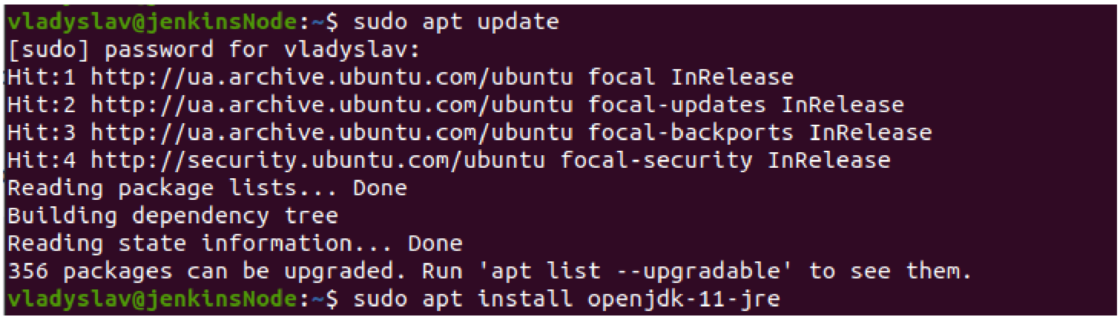
- Introduction to Cloud 101
- Getting Started with Compute
- Getting Started with Storage

AWS hands-on tutorials

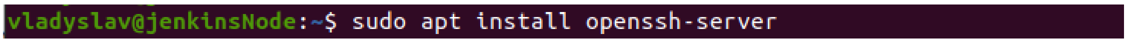
Jenkins Nodes

27) I create a new virtual machine; install java on it; configure SSH, Jenkins user, jenkins directory.

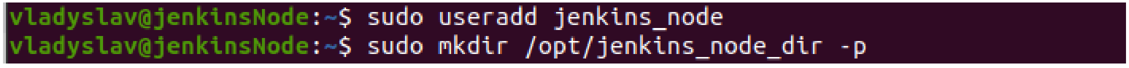
Jenkins Node:



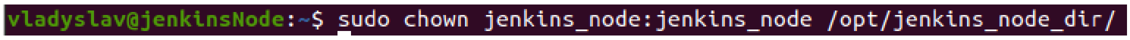
```
vladyslav@jenkinsNode:~$ sudo apt update
[sudo] password for vladyslav:
Hit:1 http://ua.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ua.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ua.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
356 packages can be upgraded. Run 'apt list --upgradable' to see them.
vladyslav@jenkinsNode:~$ sudo apt install openjdk-11-jre
```



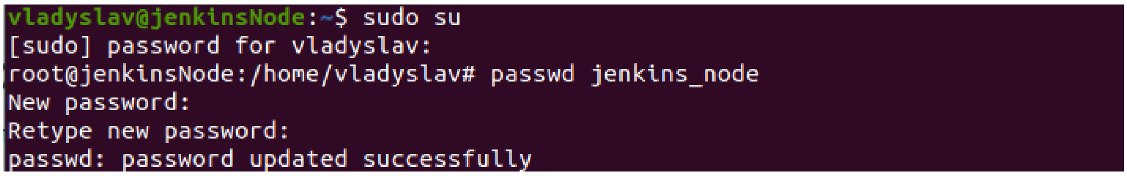
```
vladyslav@jenkinsNode:~$ sudo apt install openssh-server
```



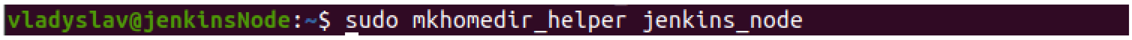
```
vladyslav@jenkinsNode:~$ sudo useradd jenkins_node
vladyslav@jenkinsNode:~$ sudo mkdir /opt/jenkins_node_dir -p
```



```
vladyslav@jenkinsNode:~$ sudo chown jenkins_node:jenkins_node /opt/jenkins_node_dir/
```

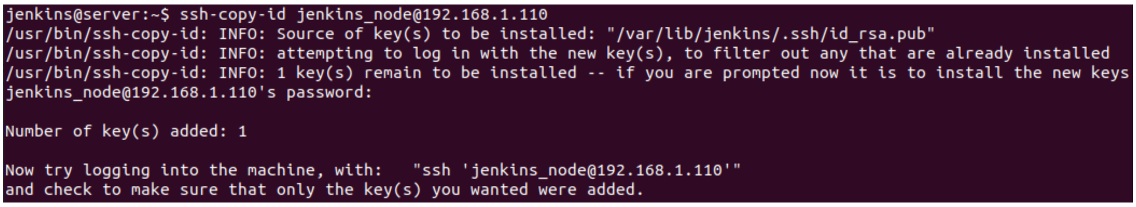


```
vladyslav@jenkinsNode:~$ sudo su
[sudo] password for vladyslav:
root@jenkinsNode:/home/vladyslav# passwd jenkins_node
New password:
Retype new password:
passwd: password updated successfully
```



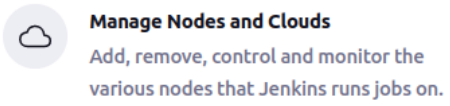
```
vladyslav@jenkinsNode:~$ sudo mkhomedir_helper jenkins_node
```

Jenkins Server:



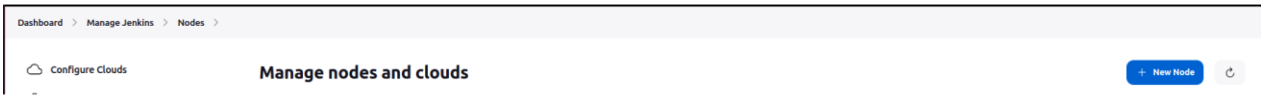
```
jenkins@server:~$ ssh-copy-id jenkins_node@192.168.1.110
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/var/lib/jenkins/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
jenkins_node@192.168.1.110's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'jenkins_node@192.168.1.110'"
and check to make sure that only the key(s) you wanted were added.
```

28) I configure a new node on Jenkins server.

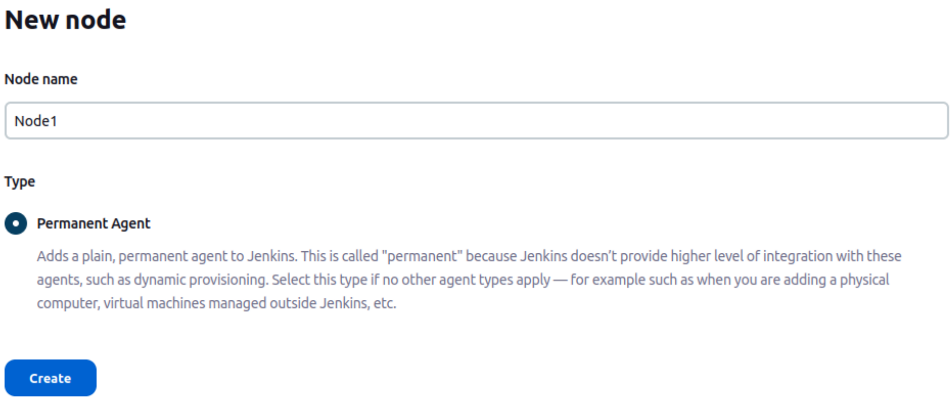


Manage Nodes and Clouds

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



Manage nodes and clouds



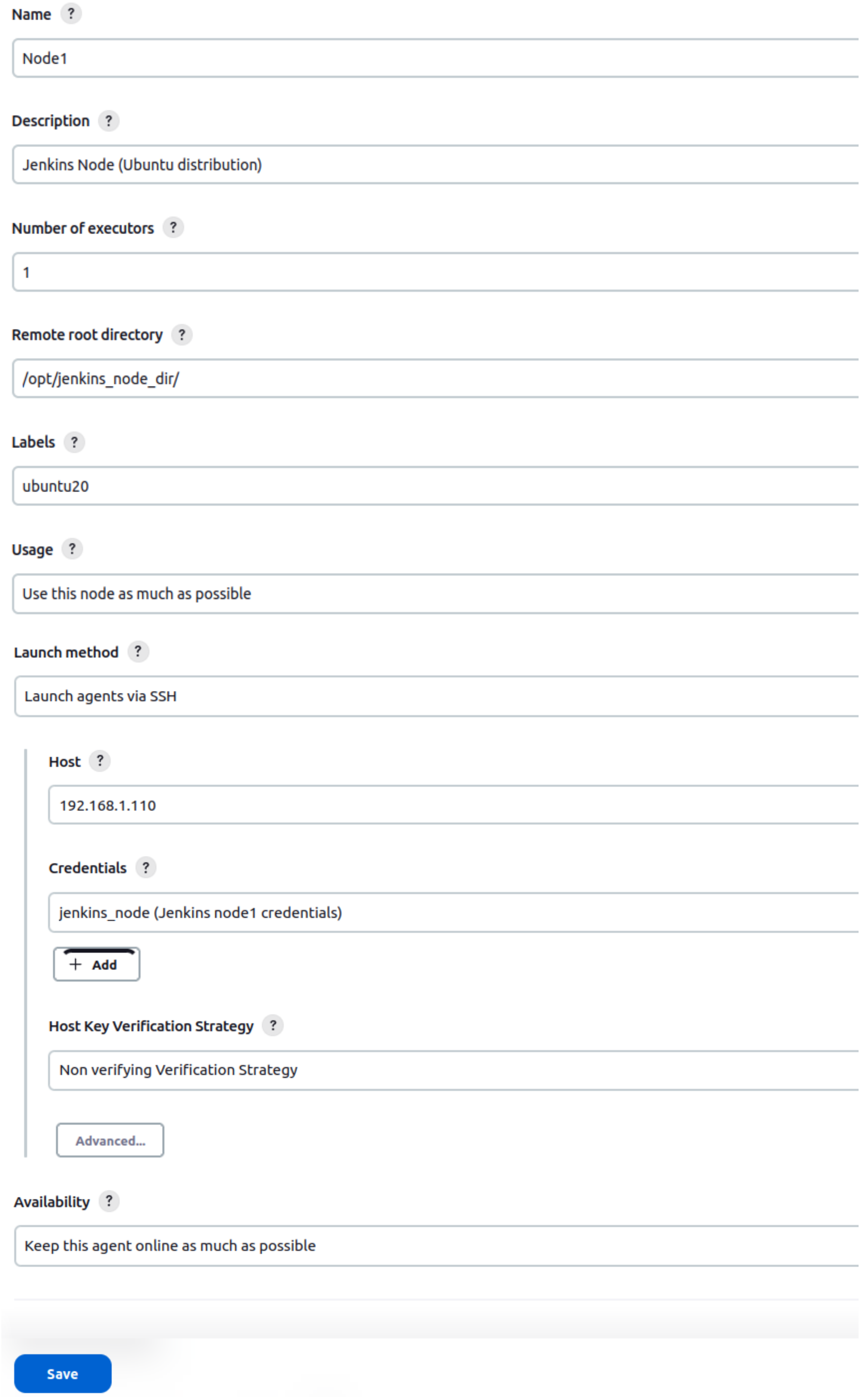
New node

Node name: Node1

Type

● Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.



Name ?

Node1

Description ?

Jenkins Node (Ubuntu distribution)

Number of executors ?

1

Remote root directory ?

/opt/jenkins_node_dir/

Labels ?

ubuntu20

Usage ?

Use this node as much as possible

Launch method ?

Launch agents via SSH

Host ?

192.168.1.110

Credentials ?

jenkins_node (Jenkins node1 credentials)

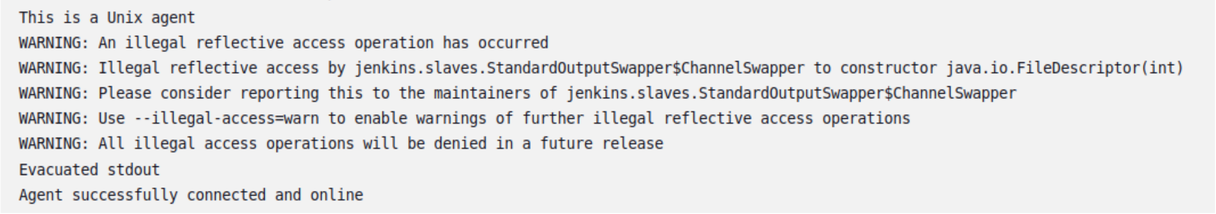
Host Key Verification Strategy ?

Non verifying Verification Strategy

Availability ?

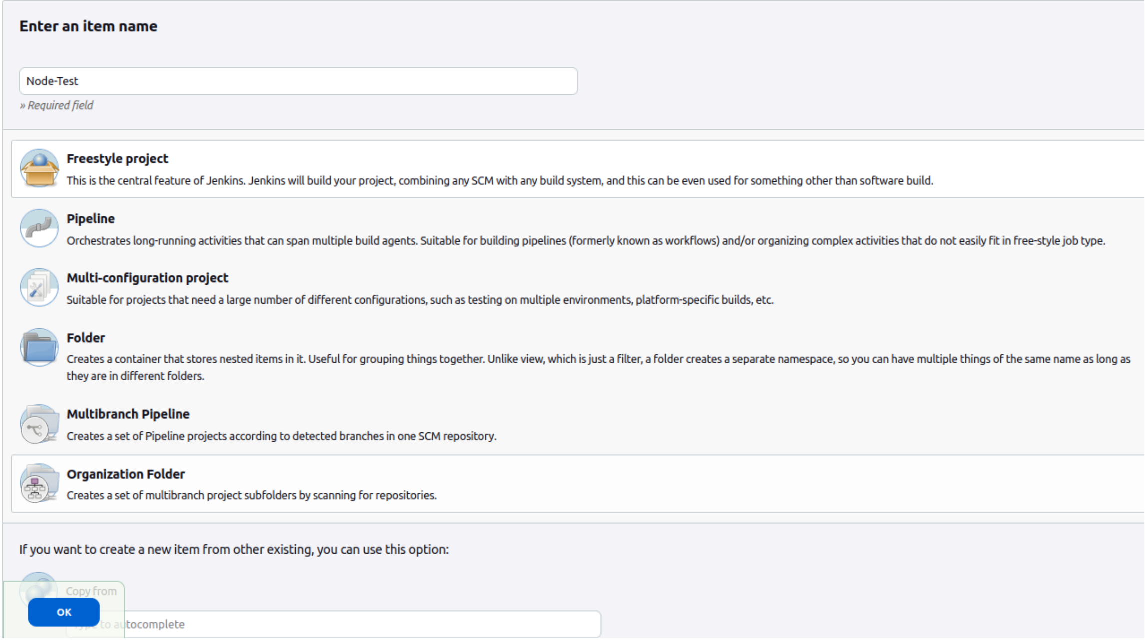
Keep this agent online as much as possible

I check a log about the node (agent successfully connected and online):



```
This is a Unix agent
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by jenkins.slaves.StandardOutputSwapper$ChannelSwapper to constructor java.io.FileDescriptor(int)
WARNING: Please consider reporting this to the maintainers of jenkins.slaves.StandardOutputSwapper$ChannelSwapper
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Evacuated stdout
Agent successfully connected and online
```

29) I create a new job to execute some command on the node.



Enter an item name

Node-Test

» *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.

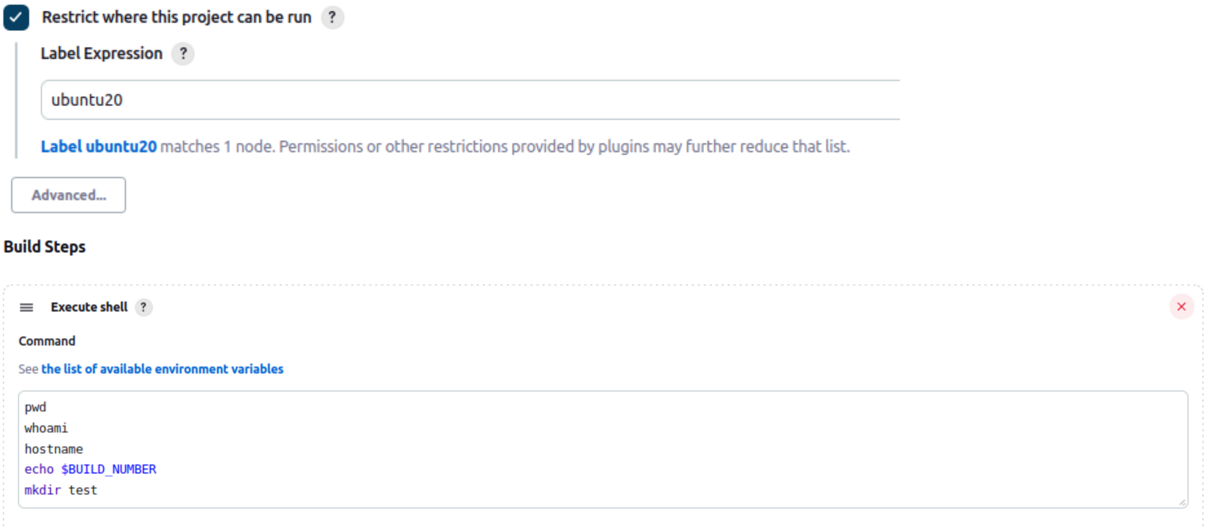
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

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



☑ Restrict where this project can be run

Label Expression: ubuntu20

Label ubuntu20 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Build Steps

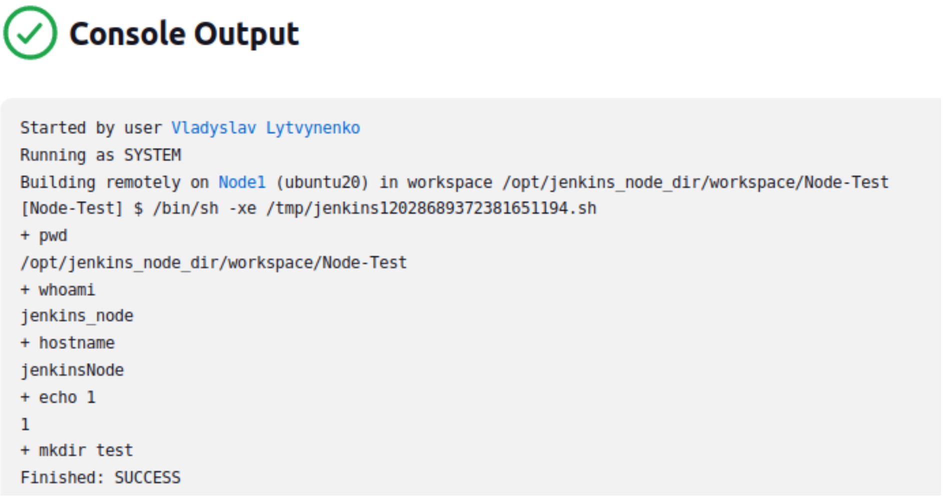
Execute shell

Command

See **the list of available environment variables**

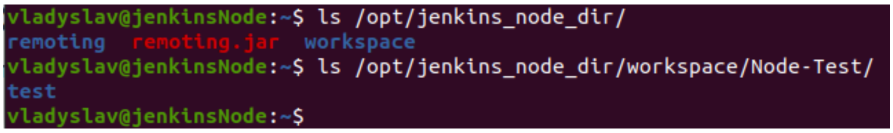
```
pwd
whoami
hostname
echo $BUILD_NUMBER
mkdir test
```

30) I run the job and check execution results.



Console Output

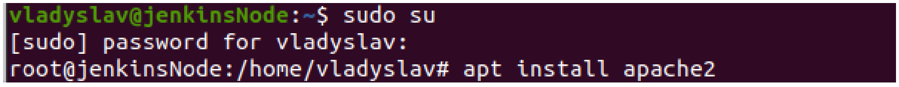
```
Started by user Vladyslav Lytvynenko
Running as SYSTEM
Building remotely on Node1 (ubuntu20) in workspace /opt/jenkins_node_dir/workspace/Node-Test
[Node-Test] $ /bin/sh -xe /tmp/jenkins1202868937238165l194.sh
+ pwd
/opt/jenkins_node_dir/workspace/Node-Test
+ whoami
jenkins_node
+ hostname
jenkinsNode
+ echo 1
1
+ mkdir test
Finished: SUCCESS
```



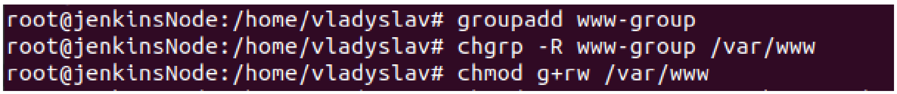
```
vladyslav@jenkinsNode:~$ ls /opt/jenkins_node_dir/
remoting  remoting.jar  workspace
vladyslav@jenkinsNode:~$ ls /opt/jenkins_node_dir/workspace/Node-Test/
test
vladyslav@jenkinsNode:~$
```

31) I install apache on the node machine, configure access, configure the previous job to deploy files to the server from the repository.

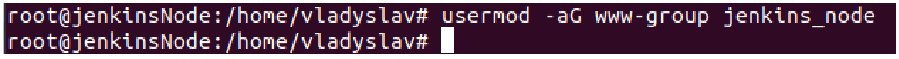
Node machine configuration:



```
vladyslav@jenkinsNode:~$ sudo su
[sudo] password for vladyslav:
root@jenkinsNode:/home/vladyslav# apt install apache2
```

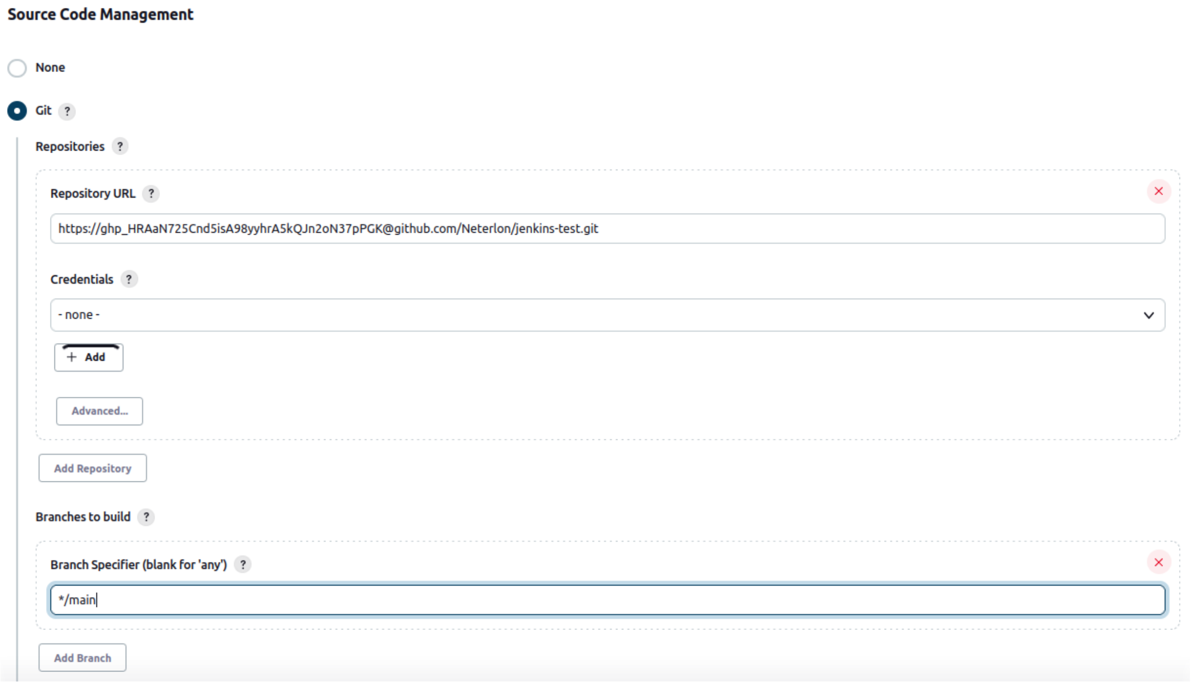


```
root@jenkinsNode:/home/vladyslav# groupadd www-group
root@jenkinsNode:/home/vladyslav# chgrp -R www-group /var/www
root@jenkinsNode:/home/vladyslav# chmod g+rw /var/www
```



```
root@jenkinsNode:/home/vladyslav# usermod -aG www-group jenkins_node
root@jenkinsNode:/home/vladyslav#
```

Job configuration:



Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?

https://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@github.com/Neterlon/jenkins-test.git

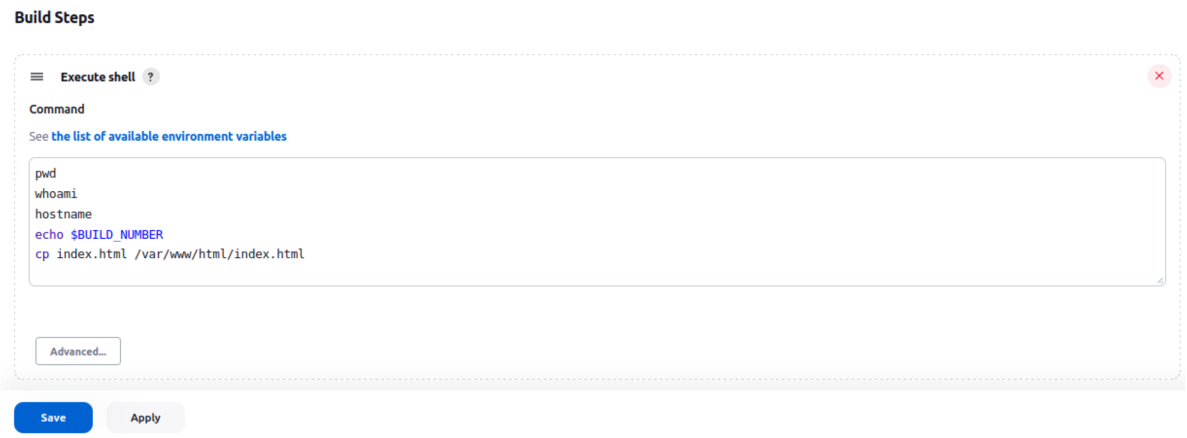
Credentials ?

- none -

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main



Build Steps

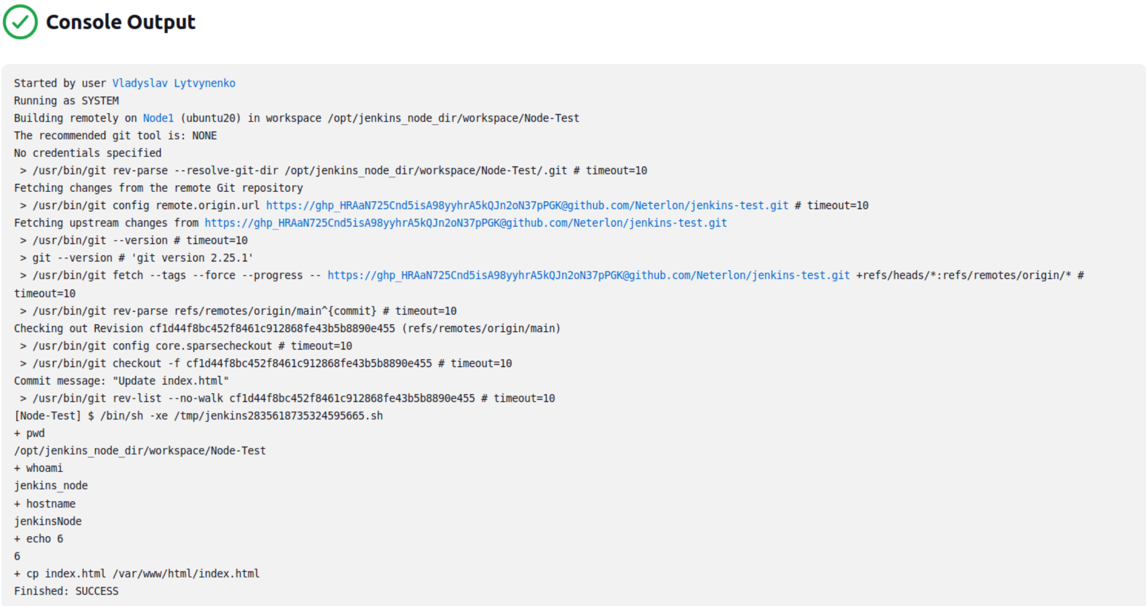
Execute shell ?

Command

See the list of available environment variables

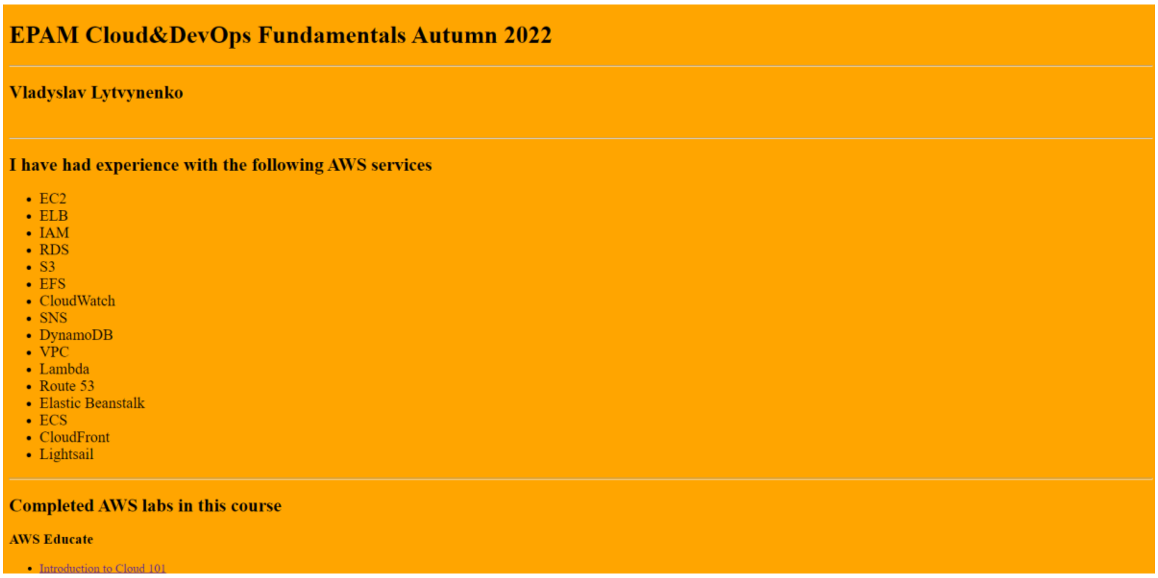
```
pwd
whoami
hostname
echo $BUILD_NUMBER
cp index.html /var/www/html/index.html
```

32) I start the job, check results.



Console Output

```
Started by user Vladyslav Lytvynenko
Running as SYSTEM
Building remotely on Node1 (ubuntu20) in workspace /opt/jenkins_node_dir/workspace/Node-Test
The recommended git tool is: NONE
No credentials specified
 > /usr/bin/git rev-parse --resolve-git-dir /opt/jenkins_node_dir/workspace/Node-Test/.git # timeout=10
Fetching changes from the remote Git repository
 > /usr/bin/git config remote.origin.url https://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@github.com/Neterlon/jenkins-test.git # timeout=10
Fetching upstream changes from https://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@github.com/Neterlon/jenkins-test.git
 > /usr/bin/git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > /usr/bin/git fetch --tags --force --progress -- https://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@github.com/Neterlon/jenkins-test.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
 > /usr/bin/git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision cf1d44f8bc452f8461c912868fe43b5b8890e455 (refs/remotes/origin/main)
 > /usr/bin/git config core.sparsecheckout # timeout=10
 > /usr/bin/git checkout -f cf1d44f8bc452f8461c912868fe43b5b8890e455 # timeout=10
Commit message: "Update index.html"
 > /usr/bin/git rev-list --no-walk cf1d44f8bc452f8461c912868fe43b5b8890e455 # timeout=10
[Node-Test] $ /bin/sh -xe /tmp/jenkins2835618735324595665.sh
+ pwd
/opt/jenkins_node_dir/workspace/Node-Test
+ whoami
jenkins_node
+ hostname
jenkinsNode
+ echo 6
6
+ cp index.html /var/www/html/index.html
Finished: SUCCESS
```



EPAM Cloud&DevOps Fundamentals Autumn 2022

Vladyslav Lytvynenko

I have had experience with the following AWS services

- EC2
- ELB
- IAM
- RDS
- S3
- EFS
- CloudWatch
- SNS
- DynamoDB
- VPC
- Lambda
- Route 53
- Elastic Beanstalk
- ECS
- CloudFront
- Lightsail

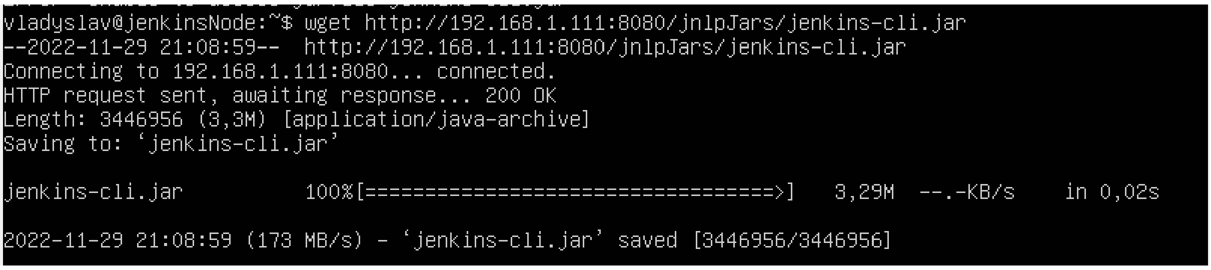
Completed AWS labs in this course

AWS Educate

- Introduction to Cloud 101

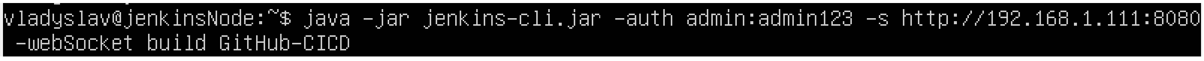
Jenkins CLI

33) I download Jenkins CLI.

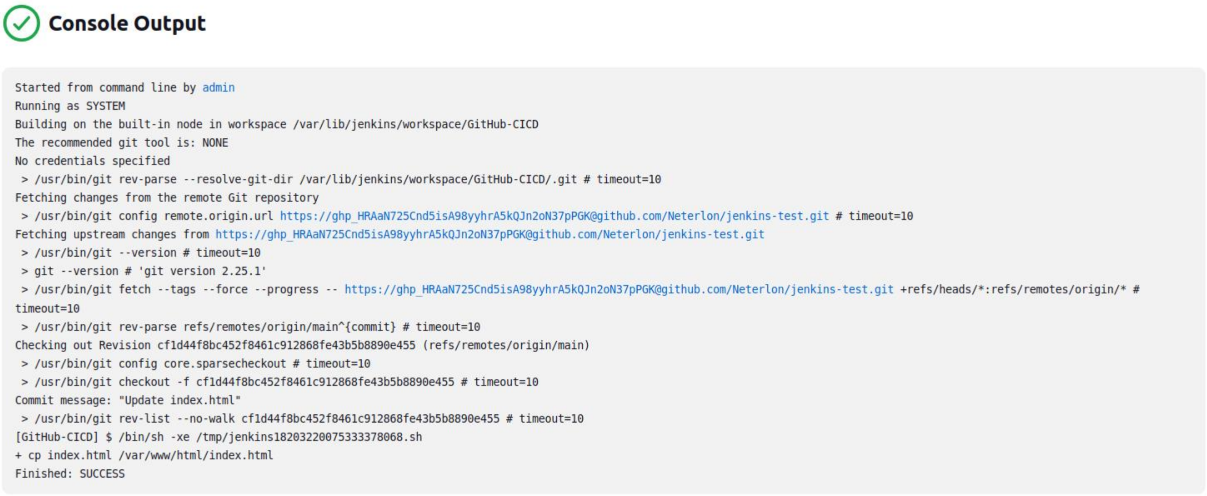


```
vladyslav@jenkinsNode:~$ wget http://192.168.1.111:8080/jnlpJars/jenkins-cli.jar
--2022-11-29 21:08:59--  http://192.168.1.111:8080/jnlpJars/jenkins-cli.jar
Connecting to 192.168.1.111:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3446956 (3,3M) [application/java-archive]
Saving to: ‘jenkins-cli.jar’

jenkins-cli.jar          100%[=================================>]   3,29M  --.-KB/s    in 0,02s

2022-11-29 21:08:59 (173 MB/s) - ‘jenkins-cli.jar’ saved [3446956/3446956]
```

34) I start a job via CLI and check the result.



```
vladyslav@jenkinsNode:~$ java -jar jenkins-cli.jar -auth admin:admin123 -s http://192.168.1.111:8080
 -webSocket build GitHub-CICD
```

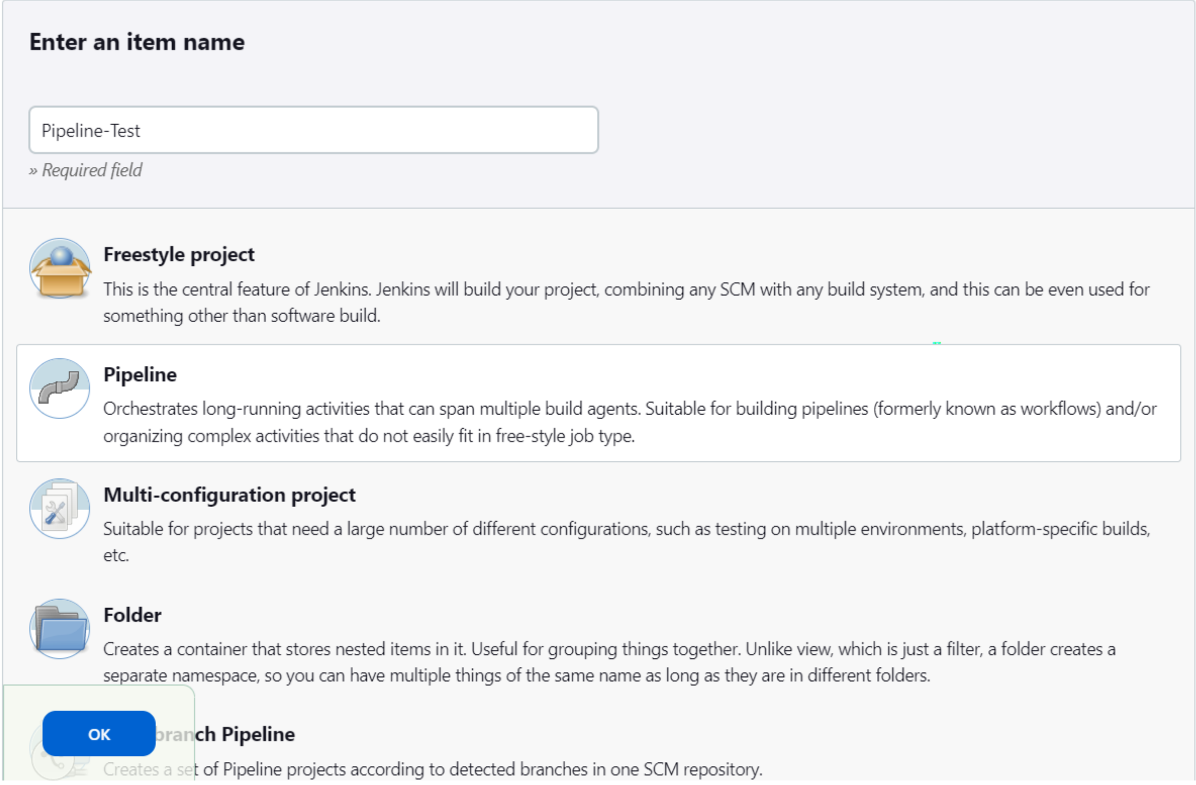


Console Output

```
Started from command line by admin
Running as SYSTEM
Building on the built-in node in workspace /var/lib/jenkins/workspace/GitHub-CICD
The recommended git tool is: NONE
No credentials specified
 > /usr/bin/git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/GitHub-CICD/.git # timeout=10
Fetching changes from the remote Git repository
 > /usr/bin/git config remote.origin.url https://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@github.com/Neterlon/jenkins-test.git # timeout=10
Fetching upstream changes from https://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@github.com/Neterlon/jenkins-test.git
 > /usr/bin/git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > /usr/bin/git fetch --tags --force --progress -- https://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@github.com/Neterlon/jenkins-test.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
 > /usr/bin/git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision cf1d44f8bc452f8461c912868fe43b5b8890e455 (refs/remotes/origin/main)
 > /usr/bin/git config core.sparsecheckout # timeout=10
 > /usr/bin/git checkout -f cf1d44f8bc452f8461c912868fe43b5b8890e455 # timeout=10
Commit message: "Update index.html"
 > /usr/bin/git rev-list --no-walk cf1d44f8bc452f8461c912868fe43b5b8890e455 # timeout=10
[GitHub-CICD] $ /bin/sh -xe /tmp/jenkins18203220075333378068.sh
+ cp index.html /var/www/html/index.html
Finished: SUCCESS
```

Jenkins Pipeline

35) I create a Jenkins Pipeline.



Enter an item name

Pipeline-Test

» Required field

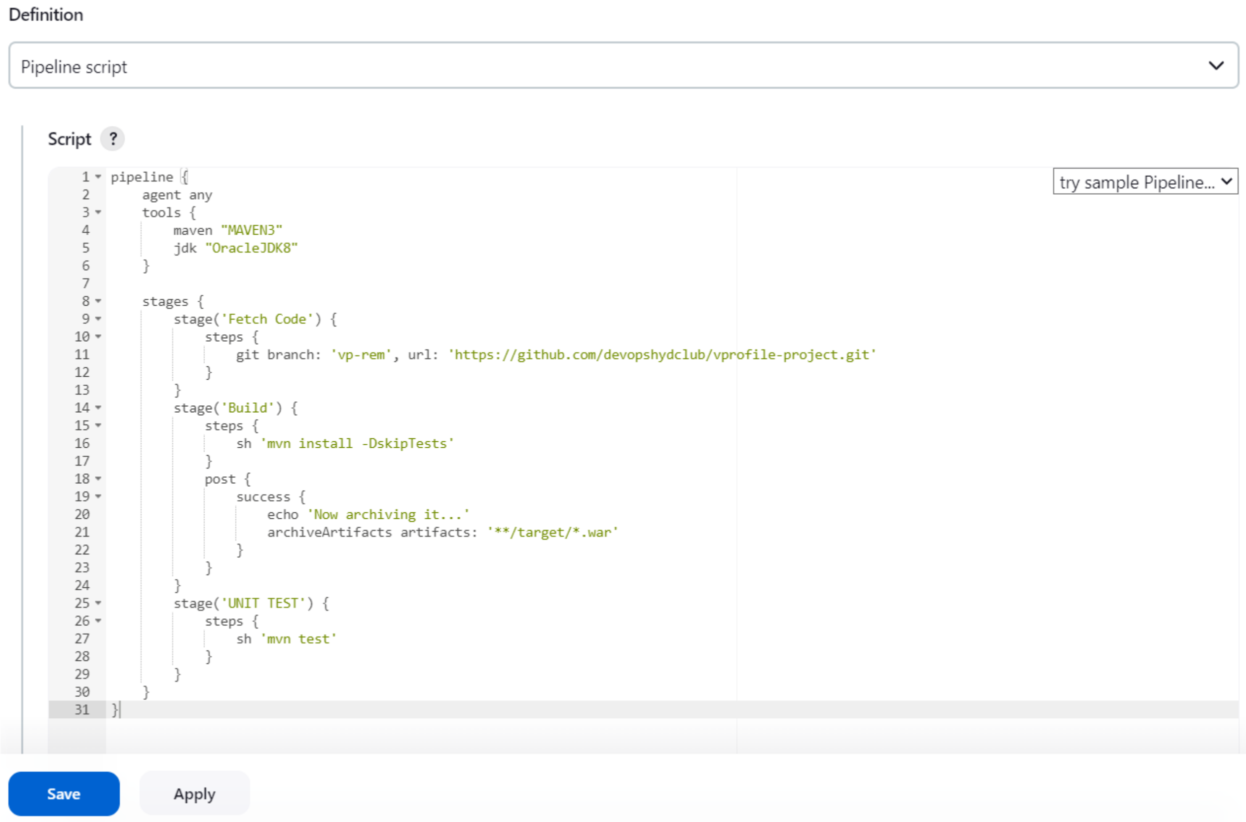
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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.



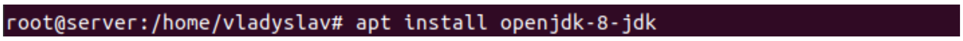
Definition

Pipeline script

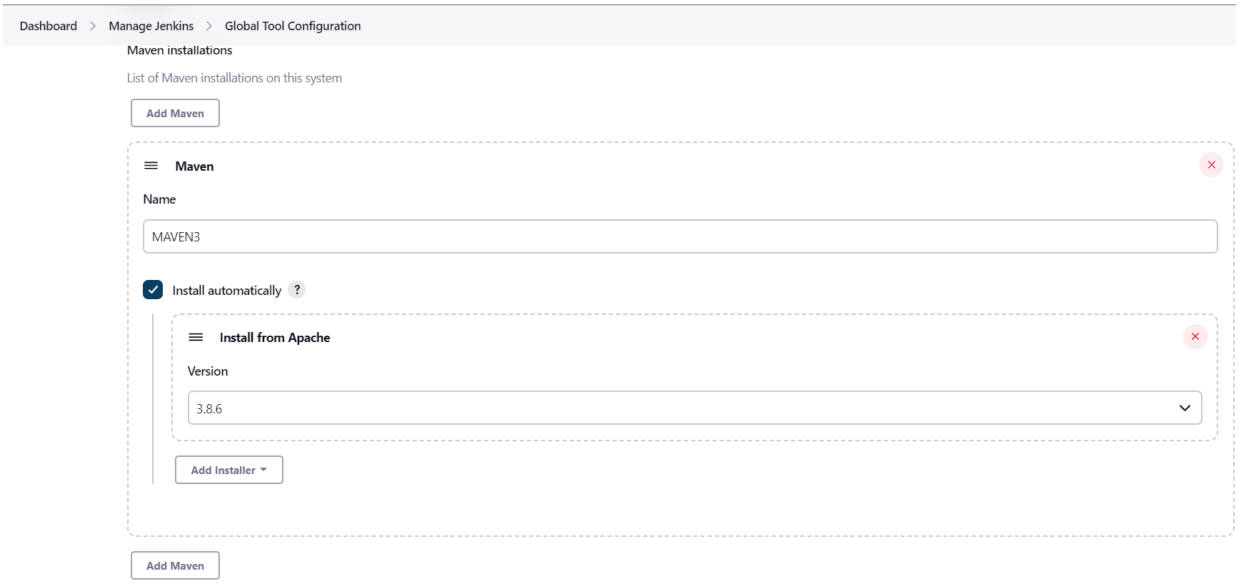
Script ?

```
pipeline {
    agent any
    tools {
        maven "MAVEN3"
        jdk "OracleJDK8"
    }

    stages {
        stage('Fetch Code') {
            steps {
                git branch: 'vp-rem', url: 'https://github.com/devopshydclub/vprofile-project.git'
            }
        }
        stage('Build') {
            steps {
                sh 'mvn install -DskipTests'
            }
            post {
                success {
                    echo 'Now archiving it...'
                    archiveArtifacts artifacts: '**/target/*.war'
                }
            }
        }
        stage('UNIT TEST') {
            steps {
                sh 'mvn test'
            }
        }
    }
}
```

36) I install JDK 8; configure Maven and JDK installation in “Global Tool Configuration”.



```
root@server:/home/vladyslav# apt install openjdk-8-jdk
```



Maven installations

List of Maven installations on this system

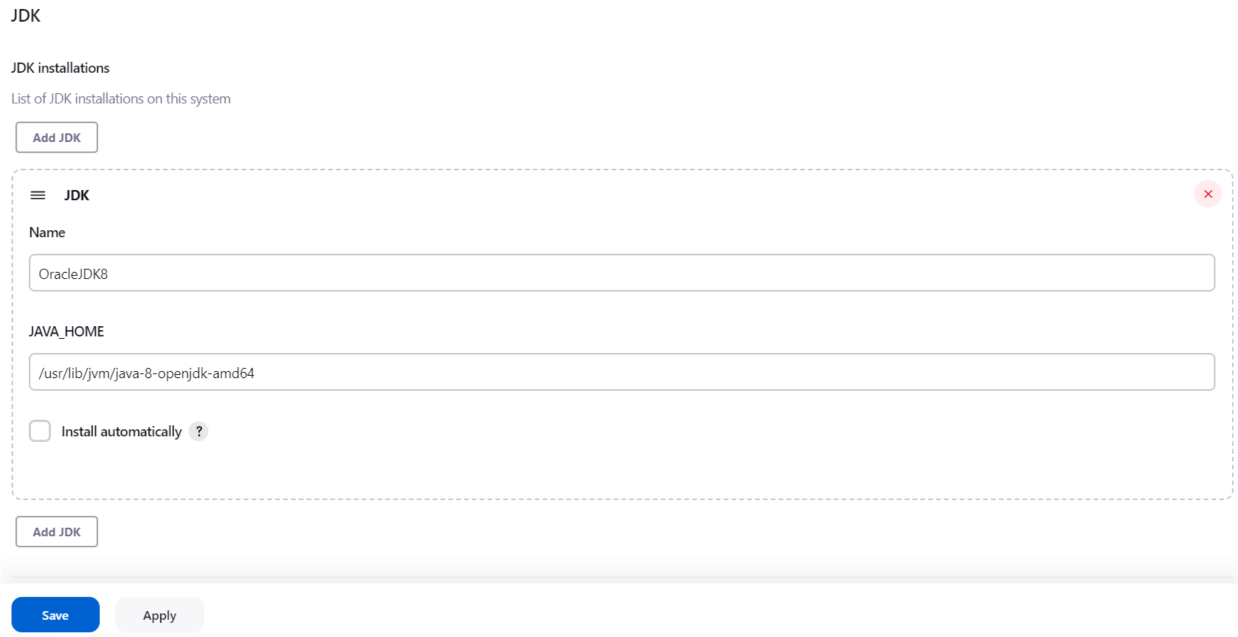
Maven

Name: MAVEN3

☑ Install automatically ?

Install from Apache

Version: 3.8.6



JDK

JDK installations

List of JDK installations on this system

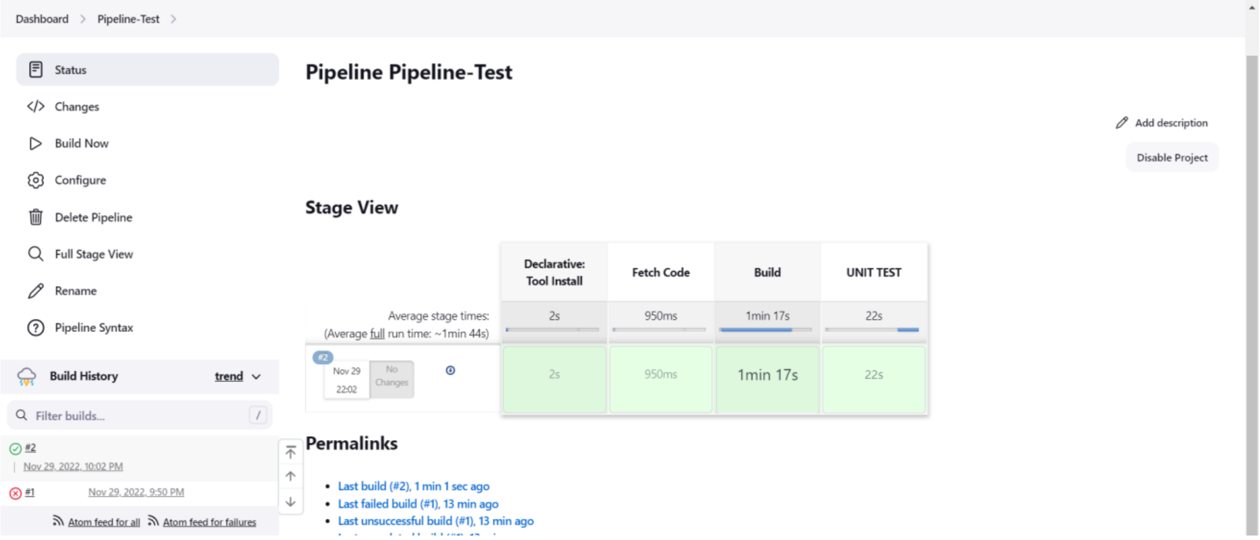
JDK

Name: OracleJDK8

JAVA_HOME: /usr/lib/jvm/java-8-openjdk-amd64

□ Install automatically ?

37) I start the pipeline and check the execution result.



Pipeline Pipeline-Test

Stage View

	Declarative: Tool Install	Fetch Code	Build	UNIT TEST
Average stage times: (Average full run time: ~1min 44s)	2s	950ms	1min 17s	22s
#2 Nov 29 22:02 No Changes	2s	950ms	1min 17s	22s

✓ #2 Nov 29, 2022, 10:02 PM

⊗ #1 Nov 29, 2022, 9:50 PM

Permalinks

- Last build (#2), 1 min 1 sec ago
- Last failed build (#1), 13 min ago
- Last unsuccessful build (#1), 13 min ago