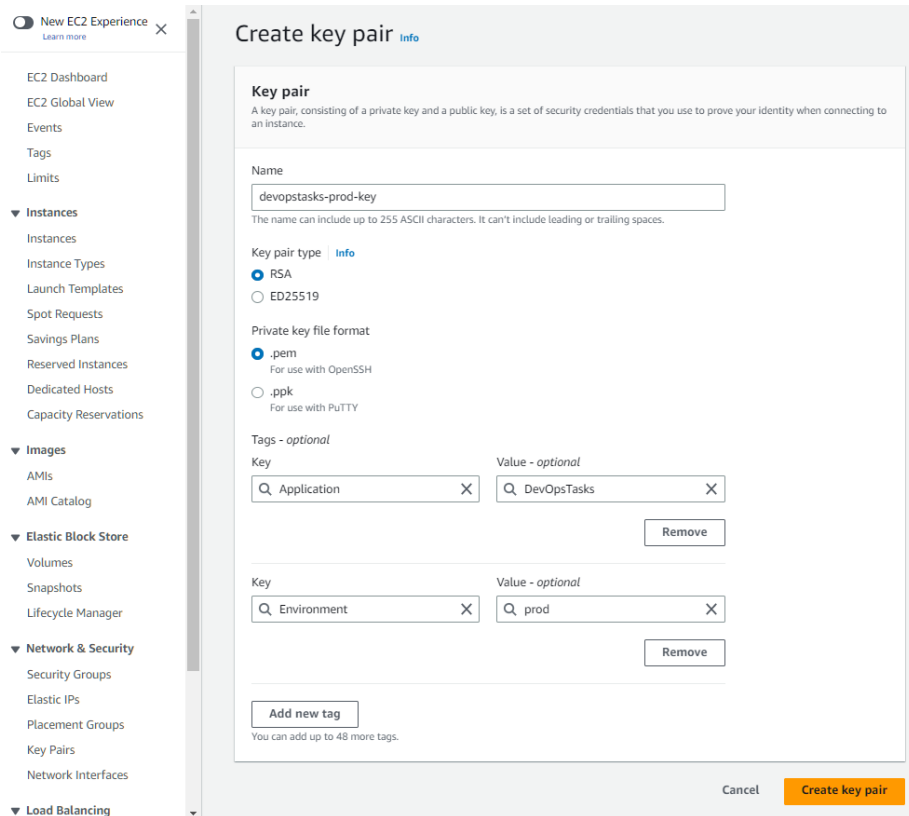


Deployment instruction

1. SSH key pair creation

You need to create a key pair for Ansible SSH connections to Web servers. Open AWS EC2 service, press "Key Pairs" on the left panel and click "Create key pair". Fill the creation form as you see on the screenshot.



The screenshot shows the AWS Management Console interface for creating a new key pair. On the left is a navigation sidebar with categories like EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The 'Key Pairs' option under 'Network & Security' is selected. The main panel is titled 'Create key pair' and contains the following fields and options:

- Name:** A text input field containing 'devopstasks-prod-key'. Below it, a note states: 'The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.'
- Key pair type:** Two radio button options: 'RSA' (selected) and 'ED25519'.
- Private key file format:** Two radio button options: '.pem' (selected, with the note 'For use with OpenSSH') and '.ppk' (with the note 'For use with PuTTY').
- Tags - optional:** A section with two rows of key-value pairs. The first row has 'Key' as 'Application' and 'Value' as 'DevOpsTasks'. The second row has 'Key' as 'Environment' and 'Value' as 'prod'. Each row has a 'Remove' button next to it.
- Add new tag:** A button at the bottom of the tags section with a note: 'You can add up to 48 more tags.'
- Buttons:** 'Cancel' and 'Create key pair' (highlighted in orange) at the bottom right.

Then press Create key pair. Now the private key file is automatically downloaded by your browser. Save the private key file in a safe place.

2. Create a Fork of this repository in your GitHub account
3. Clone your repository locally on the workstation
4. Install and configure AWS CLI on the workstation (see [Installing or updating the latest version of the AWS CLI](#))
5. Install terraform on the workstation (see [Install Terraform](#))
6. Create two parameters in AWS Systems Manager Parameter Store:
 - Username for creating RDS Database.
Type: String

Name: /config/DevOpsTasks_prod/db.username

Value: any username, for example "postgres"

- UserPassword for creating RDS Database

Type: SecureString

Name: /config/DevOpsTasks_prod/db.username

Value: any password, for example "mysecretpas"

7. Setting AWS Region

By default, deployment is using the "eu-central-1" AWS region. If you need to use another region you must change:

- aws_region variable in terraform.tfvars located at DevOps/AWS/EC2/Terraform/Infrastructure/prod
- regions field in dynamic ansible inventory settings file ansible_inventory.aws.aws_ec2.yml located at DevOps/AWS/EC2/Ansible

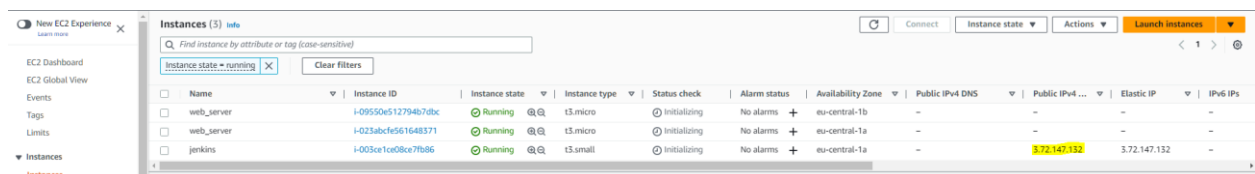
and push these changes to your GitHub repository

8. Creating infrastructure by Terraform

Change directory to DevOps/AWS/EC2/Terraform/Infrastructure/prod and run commands:

```
terraform init
terraform apply (type yes and press enter when it asks "Do you want to perform these actions?")
```

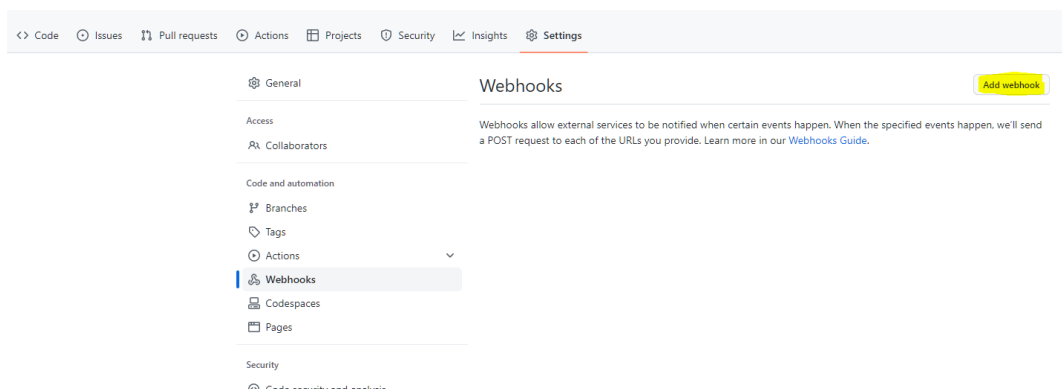
Terraform will create the entire infrastructure but without Jenkins settings. You should see two web and one Jenkins servers in AWS EC2 instances.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
web_server	i-09550a512794b7dbx	Running	t3.micro	Initializing	No alarms	eu-central-1b	-	-	-	-
web_server	i-023abcfef561648371	Running	t3.micro	Initializing	No alarms	eu-central-1a	-	-	-	-
jenkins	i-003ce1ca08ce7fb86	Running	t3.small	Initializing	No alarms	eu-central-1a	-	3.72.147.132	3.72.147.132	-

9. Setting up GitHub webhook.

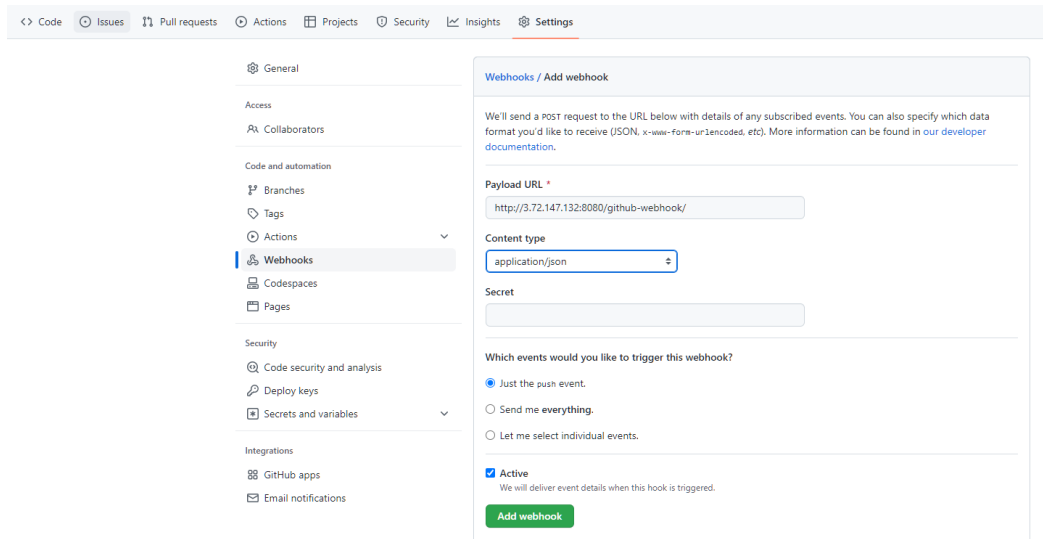
Webhooks allow you to build or set up integrations, such as GitHub Apps or OAuth Apps, which subscribe to certain events on GitHub.com. When one of those events is triggered, GitHub will send a HTTP POST payload to the webhook's configured URL. Webhooks can be used to trigger CI builds. To setting up webhook open GitHub repository settings in your GitHub account. Select Webhooks and click Add webhook.



Set the following fields:

- Payload URL: "http://<jenkins server public IP>:8080/github-webhook/"
- Content type: "application/json"

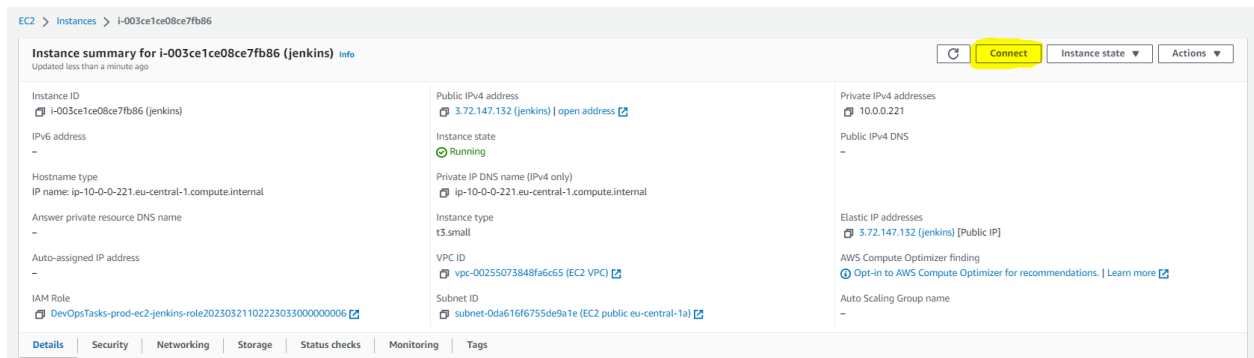
press Add webhook.



The screenshot shows the GitHub Settings page for a repository, specifically the 'Webhooks' section. The left sidebar contains navigation links: Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The 'Webhooks' link is highlighted. The main content area is titled 'Webhooks / Add webhook'. It contains a description of webhooks, a 'Payload URL' field with the value 'http://3.72.147.132:8080/github-webhook/', a 'Content type' dropdown menu set to 'application/json', a 'Secret' field, and a section for 'Which events would you like to trigger this webhook?'. The 'Just the push event.' option is selected. There is also an 'Active' checkbox which is checked. At the bottom, there is a green 'Add webhook' button.

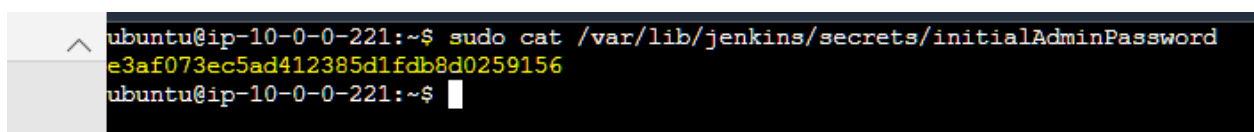
10. Unlocking Jenkins

Connect over SSH to Jenkins server from AWS Console.



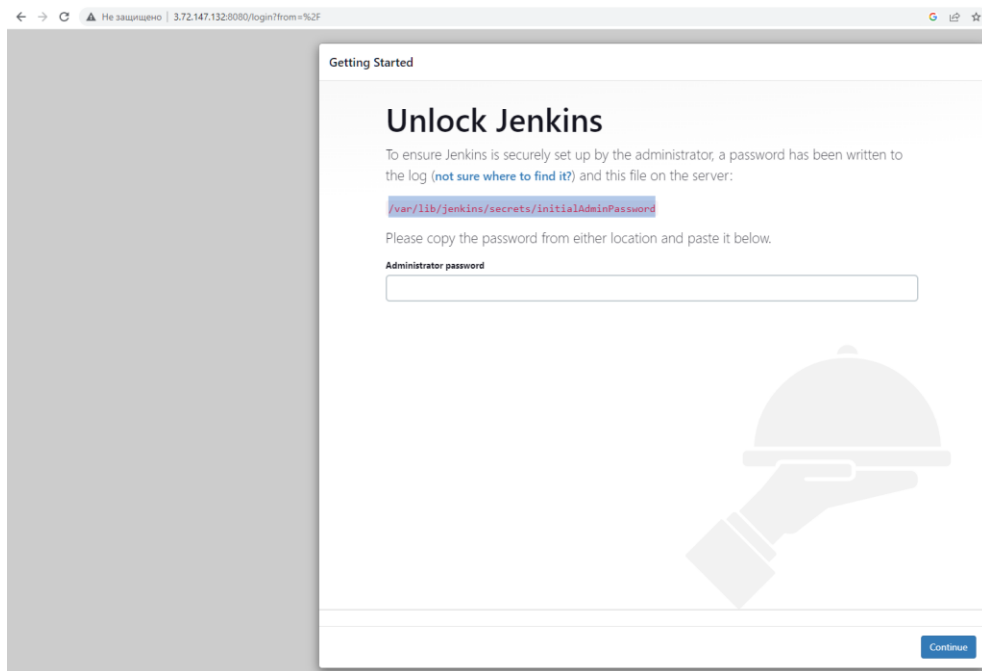
Run command:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

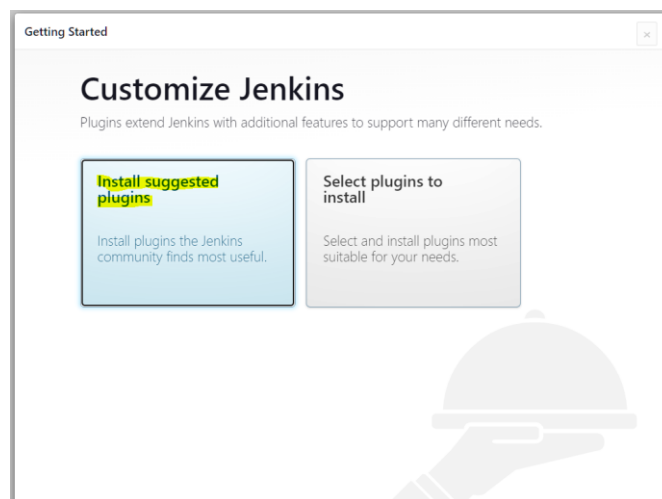


Copy received Administrator password.

Open link <Jenkins server public ip>:8080 in your browser:



Paste the administrator password in a field and press continue



Click Install suggested plugins and wait until installing process is finished.

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestampers	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

Ionicons API

JavaBeans Activation Framework (JAF) API

JavaMail API

bouncycastle API

Instance Identity

Mina SSHD API :: Common

Mina SSHD API :: Core

SSH server

OWASP Markup Formatter

Struts

Token Macro

Build Timeout

Credentials

Trilead API

SSH Credentials

Pipeline: Step API

Plain Credentials

Credentials Binding

SCM API

Pipeline: API

commons-lang3 v3.x Jenkins API

Timestampers

Caffeine API

Script Security

JAXB

SnakeYAML API

Jackson 2 API

commons-text API

Pipeline: Supporting APIs

Plugin Utilities API

Font Awesome API

Bootstrap 5 API

JQuery3 API

ECharts API

- required dependency

Jenkins 2.387.1

Fill in all fields to create a user and press «Save and Continue»

Getting Started

Create First Admin User

Username

devops

Password

...

Confirm password

...

Full name

dev ops

E-mail address

devops@example.com

Jenkins 2.387.1

Skip and continue as admin

Save and Continue

Getting Started

Instance Configuration

Jenkins URL:

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 2.387.1

Not nowSave and Finish

Click «Save and Finish»

11. Installing Jenkins plugins

Open Manage Jenkins -> Manage Plugins -> Available plugins

Select plugins: Docker Pipeline, Ansible and install them.

Jenkins

Search (CTRL+K)

dev ops

log out

Dashboard > Manage Jenkins > Plugin Manager

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Plugins

ansible

Install	Name	Released
<input checked="" type="checkbox"/>	<div>Docker Pipeline 563.vd5d2e5c4007f</div> <div>pipeline DevOps Deployment docker</div> <div>Build and use Docker containers from pipelines.</div> <div>This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.</div>	3 mo 19 days ago
<input checked="" type="checkbox"/>	<div>Ansible 174.vfd5323d2b_9d8</div> <div>pipeline External Site/Tool Integrations DevOps Build Tools Deployment</div> <div>Invoke Ansible Ad-Hoc commands and playbooks.</div>	5 hr 26 min ago
<input type="checkbox"/>	<div>Ansible Tower 0.16.0</div> <div>This plugin connects Jenkins with Ansible Tower</div>	2 yr 9 mo ago

Install without restart


Download now and install after restart

Update information obtained: 10 min ago

Check now

Dashboard > Manage Jenkins > Plugin Manager	
Pipeline: Declarative	✓ Success
Pipeline	✓ Success
Java JSON Web Token (JWT)	✓ Success
OkHttp	✓ Success
GitHub API	✓ Success
Git	✓ Success
GitHub	✓ Success
GitHub Branch Source	✓ Success
Pipeline: GitHub Groovy Libraries	✓ Success
Pipeline Graph Analysis	✓ Success
Pipeline: REST API	✓ Success
Pipeline: Stage View	✓ Success
Git	✓ Success
SSH Build Agents	✓ Success
Matrix Authorization Strategy	✓ Success
PAM Authentication	✓ Success
LDAP	✓ Success
Email Extension	✓ Success
Mailer	✓ Success
Loading plugin extensions	✓ Success
Authentication Tokens API	⚠ Downloaded Successfully. Will be activated during the next boot
Docker Commons	⚠ Downloaded Successfully. Will be activated during the next boot
Docker Pipeline	⚠ Downloaded Successfully. Will be activated during the next boot
Ansible	⚠ Downloaded Successfully. Will be activated during the next boot

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

→  Restart Jenkins when installation is complete and no jobs are running

Click "Restart Jenkins when installation is complete and no jobs are running"

12. Setting up Maven

Open Manage Jenkins -> Global Tool Configuration

In the Maven section, click "Add Maven", fill the following fields:

- Name: Maven_3.9.0
- Version: 3.9.0

click Save.

Dashboard > Manage Jenkins > Global Tool Configuration

[Add Tool](#)

Maven

Maven installations

List of Maven installations on this system

[Add Maven](#)

Maven

Name

Maven_3.9.0

☒ Install automatically ⓘ

Install from Apache

Version

3.9.0

[Add Installer](#)

[Add Maven](#)

Ansible

Ansible installations

List of Ansible installations on this system

[Add Ansible](#)

Docker

Docker installations

[Save](#) [Apply](#)

13. Setting up SSH key for authentication on web servers

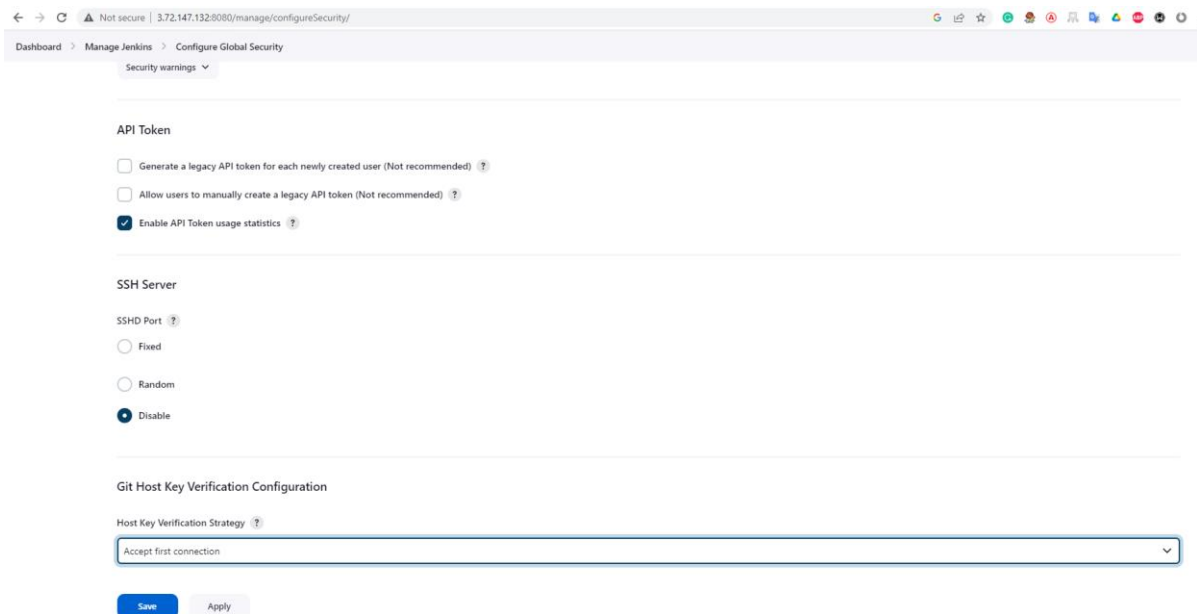
Open Manage Jenkins -> Manage Credentials. Put mouse cursor on Global domain and press Add credentials from drop-down list. In new window fill the following fields:

- Kind: "SSH Username with private key"
- ID: "devopstasks-prod-key"
- Username: "ubuntu"

Select «Enter directly» in private Key section and press Add. Paste private key downloaded on step 1 and click Create.

14. Setting up SSH keys for authentication on GitHub

Open Manage Jenkins -> Configure Global Security. In «Git Host Key Verification» section set «Host Key Verification Strategy» to «Accept first connection» and click Save.



You must set up an SSH key pair to authenticate Jenkins on GitHub. You need to generate this pair. You can use Jenkins SSH connection for key generation. Run command:

```
ssh-keygen -t rsa
```

```
ubuntu@ip-10-0-0-221:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): git-hub
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in git-hub
Your public key has been saved in git-hub.pub
The key's fingerprint is:
SHA256:J3UNx3YSEtNTrCrG02Wru3obkasgWhKBFk6yGoRrPrQ ubuntu@ip-10-0-0-221
The key's randomart image is:
+----[RSA 3072]-----+
|o.o      =+o+.|
|o= o      ** o|
|o.+ .    . .+= |
|o= .    . .+. |
|=. .    S...o+ . |
| E .    o= oo. |
| . . o .. oo. |
| . . . . .o. |
| . . . . .o+ |
| . . . . .o++ |
+----[SHA256]-----+
ubuntu@ip-10-0-0-221:~$ cat git-hub.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCT0pTBgOG8fvOo16vGw01ziilzwjDYE9agc5S10x005Sg3s6jEAQNT/rIpJzc47o52GLjaObQxgZqpY8V2yew4rJOt
KozeU+crTZDCas2Y16vsm+FKFXIfvtn9VVMYkW9eCuqLaDRbWgBFTTT/4/ns9ev2vHcF21kYlndOs8yuqPXmyVUDBjn6KYM8UshUCzyw8eCnLiX6VpkNcAi25Uy5+WA7
NTZVUXQgDSM2GsZJegKJHGkrEtKed9XOI8glqXEzj2y1EE0Yca/cNhwvjCZ5uLidb/Qqw49BbQE= ubuntu@ip-10-0-0-221
ubuntu@ip-10-0-0-221:~$
```

After executing the command, public (git-hub.pub) and private (git-hub) key files will be generated. To display the public key, run the command:


```
cat git-hub.pub
```

Copy public key. Open GitHub user settings and create new SSH key, paste public key in the "Key" field:

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Moderation

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication Keys

	Delete
	Delete

Check out our guide to [generating SSH keys](#) or [troubleshoot common SSH problems](#).

SSH keys / Add new

Title

DevOpsTasks

Key type

Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQgQCt0pTBgOG8fvOol6vGvolziilzwjDYE9agc5SIC
jaObQxgZqpY8V2yew4rJOuGNn8471CZuflCAi75MDbOjC5EzMGQLBdDfZGBNx+Da46
e8FlaVINkPBqjZ8JuWrvJcH1uGOssEwkDynnCtvFKozeU+crTZDCas2Y16vsm+FKPXlfvtn
/ns9evZvHcF2lKY1ndOs8yuqPXmyVUDBjn6KYM8UshUCzyw8eCnLiX6VpkNcAI25Uy5+
DLQnn8FC1yfZE+nzkk4yVKlmmminFi/EQosvLb65Lscu8ajMgfGYITJ8C6oaeBe+7cj9aPca8
M2GsZJegKJHGkrEtKed9XOi8g1qXEzj2y1EE0Yca/cNhwvjCZ5uLidb/Qqw49BbQE= ubu
```

Add SSH key

Code, planning, and automation

Repositories

To display the private key, run the command:

```
cat git-hub
```

You will need the private key on the next step 15.

15. Setting up a job in Jenkins

Open Jenkins dashboard and click "New Item".

Jenkins

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

Fill in the name of the job, select the pipeline and click OK.

Enter an item name

DevOpsTasks

» 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.

OK

Select "This project is parameterized" in General section, click "Add parameter", select "String Parameter" and fill fields as on screenshot.

☐ Preserve stashes from completed builds ?

☒ This project is parameterized ?

String Parameter ?

Name ?
DEVOPSTASKS_ENVIRONMENT

Default Value ?
prod

Description ?

[Plain text] [Preview](#)

☐ Trim the string ?

Add Parameter +

☐ Throttle builds ?

Select "GitHub hook trigger for GITScm polling" checkbox in Build Triggers section.

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) ?

In Pipeline section set:

- Definition: Pipeline script from SCM
- SCM: Git

- Repository URL: an SSH URL to the forked repository, like:
"git@github.com:accountname/repository.git"

Click Add below the credentials field and select Jenkins. In new window fill the following fields:

- Kind: SSH Username with private key
- ID: devopstasks-prod
- Username: your GitHub user name
- Branch Specifier: */main
- Script Path: DevOps/AWS/EC2/Jenkinsfile

Select «Enter directly» in the private Key section and press Add. Paste the private key generated on step 14 and click Add. Select added key in the Credentials field. Click Save.

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

git@github.com:serg.../DevOpsTas...git

Credentials ?

serg...

Add +

Advanced v

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add +

Script Path ?

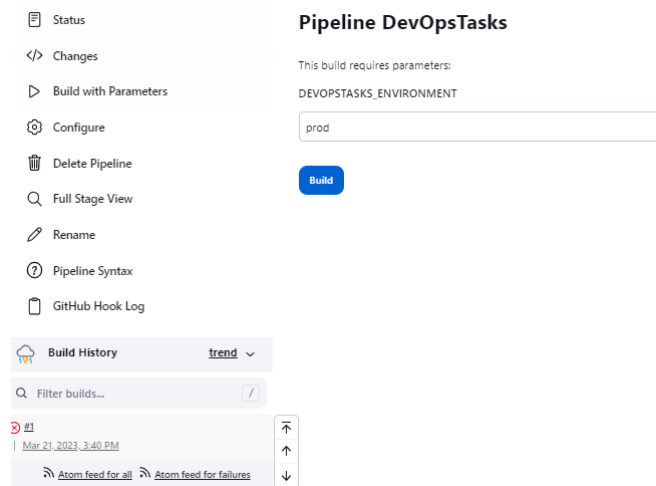
DevOps/AWS/EC2/Jenkinsfile

☒ Lightweight checkout ?

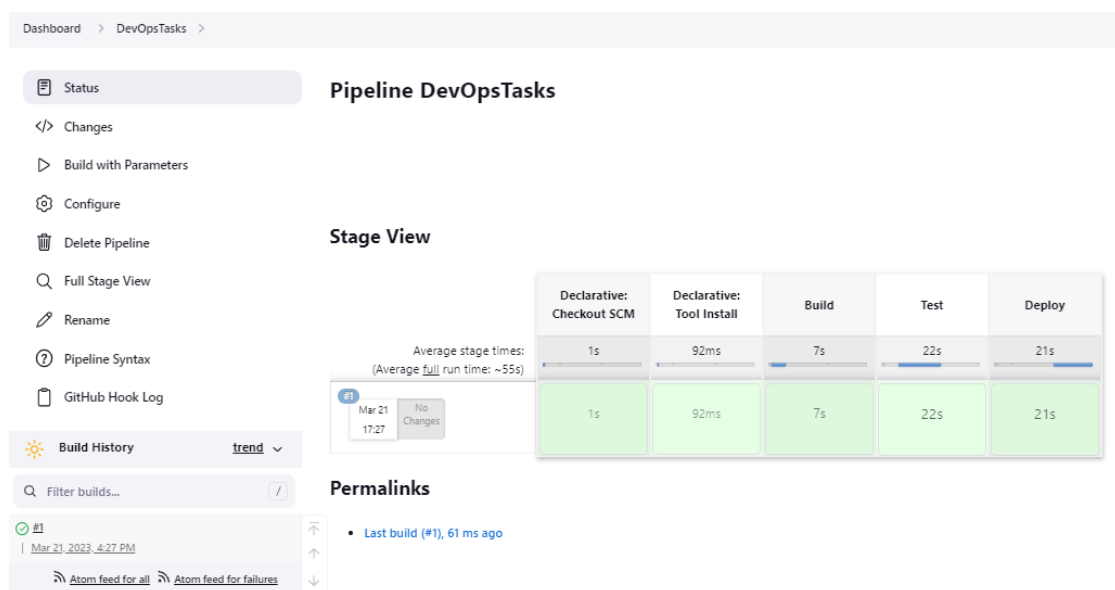
Save Apply

16. Running pipeline in manual mode

Click «Build with Parameters» and press Build



After the build is completed, you should see all stages green:



17. Running pipeline in automatic mode

We have made all settings for automatically running the pipeline on push event in the GitHub repository. You can check it by changing and committing changes in any repository file.

18. Checking the application

Now you can find the load balancer DNS name in the AWS EC2 service:

Limits

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

▼ Load Balancing

Load Balancers

Target Groups

▼ Auto Scaling

Launch Configurations

Auto Scaling Groups

Create Load Balancer Actions

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
DevOpsTasks-prod-nlb	DevOpsTasks-prod-nlb-6c7d...	Active	vpc-09300af824e4356dd	eu-central-1a, eu-centr...	network	March 21, 2023 at 4:02:33 P...	

Load balancer: DevOpsTasks-prod-nlb

DescriptionListenersMonitoringIntegrated servicesTags

Basic Configuration

Name

DevOpsTasks-prod-nlb

ARN

arn:aws:elasticloadbalancing:eu-central-1:273283800303:loadbalancer/net/DevOpsTasks-prod-nlb/6c7d4c8f38f76468

DNS name

DevOpsTasks-prod-nlb-6c7d4c8f38f76468.elb.eu-central-1.amazonaws.com

(A Record)

State

Active

Type

network

Scheme

internet-facing

IP address type

ipv4

Edit IP address type

VPC

vpc-09300af824e4356dd

Availability Zones

subnet-05f3dcc14c58e01af - eu-central-1a

IPv4 address: Assigned by AWS

subnet-0f209031b238e25ac - eu-central-1b

IPv4 address: Assigned by AWS

Edit subnets

You can open this link in a web browser and should see the application:

← → ↻

Not secure

devopstasks-prod-nlb-6c7d4c8f38f76468.elb.eu-central-1.amazonaws.com

This is a simple Java Spring boot application

Database connection parameters:

Url = jdbc:postgresql://db-devopstasks-prod.cfds94vucpp.eu-central-1.rds.amazonaws.com:5432/postgres

Username = postgres

Password = *****

Top 5 movies according to IMDB:

Id	Title	Year	Director
1	The Shawshank Redemption	1994	Frank Darabont
51	The Godfather	1972	Francis Ford Coppola
101	The Dark Knight	2008	Christopher Nolan
151	The Godfather Part II	1974	Francis Ford Coppola
201	12 Angry Men	1957	Sidney Lumet

If you see the movie list, then the database connection has been established

19. Clean up

For cleanup you should:

1. Remove created SSH key from GitHub user;
2. Remove created AWS Key pair;
3. Run "terraform destroy" command from DevOps/AWS/EC2/Terraform/Infrastructure/prod on you workstation.