EPAM DevOps Summer online Graduation Work

Pipeline for building java applications

Done by: Nataliia Remaieva

Motivation

My solution is important because it grants high scalability and management of services in a short time.

Topicality

I have choosen that approach infrastructure because:

1. With the help of Docker we can separate processes to avoid chain of falling infrastructure
2. Nginx helps us to manage http connections
3. Terraform provides easy managing, creation and recovery full our infrastructure

Main aim

The main aim of this project is to make automatic pipeline and create IaC for fast management and scalability.

Tasks

- Create on-cloud pipeline
- Create stages for docker/clone/build
- Create IaC

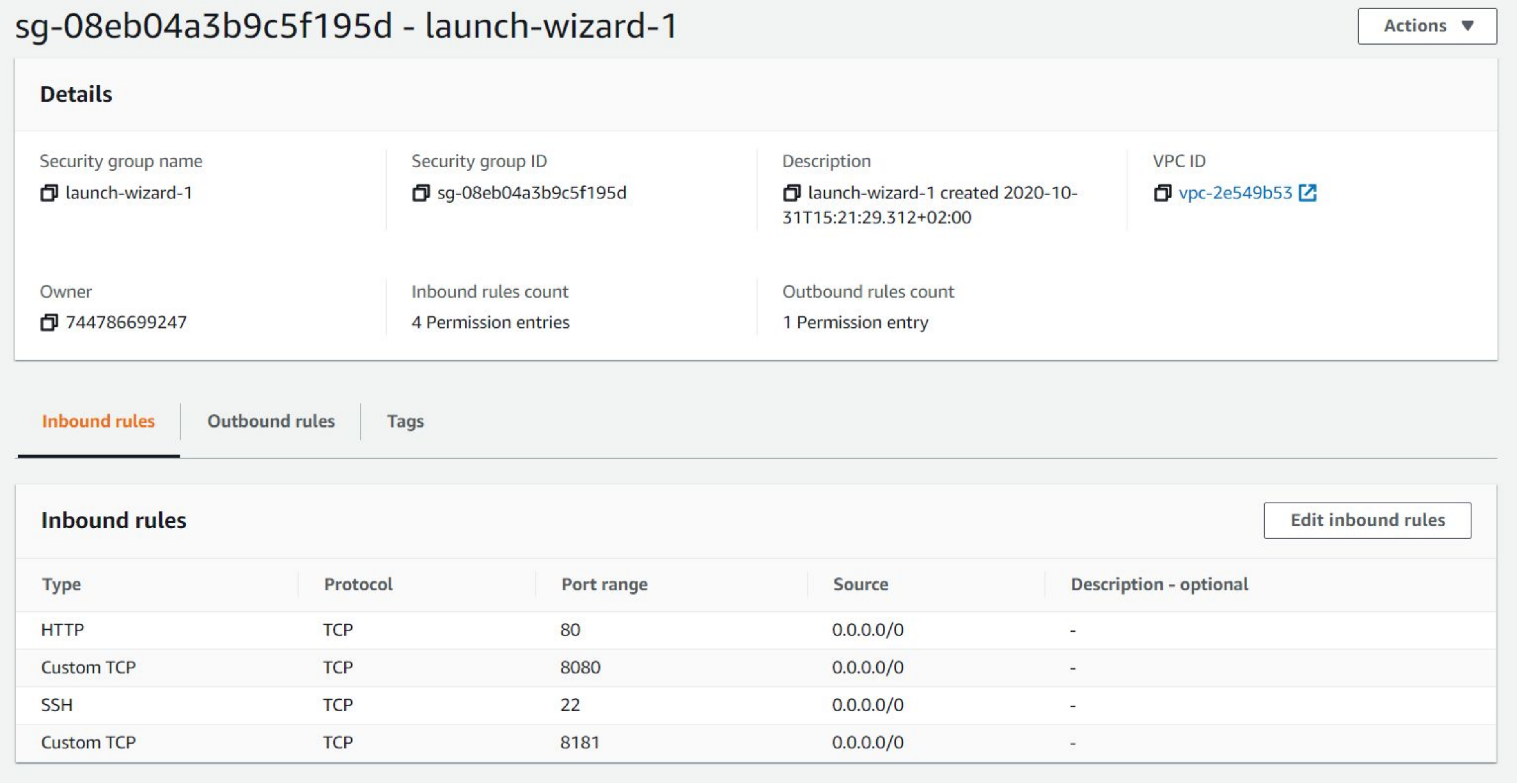
Designing

Technology stack consist of AWS, Nginx, Terraform, Docker and Jenkins.
AWS is the most flexible and all-you-need service among others. And it is the most popular cloud.
Jenkins - for building pipeline.
Nginx - for make proxypass to redirect Jenkins port to port :80.
Docker decompose the infastructure.
Terraform creates IaC.

Realization

Set up and configure instance on AWS was much easier than configure Jenkins master.
The first challenge was to find out how to run all of stages it in the Docker. So i needed to grant privileges Jenkins to run docker. Second was to make IaC. With the help of different tools like terraform import I have created tfstate file for my instance.
So I achieve all my aims on this project.

Demonstration



sg-08eb04a3b9c5f195d - launch-wizard-1

Details

Security group name
launch-wizard-1

Security group ID
sg-08eb04a3b9c5f195d

Description
launch-wizard-1 created 2020-10-31T15:21:29.312+02:00

VPC ID
vpc-2e549b53

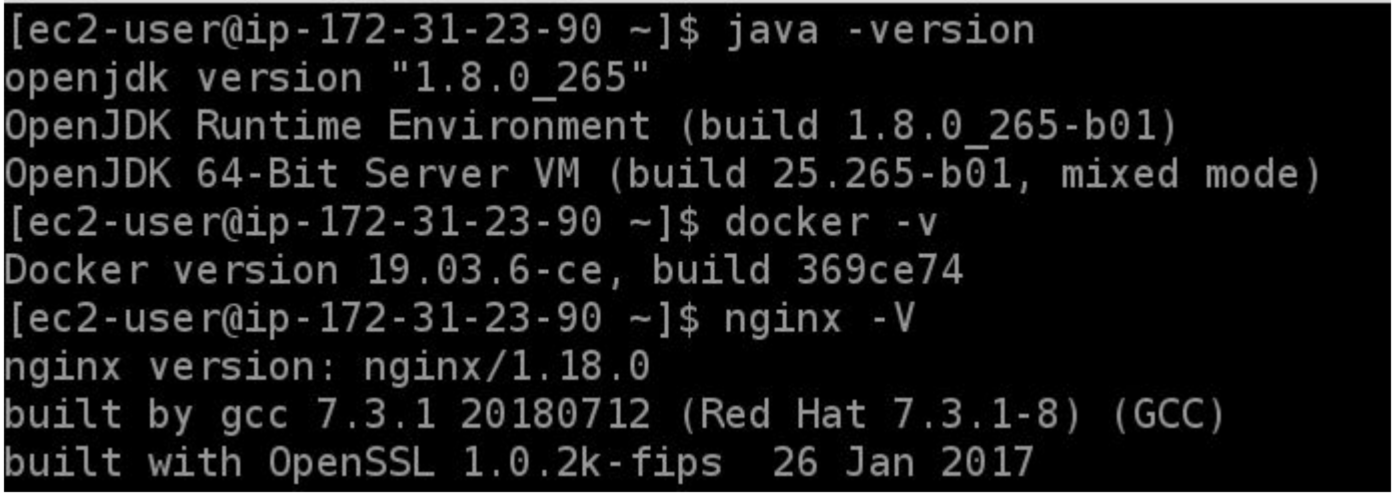
Owner
744786699247

Inbound rules count
4 Permission entries

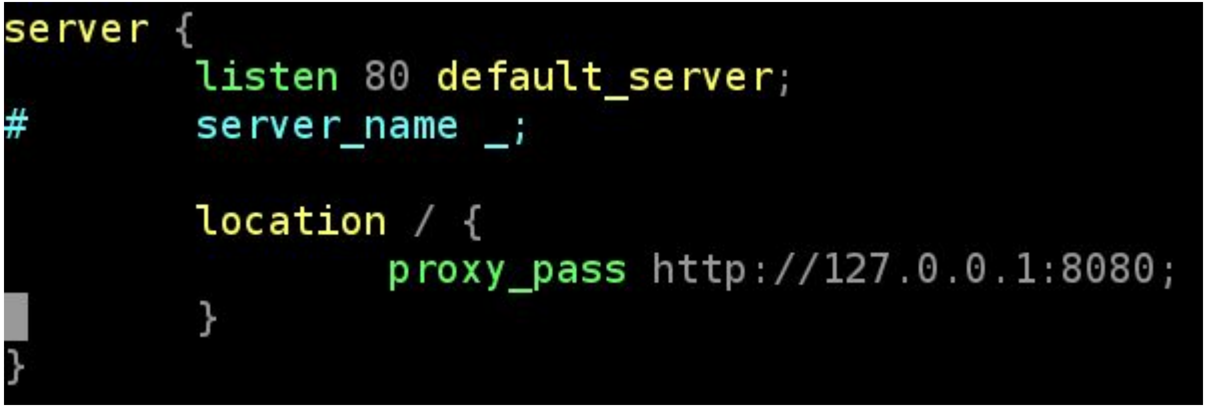
Outbound rules count
1 Permission entry

Inbound rules

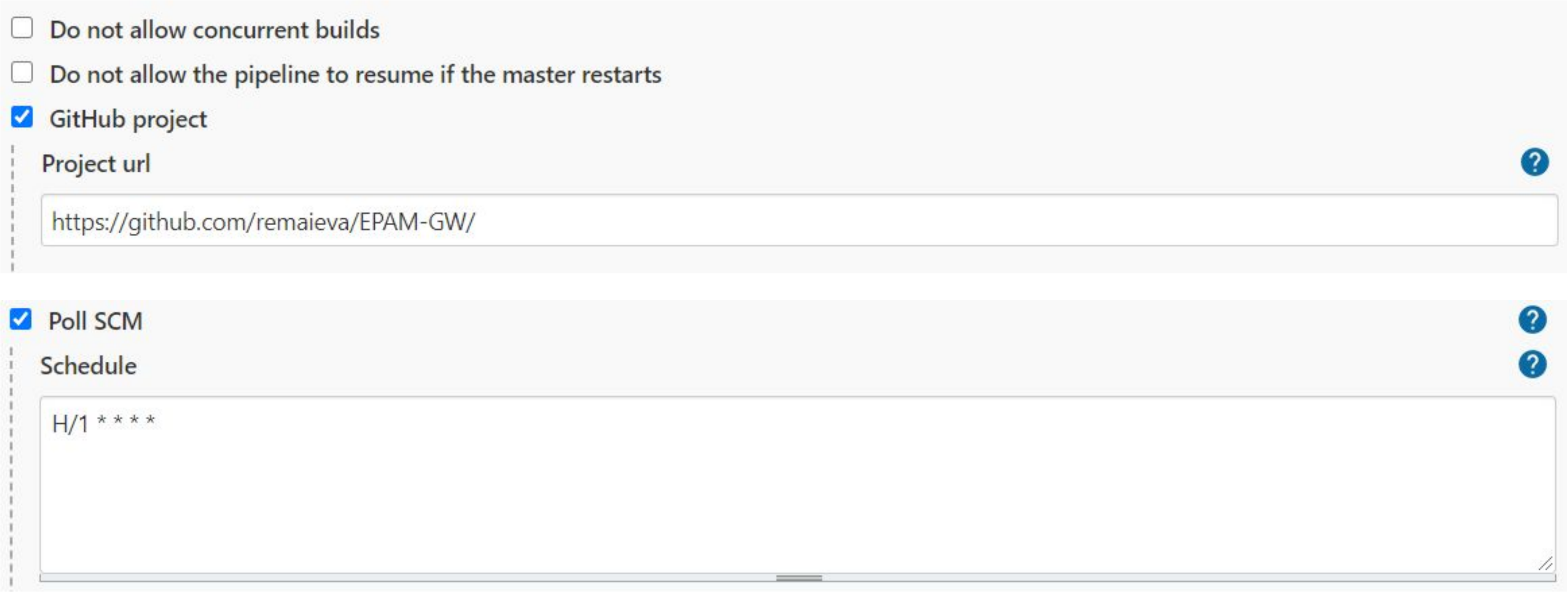
Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
Custom TCP	TCP	8080	0.0.0.0/0	-
SSH	TCP	22	0.0.0.0/0	-
Custom TCP	TCP	8181	0.0.0.0/0	-



```
[ec2-user@ip-172-31-23-90 ~]$ java -version
openjdk version "1.8.0_265"
OpenJDK Runtime Environment (build 1.8.0_265-b01)
OpenJDK 64-Bit Server VM (build 25.265-b01, mixed mode)
[ec2-user@ip-172-31-23-90 ~]$ docker -v
Docker version 19.03.6-ce, build 369ce74
[ec2-user@ip-172-31-23-90 ~]$ nginx -V
nginx version: nginx/1.18.0
built by gcc 7.3.1 20180712 (Red Hat 7.3.1-8) (GCC)
built with OpenSSL 1.0.2k-fips  26 Jan 2017
```



```
server {
        listen 80 default_server;
#       server_name _;

        location / {
                proxy_pass http://127.0.0.1:8080;
        }
}
```



□ Do not allow concurrent builds

□ Do not allow the pipeline to resume if the master restarts

☑ GitHub project

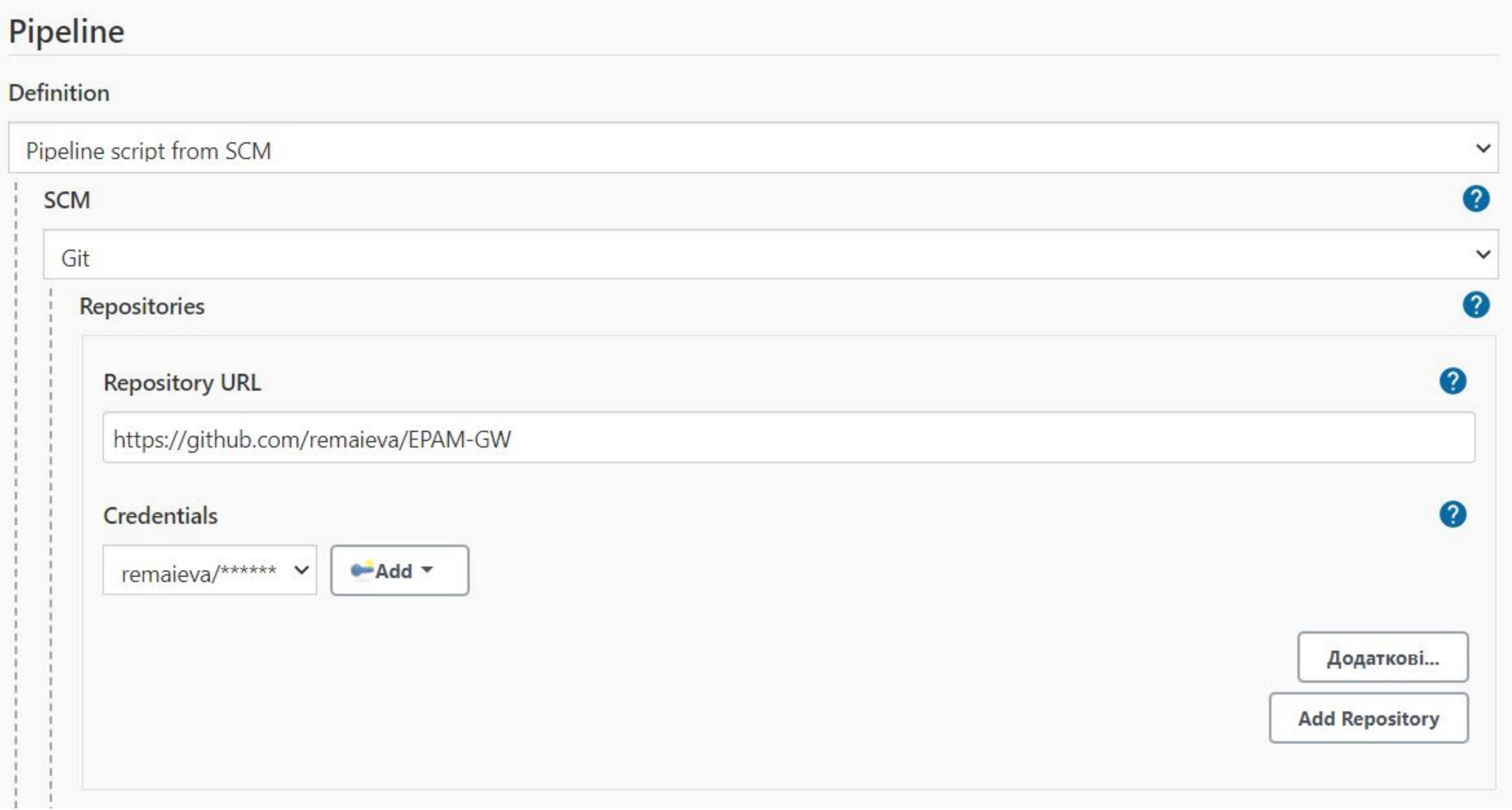
Project url

https://github.com/remaieva/EPAM-GW/

☑ Poll SCM

Schedule

```
H/1 * * * *
```



Pipeline

Definition

Pipeline script from SCM

SCM

Git

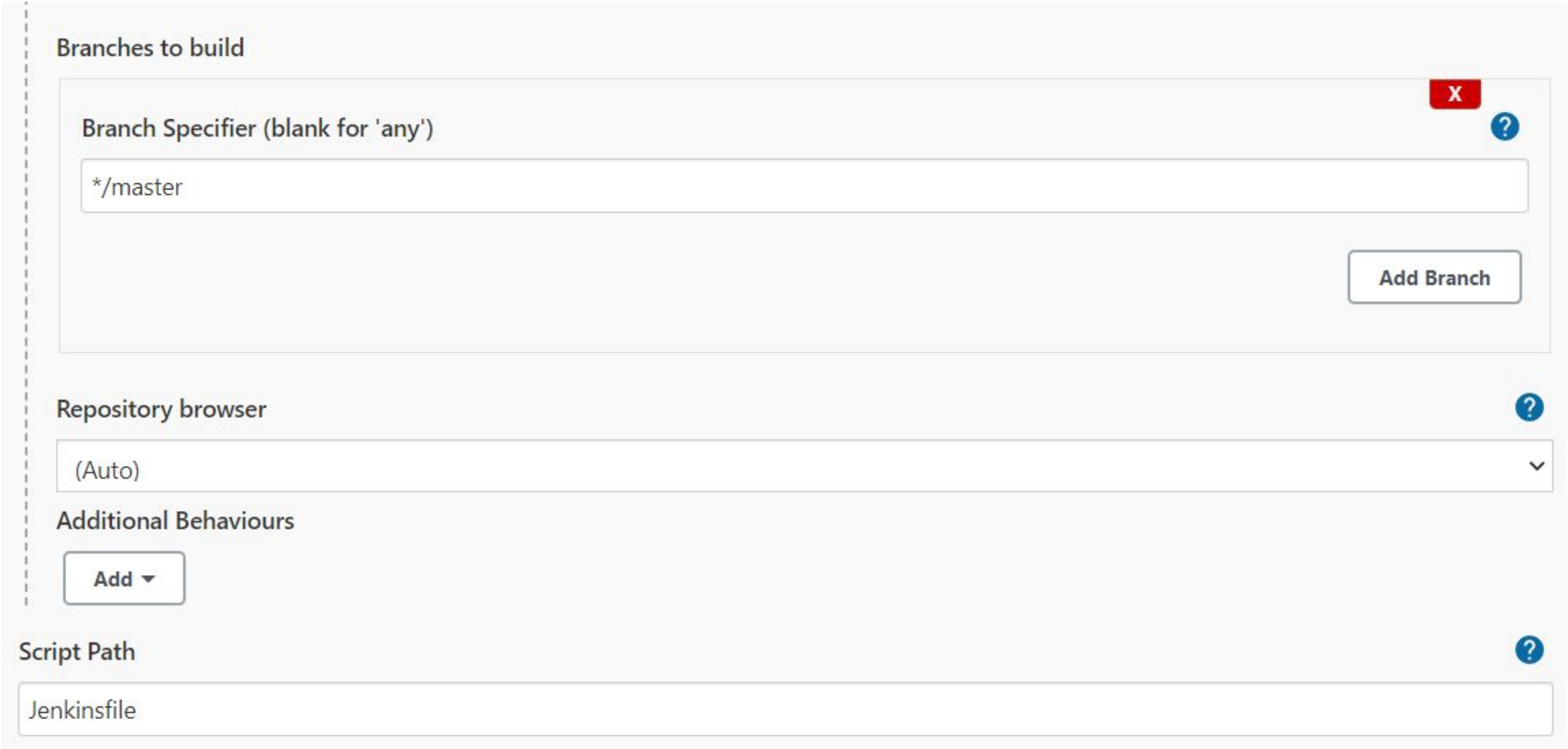
Repositories

Repository URL

https://github.com/remaieva/EPAM-GW

Credentials

remaieva/******



Branches to build

Branch Specifier (blank for 'any')

*/master

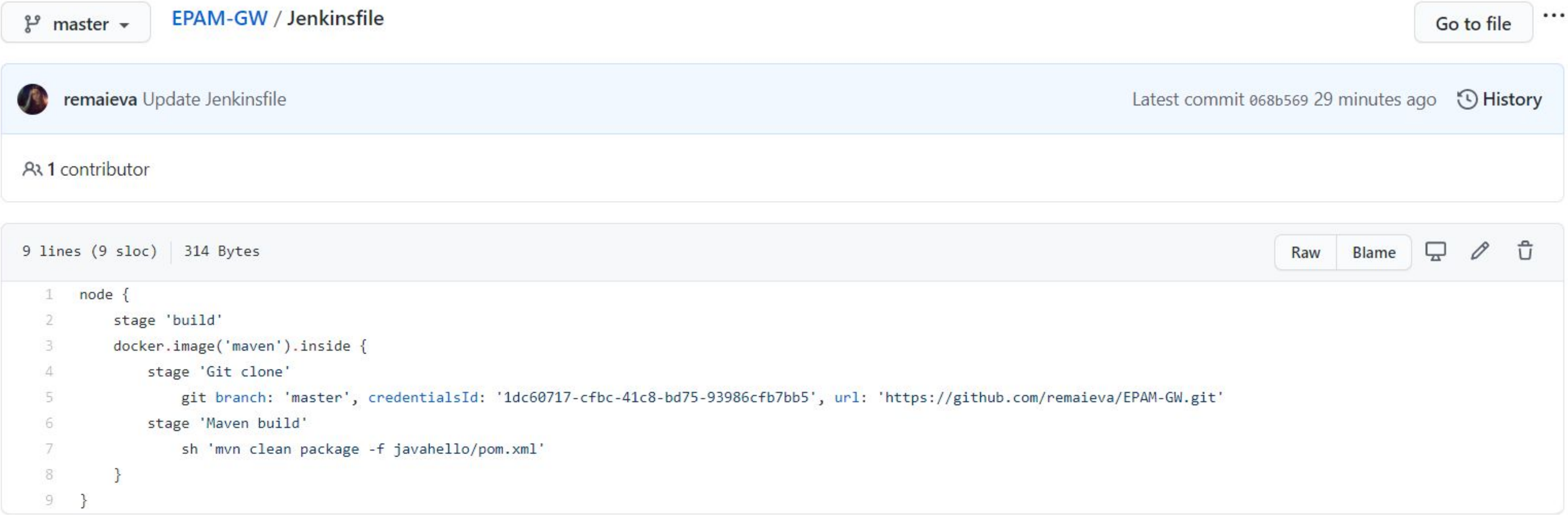
Repository browser

(Auto)

Additional Behaviours

Script Path

Jenkinsfile



EPAM-GW / Jenkinsfile

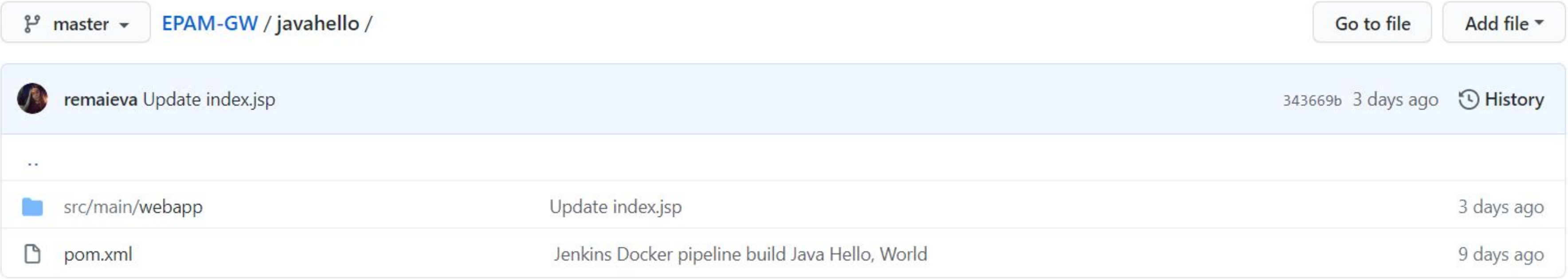
remaieva Update Jenkinsfile

Latest commit 068b569 29 minutes ago

1 contributor

9 lines (9 sloc) | 314 Bytes

```
node {
    stage 'build'
    docker.image('maven').inside {
        stage 'Git clone'
            git branch: 'master', credentialsId: '1dc60717-cfbc-41c8-bd75-93986cfb7bb5', url: 'https://github.com/remaieva/EPAM-GW.git'
        stage 'Maven build'
            sh 'mvn clean package -f javahello/pom.xml'
    }
}
```

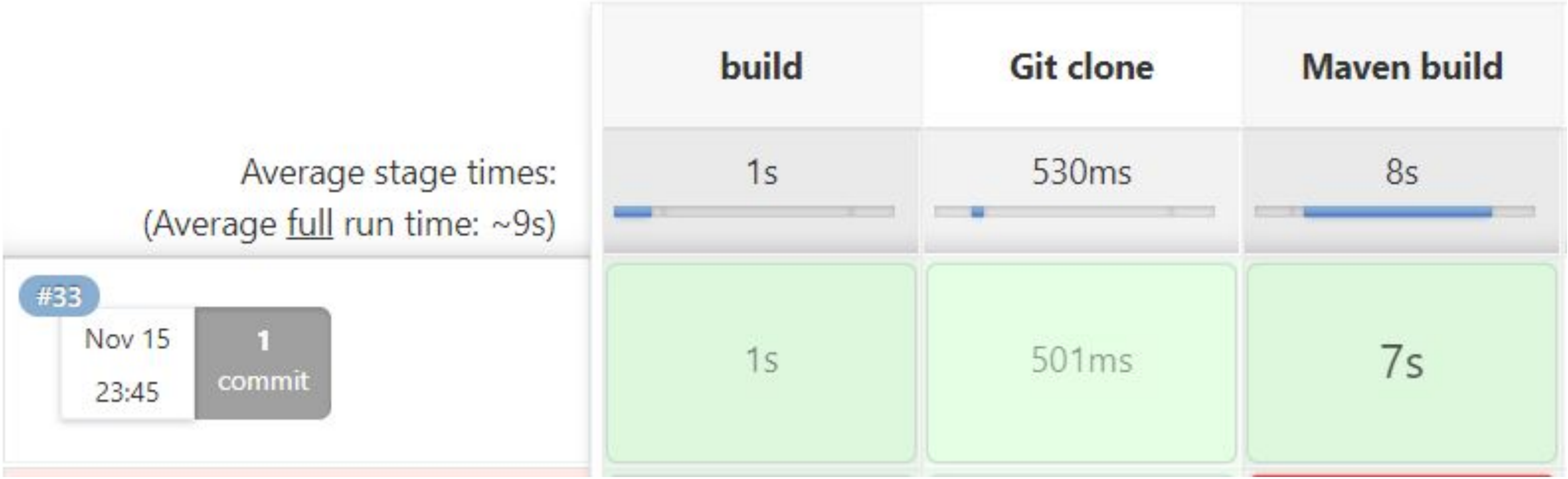


EPAM-GW / javahello /

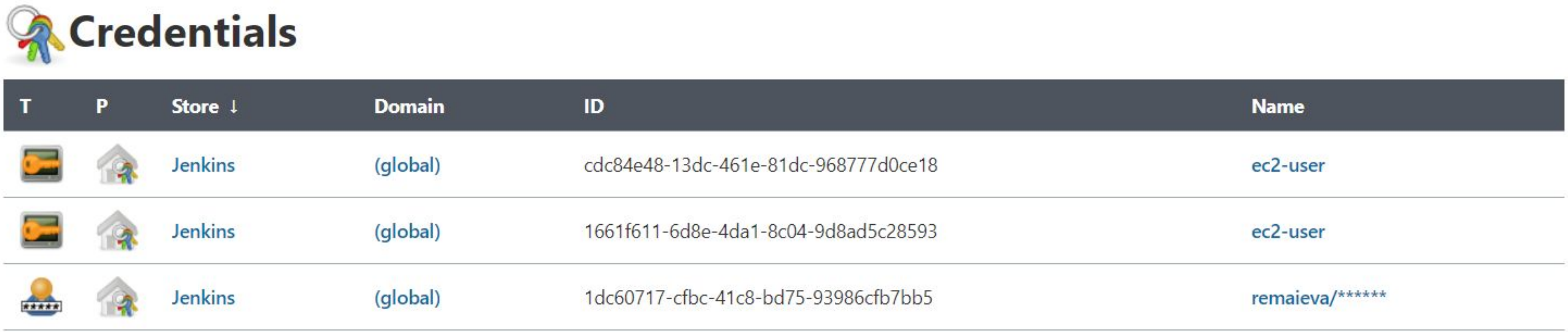
remaieva Update index.jsp 343669b 3 days ago

..

src/main/webapp	Update index.jsp	3 days ago
pom.xml	Jenkins Docker pipeline build Java Hello, World	9 days ago

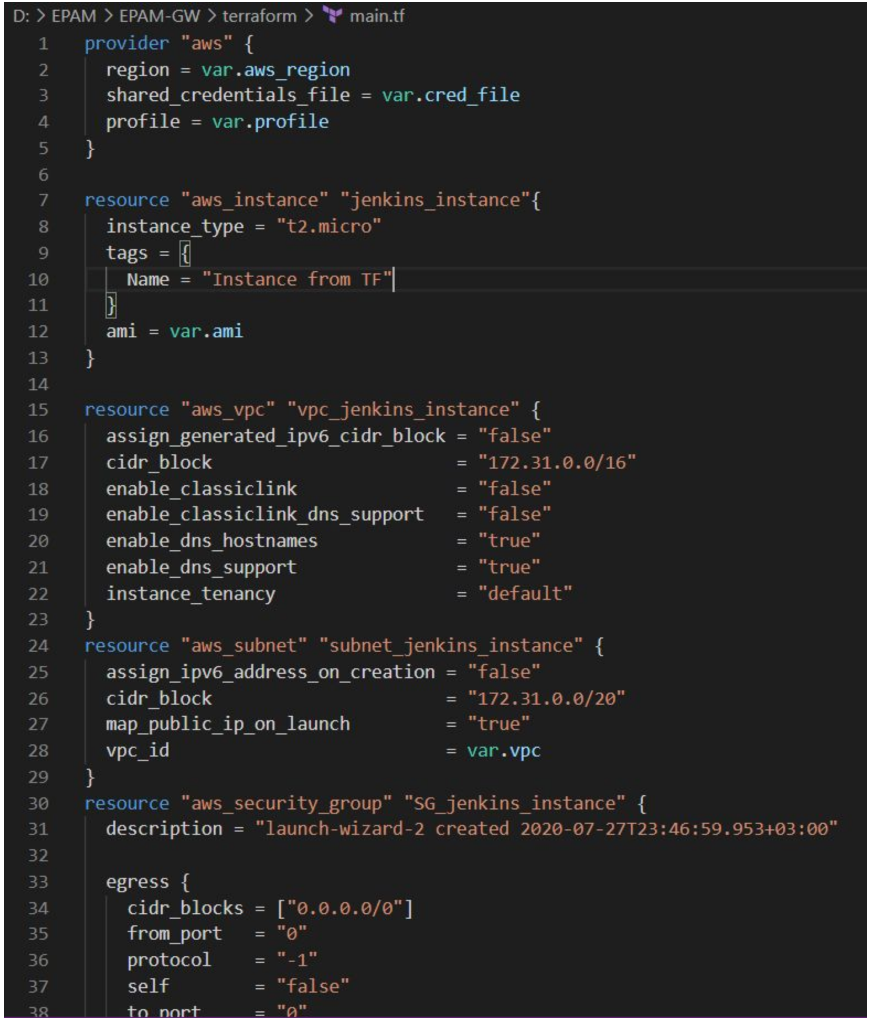


	build	Git clone	Maven build
Average stage times: (Average full run time: ~9s)	1s	530ms	8s
#33 Nov 15 23:45 1 commit	1s	501ms	7s

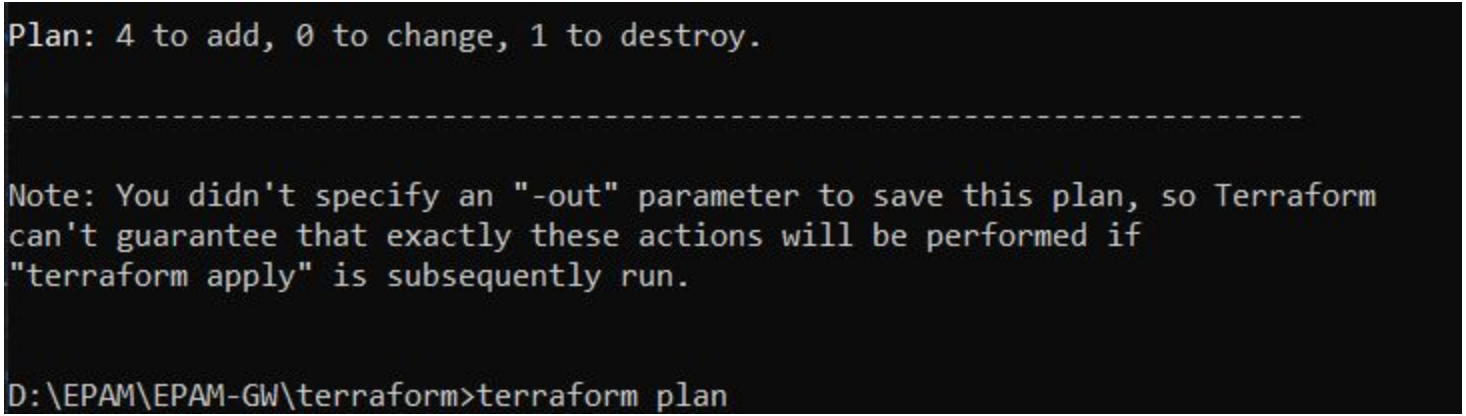


Credentials

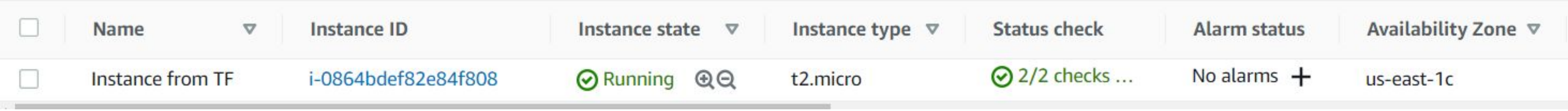
T	P	Store ↓	Domain	ID	Name
🔑	🏠	Jenkins	(global)	cdc84e48-13dc-461e-81dc-968777d0ce18	ec2-user
🔑	🏠	Jenkins	(global)	1661f611-6d8e-4da1-8c04-9d8ad5c28593	ec2-user
*****	🏠	Jenkins	(global)	1dc60717-cfbc-41c8-bd75-93986cfb7bb5	remaieva/******



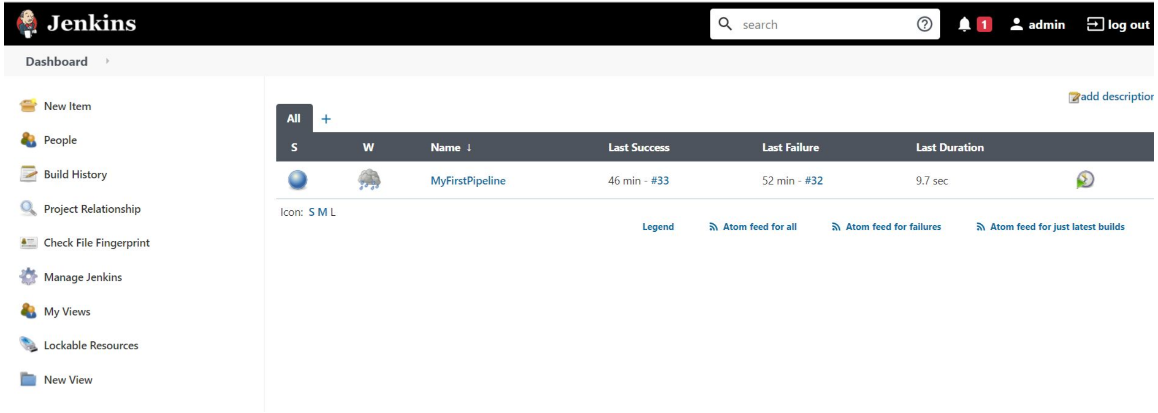
```
provider "aws" {
  region = var.aws_region
  shared_credentials_file = var.cred_file
  profile = var.profile
}

resource "aws_instance" "jenkins_instance"{
  instance_type = "t2.micro"
  tags = {
    Name = "Instance from TF"
  }
  ami = var.ami
}

resource "aws_vpc" "vpc_jenkins_instance" {
  assign_generated_ipv6_cidr_block = "false"
  cidr_block                       = "172.31.0.0/16"
  enable_classiclink               = "false"
  enable_classiclink_dns_support   = "false"
  enable_dns_hostnames             = "true"
  enable_dns_support               = "true"
  instance_tenancy                 = "default"
}
resource "aws_subnet" "subnet_jenkins_instance" {
  assign_ipv6_address_on_creation = "false"
  cidr_block                      = "172.31.0.0/20"
  map_public_ip_on_launch         = "true"
  vpc_id                          = var.vpc
}
resource "aws_security_group" "SG_jenkins_instance" {
  description = "launch-wizard-2 created 2020-07-27T23:46:59.953+03:00"

  egress {
    cidr_blocks = ["0.0.0.0/0"]
    from_port   = "0"
    protocol    = "-1"
    self        = "false"
    to_port     = "0"
```



```
Plan: 4 to add, 0 to change, 1 to destroy.

------------------------------------------------------------------------

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.


D:\EPAM\EPAM-GW\terraform>terraform plan
```



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	Instance from TF	i-0864bdef82e84f808	✓ Running	t2.micro	✓ 2/2 checks ...	No alarms +	us-east-1c



S	W	Name ↓	Last Success	Last Failure	Last Duration
●		MyFirstPipeline	46 min - #33	52 min - #32	9.7 sec

Icon: S M L

Conclusion

I created on-cloud pipeline with high scalability.
Also I will be good to add webhooks for Jenkins to trigger build and add push stage in Jenkinsfile. I get difficulties with it.