Infrastructure Management & CI/CD

EPAM external course final project 2020/Q1

Author: **Suprun Sergey**
email : serg.suprun@gmail.com

Tools

Terraform
Use Infrastructure as Code to provision and manage any cloud, infrastructure, or service

Amazon Web Services (**AWS**) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.

Ansible
Automate, accelerate, collaborate, integrate.

Jenkins an open source automation server which enables developers around the world to reliably build, test, and deploy their software.
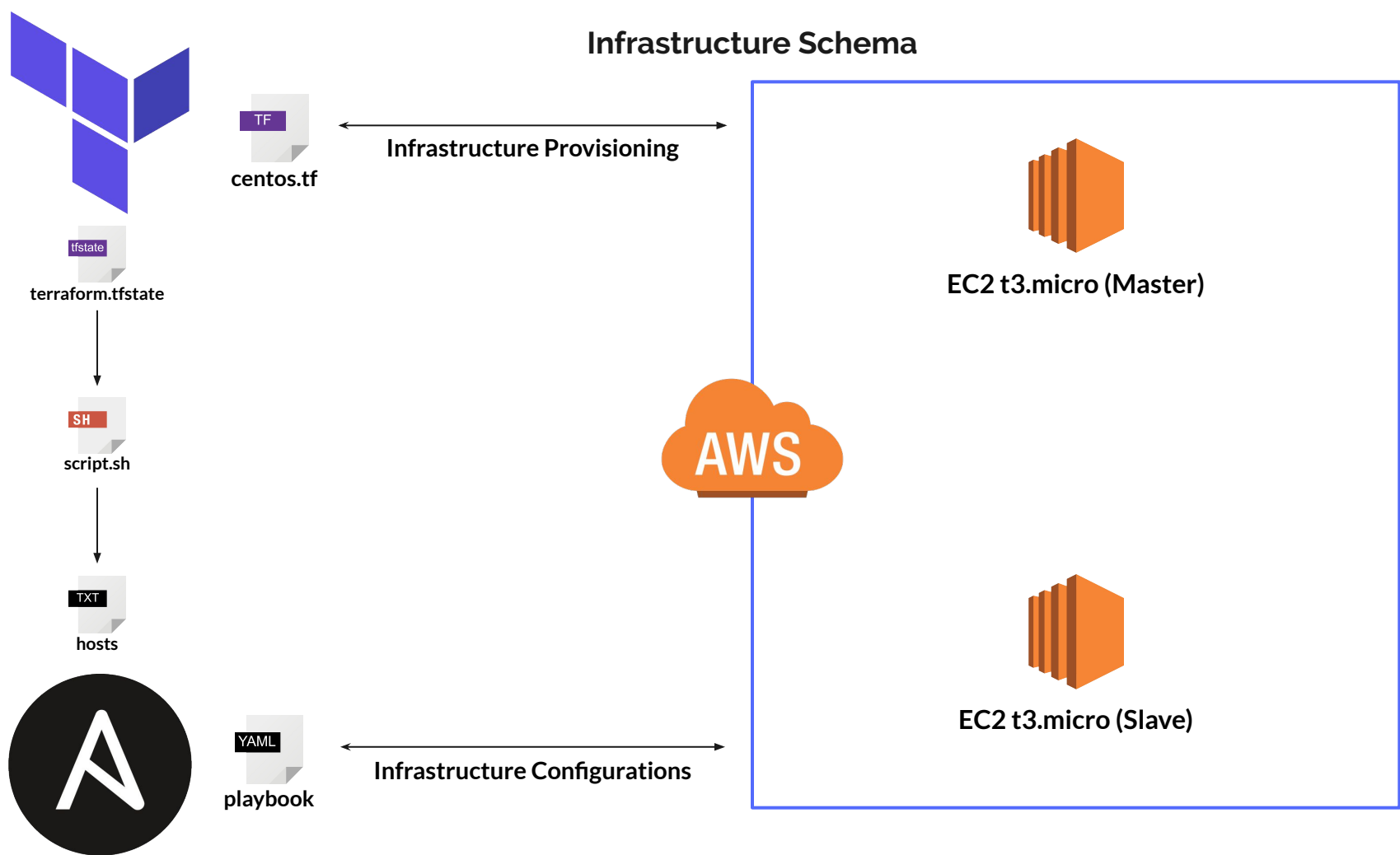
Docker
Securely build and share any application, anywhere

GitHub
It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features.

Infrastructure Schema

centos.tf

Infrastructure Provisioning

terraform.tfstate

script.sh

hosts

playbook

Infrastructure Configurations

AWS

EC2 t3.micro (Master)

EC2 t3.micro (Slave)

3

Terraform Infrastructure Management

```
resource "aws_instance" "centos_master" {
  ami         = "ami-0f2b4fc905b0bd1f1"
  instance_type = "t3.micro"
  key_name = "project"
  tags = {
     Name = "CentOS master"
     Owner = "Sergey Suprun"
     Project = "EPAM external course project"
  }
  vpc_security_group_ids = ["${aws_security_group.web_server_ssh.id}", "${aws_security_group.web_server_jenkins.id}"]
}

resource "aws_instance" "centos_slave" {
  ami         = "ami-0f2b4fc905b0bd1f1"
  instance_type = "t3.micro"
  key_name = "project"
  tags = {
     Name = "CentOS Slave"
     Owner = "Sergey Suprun"
     Project = "EPAM external course project"
  }
  vpc_security_group_ids = ["${aws_security_group.web_server_ssh.id}", "${aws_security_group.web_server.id}"]
}
```

BASH script to generate the file hosts for Ansible

terraform.tfstate → **script.sh** → **hosts**

```
"resources": [
  {
   "mode": "managed",
   "type": "aws_instance",
   "name": "centos_master",
   "provider": "provider.aws",
   "instances": [
    {
     "schema_version": 1,
     "attributes": {
      "ami": "ami-0f2b4fc905b0bd1f1",
      "arn": "arn:aws:ec2:us-east-2:573092740426:instance/i-0659c8f51b22a4d5b",
      "associate_public_ip_address": true,
      "availability_zone": "us-east-2c",
      "cpu_core_count": 1,
      "cpu_threads_per_core": 2,
      ...
      ...
      "primary_network_interface_id": "eni-098b065528a748193",
      "private_dns": "ip-172-31-42-169.us-east-2.compute.internal",
      "private_ip": "172.31.42.169",
      "public_dns": "ec2-3-19-245-104.us-east-2.compute.amazonaws.com",
      "public_ip": "3.19.245.104",
      ...
      ...
```

```
[aws_master]
centos_master          ansible_host=3.19.245.104

[aws_slave]
centos_slave           ansible_host=3.17.63.125

[aws:children]
aws_master
aws_slave

[aws:vars]
ansible_user=centos
ansible_ssh_private_key_file=/home/spc/project/.ssh/project.pem
```
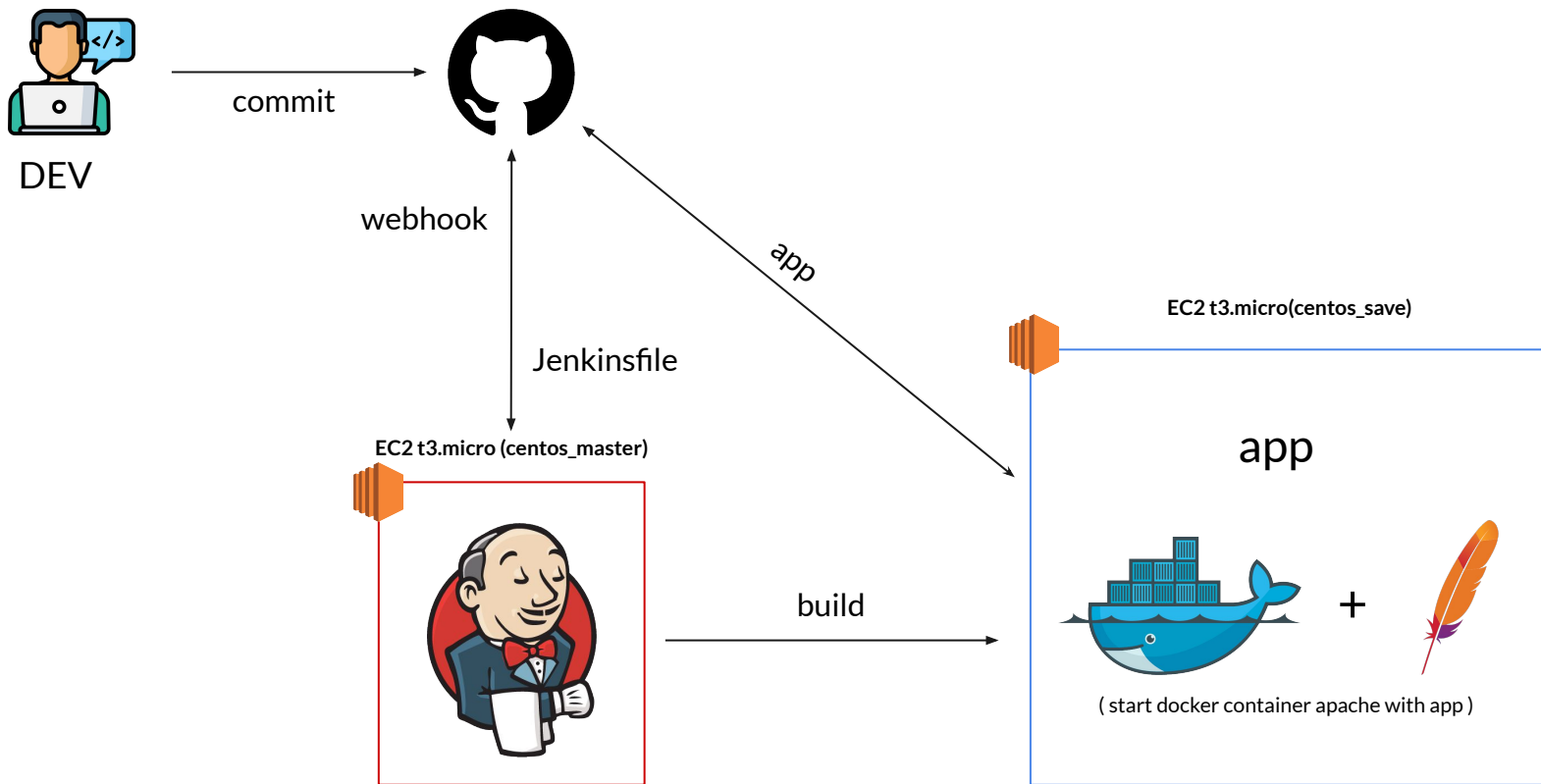
5

Ansible Infrastructure Configuring

Ansible directory structure

```
ansible
├── hosts
├── playbook.yaml ───────────────────────►
└── roles
    ├── docker
    │   └── tasks
    │       └── main.yml
    ├── git
    │   └── tasks
    │       └── main.yml
    ├── java
    │   └── tasks
    │       └── main.yml
    ├── jenkins
    │   └── tasks
    │       └── main.yml
    └── reboot
        └── tasks
            └── main.yml
```

```yaml
---
- hosts: aws_slave
  gather_facts: false
  become: true
  roles:
    - git
    - docker
    - java
    - reboot

- hosts: aws_master
  gather_facts: false
  become: true
  roles:
    - git
    - jenkins
```
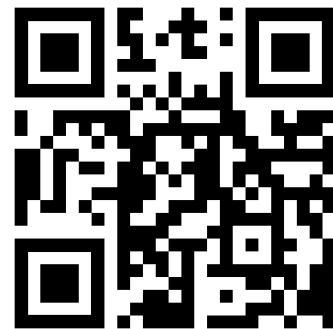
Jenkins build schema

DEV

commit

webhook

app

Jenkinsfile

EC2 t3.micro(centos_save)

EC2 t3.micro (centos_master)

app

build

+

(start docker container apache with app)

Thank you

Jenkins

GitHub/project

GitHub/app

web-app