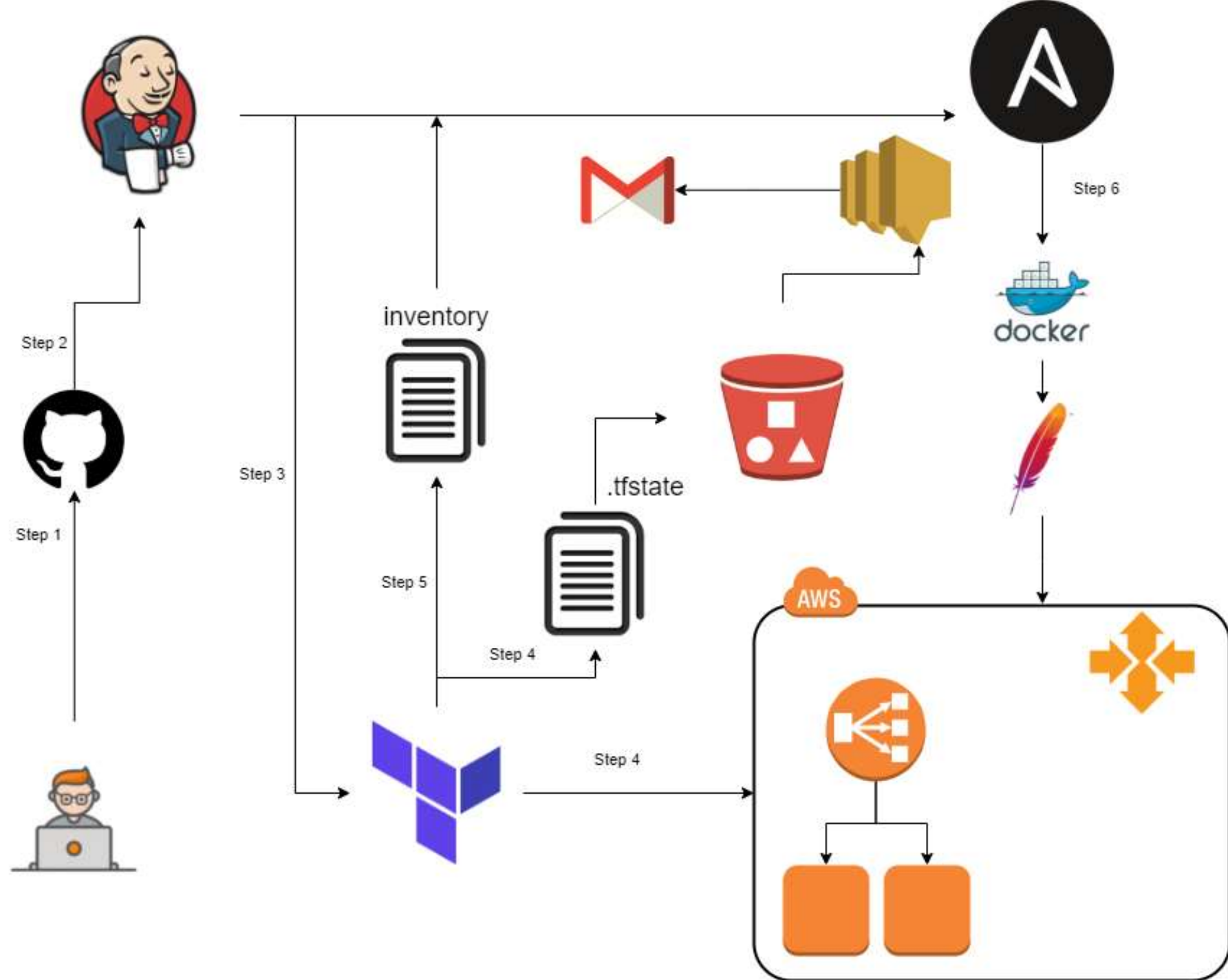




EPAM DevOps Summer online Graduation Work

Highly available web server on elastic load balancer

Author: Oleksii Hurtov



Motivation

My solution is essential because it provides high availability, fast re-creation and complete automation of product deployment.

Relevance

The solution to this problem is vital because:

- ▶ It saves time and reduces the risk of human error. The entire deployment process is automated when developer pushing changes to GitHub.
- ▶ Ansible and Terraform provides easy and fast creation, management and restore entire infrastructure
- ▶ Docker enables to separate application from infrastructure, which improves security
- ▶ AWS provides high availability, security, and scalability

Purpose of work

- ▶ Automate the creation of infrastructure for the website
- ▶ Automate the website deployment process
- ▶ Improve infrastructure security with docker
- ▶ Build a reliable and highly available solution on a cloud provider

Tasks

- ▶ Create infrastructure on AWS using Terraform
- ▶ Configure server environments using Ansible
- ▶ Install Apache web server in Docker container
- ▶ Automate the deployment process with Jenkins pipelines

Designing

Technology stack consists of Terraform, Ansible, Docker, Apache, AWS, Jenkins, GitHub.

- ▶ Terraform is the configuration orchestration tool that works with any cloud, and allows safe and convenient design, management and improvement for infrastructure as code.
- ▶ Ansible automates and simplifies repetitive, complex, and tedious operations.
- ▶ Docker enables to easily pack, ship, and run any application as a lightweight, portable, self-sufficient container, which can run virtually anywhere.
- ▶ Apache web servers are easy to customize environments, they are fast, reliable, and highly secure.
- ▶ AWS is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally.
- ▶ Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project
- ▶ GitHub is a version control system, which allows for seamless collaboration without compromising the integrity of the original project.

Realization

The first challenge was to find out how to run Ansible after Terraform has created servers. I used function “templatefile” for creating inventory file for ansible using “local_file” resource.

Then I configured Ansible to installing Docker on servers, deploying files and running Docker containers. It was not difficult because Ansible has clear documentation.

Finally, I configured Jenkins master and created 3 jobs:

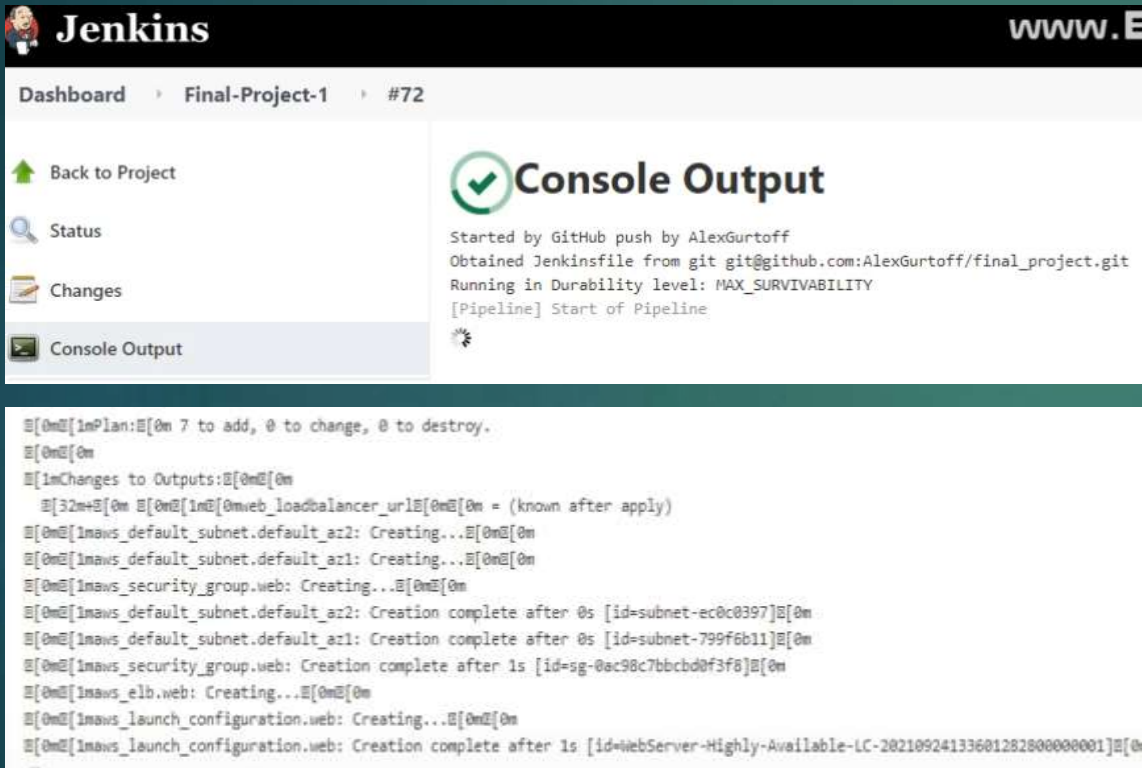
1. Runs on changes in all folders except “WebSite”. This job creates infrastructure, configures servers and deploys the site to them
2. Runs on changes only in the “WebSite” folder. This job just deploys the site to target servers and recreate containers. I made this job to avoid unnecessary actions, when we have changes that do not affect the infrastructure or server configuration.
3. This job we can run remotely by link to destroy entire infrastructure. This can be useful when we forgot to destroy the infrastructure and are already far from the laptop. Just click on the link. Also this can be used in scripts.

So I have achieved all my goals

Demonstration

Here are some screenshots of my project demo.

After pushing changes to the repository:

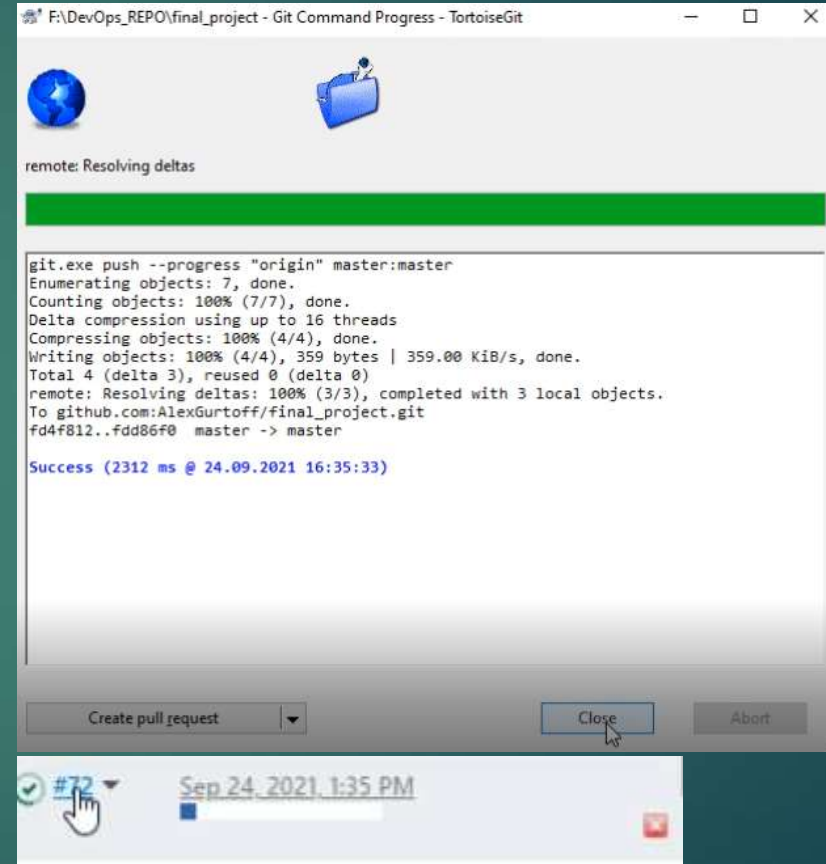


The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, the text 'Jenkins', and the URL 'www.E'. Below the navigation bar, the breadcrumb trail is 'Dashboard > Final-Project-1 > #72'. On the left sidebar, there are links for 'Back to Project', 'Status', 'Changes', and 'Console Output'. The main content area is titled 'Console Output' with a green checkmark icon. It displays the following text:

```
Started by GitHub push by AlexGurtoff
Obtained Jenkinsfile from git git@github.com:AlexGurtoff/final_project.git
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
```

Below the console output, there is a scrollable log of Terraform commands and outputs:

```
[0m][1mPlan:[0m 7 to add, 0 to change, 0 to destroy.
[0m][0m
[1mChanges to Outputs:[0m[0m[0m
  [32m+:[0m [0m[1m[0mweb_loadbalancer_url[0m[0m = (known after apply)
[0m[1maws_default_subnet.default_az2: Creating...[0m[0m
[0m[1maws_default_subnet.default_az1: Creating...[0m[0m
[0m[1maws_security_group.web: Creating...[0m[0m
[0m[1maws_default_subnet.default_az2: Creation complete after 0s [id=subnet-ec0c0397][0m
[0m[1maws_default_subnet.default_az1: Creation complete after 0s [id=subnet-799f6b11][0m
[0m[1maws_security_group.web: Creation complete after 1s [id=sg-0ac98c7bbcbdbf3f6][0m
[0m[1maws_elb.web: Creating...[0m[0m
[0m[1maws_launch_configuration.web: Creating...[0m[0m
[0m[1maws_launch_configuration.web: Creation complete after 1s [id=webServer-Highly-Available-LC-20210924133601282800000001][0m
```



The screenshot shows the 'Git Command Progress - TortoiseGit' window. The title bar indicates the path 'F:\DevOps_REPO\final_project'. The window displays the progress of a git push command. The status bar at the top shows 'remote: Resolving deltas'. The main text area shows the following output:

```
git.exe push --progress "origin" master:master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 359 bytes | 359.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:AlexGurtoff/final_project.git
fd4f812..fdd86f0  master -> master

Success (2312 ms @ 24.09.2021 16:35:33)
```

At the bottom of the window, there are buttons for 'Create pull request', 'Close', and 'Abort'. A mouse cursor is hovering over the 'Close' button. Below the window, there is a small status bar showing a green checkmark, the commit hash '#72', and the date and time 'Sep 24, 2021 1:35 PM'.

```

E[0mweb_loadbalancer_url = "WebServer-944659891.eu-west-3.elb.amazonaws.com"
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] ansiblePlaybook
[Final-Project-1] $ ansible-playbook ansible/playbook.yml -i /var/lib/jenkins/inve
PLAY [all] *****

TASK [install_docker : Wait for system to become reachable] *****
ok: [15.237.36.127]
ok: [13.36.235.35]

TASK [install_docker : Gather facts for first time] *****
ok: [15.237.36.127]
ok: [13.36.235.35]

TASK [install_docker : Install required system packages] *****
changed: [13.36.235.35] => (item=apt-transport-https)
changed: [15.237.36.127] => (item=apt-transport-https)
changed: [13.36.235.35] => (item=ca-certificates)
changed: [15.237.36.127] => (item=ca-certificates)
changed: [13.36.235.35] => (item=curl)
changed: [15.237.36.127] => (item=curl)
ok: [13.36.235.35] => (item=software-properties-common)
ok: [15.237.36.127] => (item=software-properties-common)
changed: [15.237.36.127] => (item=python3-pip)
changed: [13.36.235.35] => (item=python3-pip)
changed: [15.237.36.127] => (item=virtualenv)
ok: [15.237.36.127] => (item=python3-setuptools)
changed: [13.36.235.35] => (item=virtualenv)
ok: [13.36.235.35] => (item=python3-setuptools)

TASK [install_docker : Add Docker GPG apt Key] *****
changed: [13.36.235.35]
changed: [15.237.36.127]

TASK [install_docker : Add Docker Repository] *****

```

```

TASK [install_docker : Add Docker Repository] *****
changed: [13.36.235.35]
changed: [15.237.36.127]

TASK [install_docker : Update apt and install docker-ce] *****
changed: [15.237.36.127]
changed: [13.36.235.35]

TASK [install_docker : Install Docker Module for Python] *****
changed: [13.36.235.35]
changed: [15.237.36.127]

TASK [run_docker_container : Copy WebSite to Servers] *****
changed: [13.36.235.35]
changed: [15.237.36.127]

TASK [run_docker_container : Add private IP to index.htm] *****
changed: [13.36.235.35]
changed: [15.237.36.127]

PLAY RECAP *****
13.36.235.35      : ok=10   changed=8   unreachable=0   failed=0   skipped=0
15.237.36.127    : ok=10   changed=8   unreachable=0   failed=0   skipped=0

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Stage View



**My CV**

Oleksii_Hurtov_CV.pdf

Linkedin Profilemy [Linked in](#) profile**My Final Project for EPAM DevOps online Summer 2021**[GitHub Link](#)**Tutorials or qwiklabs which I have completed:**

1. [Store and Retrieve a File with Amazon S3](#)
2. [Launch and configure a WordPress instance with Amazon Lightsail](#)
3. [Launch a Linux Virtual Machine with Amazon Lightsail](#)
4. [AWS Quick Start Guide: Launch a Linux Virtual Machine](#)
5. [Batch upload files to the cloud to Amazon S3 using the AWS CLI](#)
6. [Register a Domain Name with Amazon Route 53](#)
7. [Deploy Docker Containers on Amazon Elastic Container Service \(Amazon ECS\)](#)
8. [Configuring a static website on Amazon S3](#)
9. [Amazon EBS snapshots](#)
10. [Requesting a public certificate](#)
11. [Introduction to AWS Identity and Access Management \(IAM\)](#)
12. [Introduction to Amazon EC2](#)
13. [Introduction to Amazon Simple Storage Service \(S3\)](#)
14. [Introduction to Elastic Load Balancing](#)
15. [Working with Elastic Load Balancing](#)
16. [Introduction to Amazon EC2 Auto Scaling](#)
17. [Introduction to Amazon Virtual Private Cloud \(VPC\)](#)
18. [Introduction to AWS CloudFormation](#)
19. [Introduction to AWS Lambda](#)
20. [Introduction to Amazon Relational Database Service \(RDS\) \(Windows\)](#)

172.31.15.97

**My CV**

Oleksii_Hurtov_CV.pdf

Linkedin Profilemy [Linked in](#) profile**My Final Project for EPAM DevOps online Summer 2021**[GitHub Link](#)**Tutorials or qwiklabs which I have completed:**

1. [Store and Retrieve a File with Amazon S3](#)
2. [Launch and configure a WordPress instance with Amazon Lightsail](#)
3. [Launch a Linux Virtual Machine with Amazon Lightsail](#)
4. [AWS Quick Start Guide: Launch a Linux Virtual Machine](#)
5. [Batch upload files to the cloud to Amazon S3 using the AWS CLI](#)
6. [Register a Domain Name with Amazon Route 53](#)
7. [Deploy Docker Containers on Amazon Elastic Container Service \(Amazon ECS\)](#)
8. [Configuring a static website on Amazon S3](#)
9. [Amazon EBS snapshots](#)
10. [Requesting a public certificate](#)
11. [Introduction to AWS Identity and Access Management \(IAM\)](#)
12. [Introduction to Amazon EC2](#)
13. [Introduction to Amazon Simple Storage Service \(S3\)](#)
14. [Introduction to Elastic Load Balancing](#)
15. [Working with Elastic Load Balancing](#)
16. [Introduction to Amazon EC2 Auto Scaling](#)
17. [Introduction to Amazon Virtual Private Cloud \(VPC\)](#)
18. [Introduction to AWS CloudFormation](#)
19. [Introduction to AWS Lambda](#)
20. [Introduction to Amazon Relational Database Service \(RDS\) \(Windows\)](#)

172.31.23.245

After F5

We have changed here only the WebSite folder, therefore, only the second job was triggered.

```
te #8 www.BANDICAM.COM

Started by GitHub push by AlexGurtoff
Obtained JenkinsWeb from git git@github.com:AlexGurtoff/final_project.git
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Final-Project-New-Website
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential GitHub-SSH-Key
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Final-Project-New-Website/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url git@github.com:AlexGurtoff/final_project.git # timeout=10
Fetching upstream changes from git@github.com:AlexGurtoff/final_project.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
using GIT_SSH to set credentials GitHub-SSH-Key
> git fetch --tags --force --progress -- git@github.com:AlexGurtoff/final_project.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision facfd6534c338cf5c494898a9308a4821a64c677 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f facfd6534c338cf5c494898a9308a4821a64c677 # timeout=10
Commit message: "web trigger"
> git rev-list --no-walk ce57897baac833aeb41f782c8eae4b2301b924b8 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withCredentials
WARNING: Unknown parameter(s) found for class type 'com.cloudbees.jenkins.plugins.awscredentials.AmazonWebServicesCredentials'
Masking supported pattern matches of $AWS_ACCESS_KEY_ID or $AWS_SECRET_ACCESS_KEY
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] ansiblePlaybook
[Final-Project-New-Website] $ ansible-playbook ansible/playbook_deploy.yml -i /var/lib/jenkins/inventory --priv

PLAY [all] *****

TASK [run_docker_container : Copy WebSite to Servers] *****
changed: [15.236.42.59]
changed: [15.237.123.136]

TASK [run_docker_container : Add private IP to index.htm] *****
changed: [15.236.42.59]
changed: [15.237.123.136]

TASK [run_docker_container : Create apache container] *****
changed: [15.236.42.59]
changed: [15.237.123.136]

PLAY RECAP *****
15.236.42.59 : ok=3 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
15.237.123.136 : ok=3 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

```
[Final-Project-New-Website] $ ansible-playbook ansible/playbook_deploy.yml -i /var/lib/jenkins/inventory --private-key=/var/lib/jenkins/inventory/ansible/ansible-vault-key.pem

PLAY [all] *****

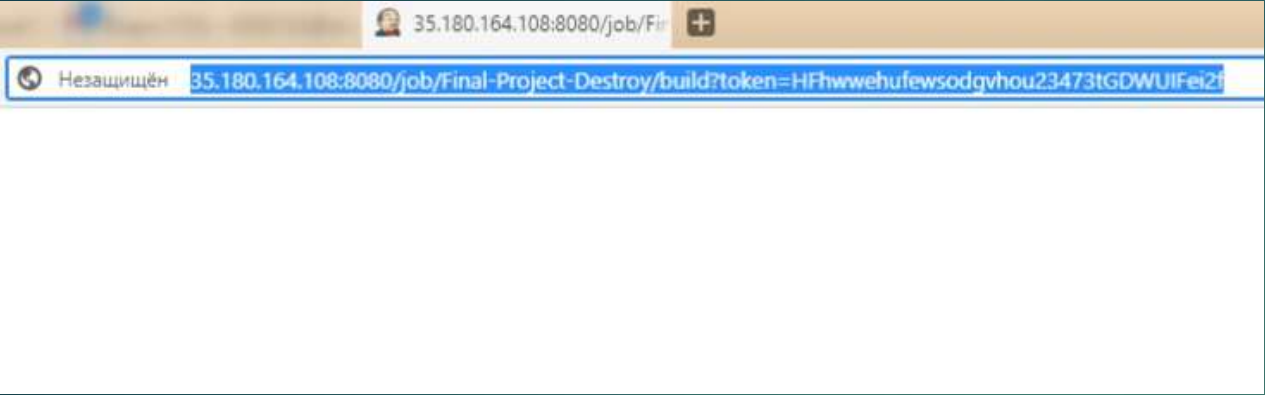
TASK [run_docker_container : Copy WebSite to Servers] *****
changed: [15.236.42.59]
changed: [15.237.123.136]

TASK [run_docker_container : Add private IP to index.htm] *****
changed: [15.236.42.59]
changed: [15.237.123.136]

TASK [run_docker_container : Create apache container] *****
changed: [15.236.42.59]
changed: [15.237.123.136]

PLAY RECAP *****
15.236.42.59 : ok=3 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
15.237.123.136 : ok=3 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

And here was triggered 3rd job, because we have followed the link:



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

Authentication Token

HFhwwehufewsodgvyhou23473tGDWUIFei2f

Use the following URL to trigger build remotely: JENKINS_URL/job/Final-Project-Destroy/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME

Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

```
0m0 0 to add, 0 to change, 7 to destroy.
0m0 0m
1mChanges to Outputs:0m0 0m
```

Instances (5) Info

Filter instances

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	WebServer in ASG	i-03ed51f3f744d27d3	Terminated	t2.micro
<input type="checkbox"/>	WebServer in ASG	i-03e4375274ea43cb2	Terminated	t2.micro
<input type="checkbox"/>	Jenkins	i-0359edf5f398c39af	Running	t3.large
<input type="checkbox"/>	WebServer in ASG	i-048c4246c74285a31	Terminated	t2.micro
<input type="checkbox"/>	WebServer in ASG	i-0be9da8bc299bc0ca	Terminated	t2.micro

	GIT	Destroy infrastructure
Average stage times: (Average full run time: ~1min 53s)	1s	1d 16h
#4 Sep 24 16:46 20 commits	1s	1min 51s

Не удается получить доступ к сайту

Проверьте, нет ли опечаток в имени хоста webserver-944659891.eu-west-3.elb.amazonaws.com.

Conclusion

- ❑ I have created highly available web server on elastic load balancer.
- ❑ One of the improvement ideas is to use the green-blue deployment method to increase the site availability time when infrastructure re-creating.
- ❑ We could also use ready-made solutions from AWS, for example, Elastic Beanstalk, is an easy-to-use service for deploying and scaling web applications and services on servers such as Apache, but in this case, I would not be able to demonstrate my skills in many tools.
- ❑ However, there is one more way. That is ECS which can manage container orchestration service. Thus, we would not have to create instances and install docker on them.