

DEVOPS ONLINE KYIV 2022
FINAL PROJECT : CICD Pipeline

By Gennadiy Molchanov

About Myself



Personal: 51 years old, born in Kyiv

Start in programming:

1987 - 1988 at school: Data processing, Fortran

1988 - 1994 at university: Basic

1994 Graduated from Kyiv National University, major in physical chemistry

1997 Get second diploma in commercial and labor law (MAUP)

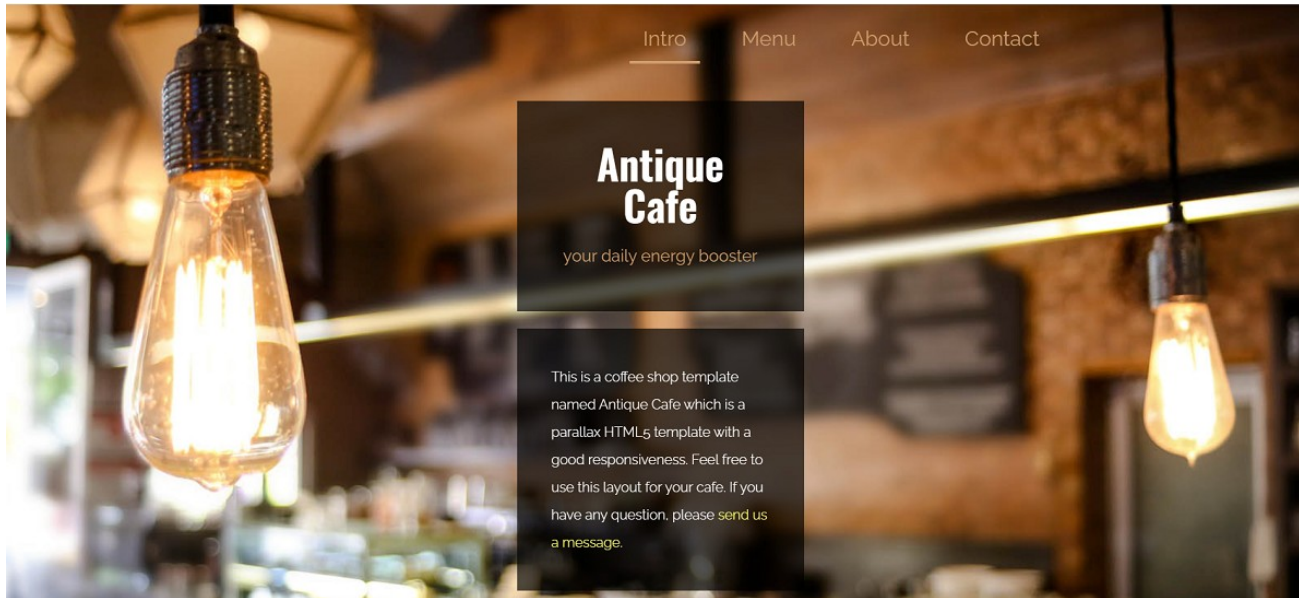
2001 - 2020 Legal practice

2020 - 2021 CS50 and CS50Web with Python and JavaScript (EDX)

Final Project

THEME: continuous integration and continuous delivery of a static website

WEBSITE: template website: Antique Cafe



PROJECT AIM: build CI/CD pipeline with combination of different technologies

Tools



Virtual space, where
resources are created



Code repository



Orchestration of process



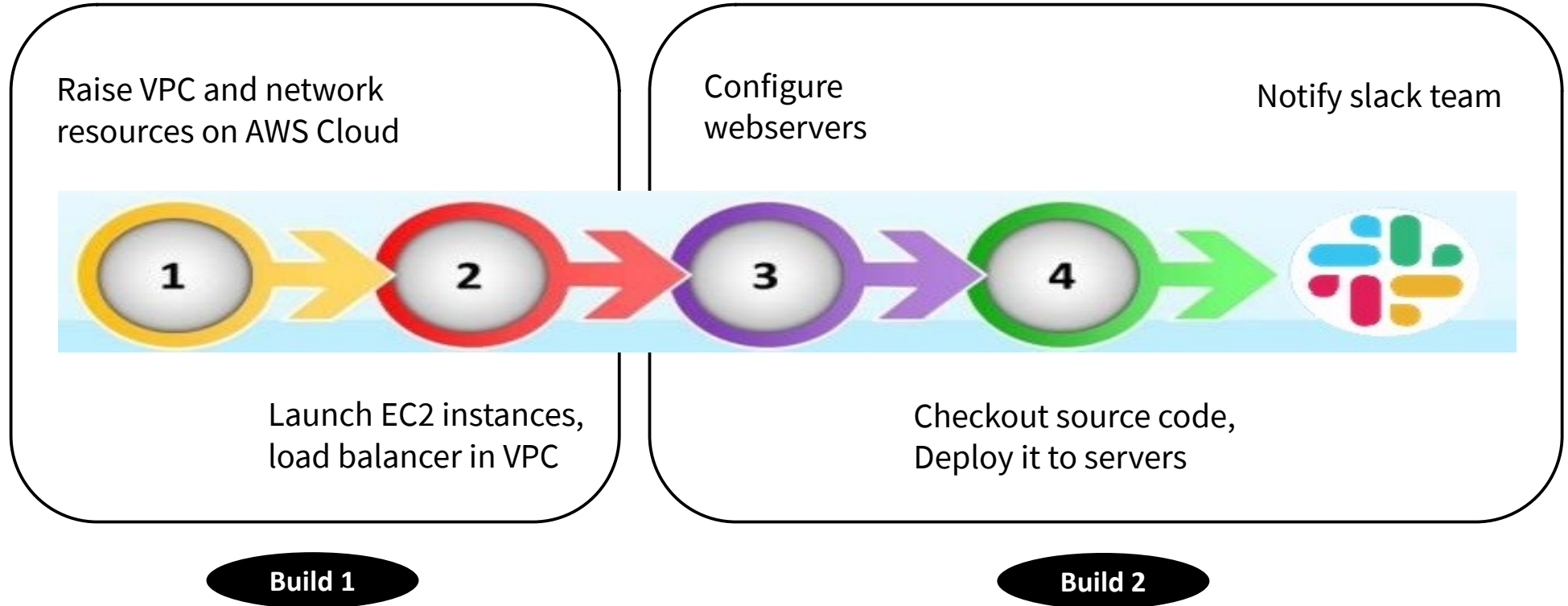
Raise environment with flexible
infrastructure units



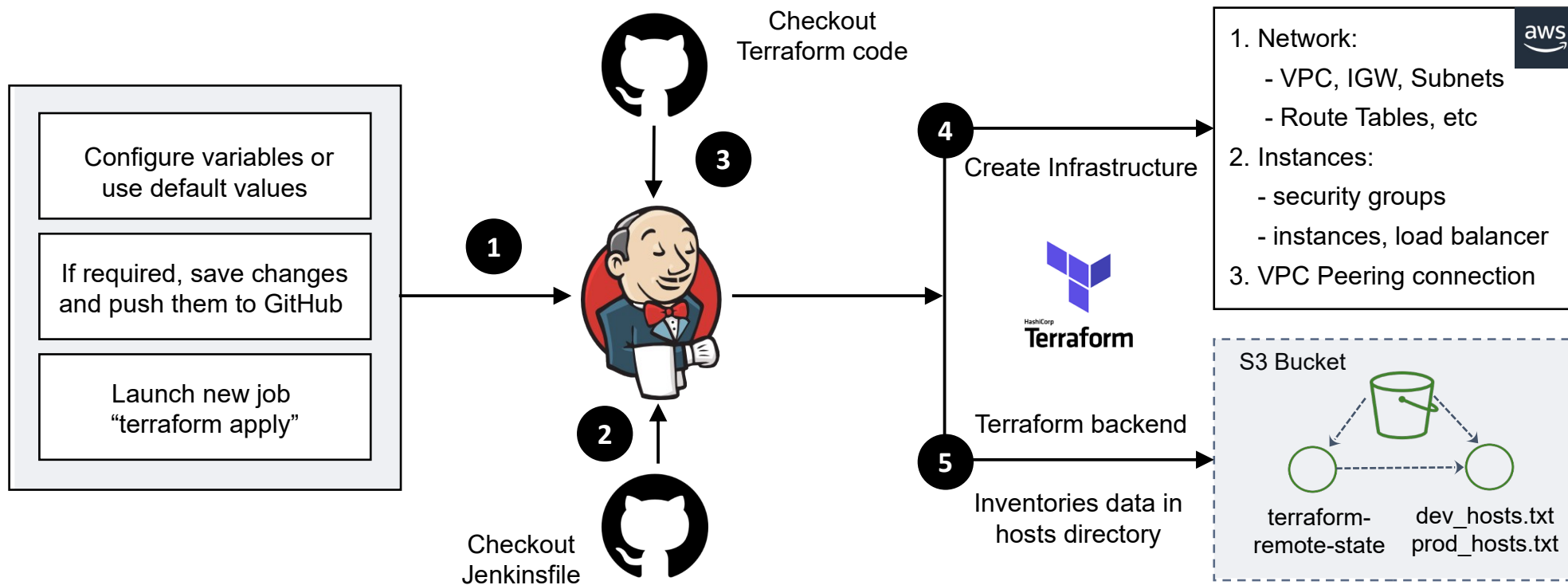
ANSIBLE

Configure EC2 instances
Deliver source code to webserver

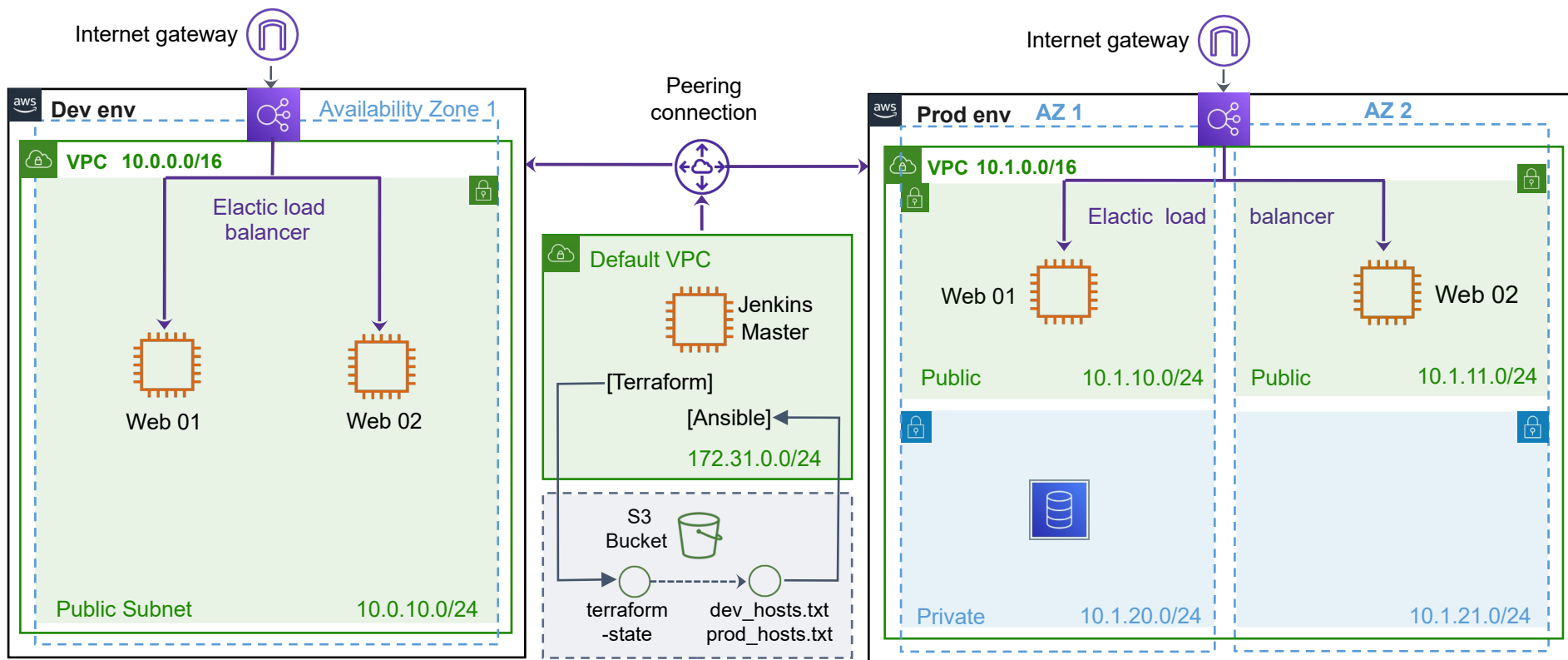
Jenkins Pipeline



BUILD 1: Raise Infrastructure | manually triggered



BUILD 1: Infrastructure



BUILD 1: Terraform code — modules

```
main.tf > module "instances" > type
13
14 # ===== dev-environment =====
15
16 You, 7 days ago | 1 author (You)
17 module "network" {
18     source      = "./aws_modules/network"
19     env         = "dev"
20     public_subnet_cidrs = ["10.0.10.0/24"]
21     private_subnet_cidrs = []
22 }
23
24 You, 43 seconds ago | 1 author (You)
25 module "instances" {
26     source      = "./aws_modules/instances"
27     vpcid       = module.network.vpc_id
28     public_ids  = module.network.public_subnet_ids
29     env         = "dev"
30     user        = "ubuntu"
31     type        = "t2.micro"
32 }

main.tf > ...
31
32 # ===== prod-environment =====
33 You, 2 hours ago | 1 author (You)
34 module "network_prod" {
35     source      = "./aws_modules/network"
36     vpc_cidr    = "10.1.0.0/16"
37     env         = "prod"
38     public_subnet_cidrs = ["10.1.10.0/24", "10.1.11.0/24"]
39     private_subnet_cidrs = ["10.1.20.0/24", "10.1.21.0/24"]
40 }
41
42 You, 43 seconds ago | 1 author (You)
43 module "instances_prod" {
44     source      = "./aws_modules/instances"
45     vpcid       = module.network_prod.vpc_id
46     public_ids  = module.network_prod.public_subnet_ids
47     env         = "prod"
48     user        = "ec2-user"
49     type        = "t2.small"
50 }
```


BUILD 1: Terraform code — backend and inventory data

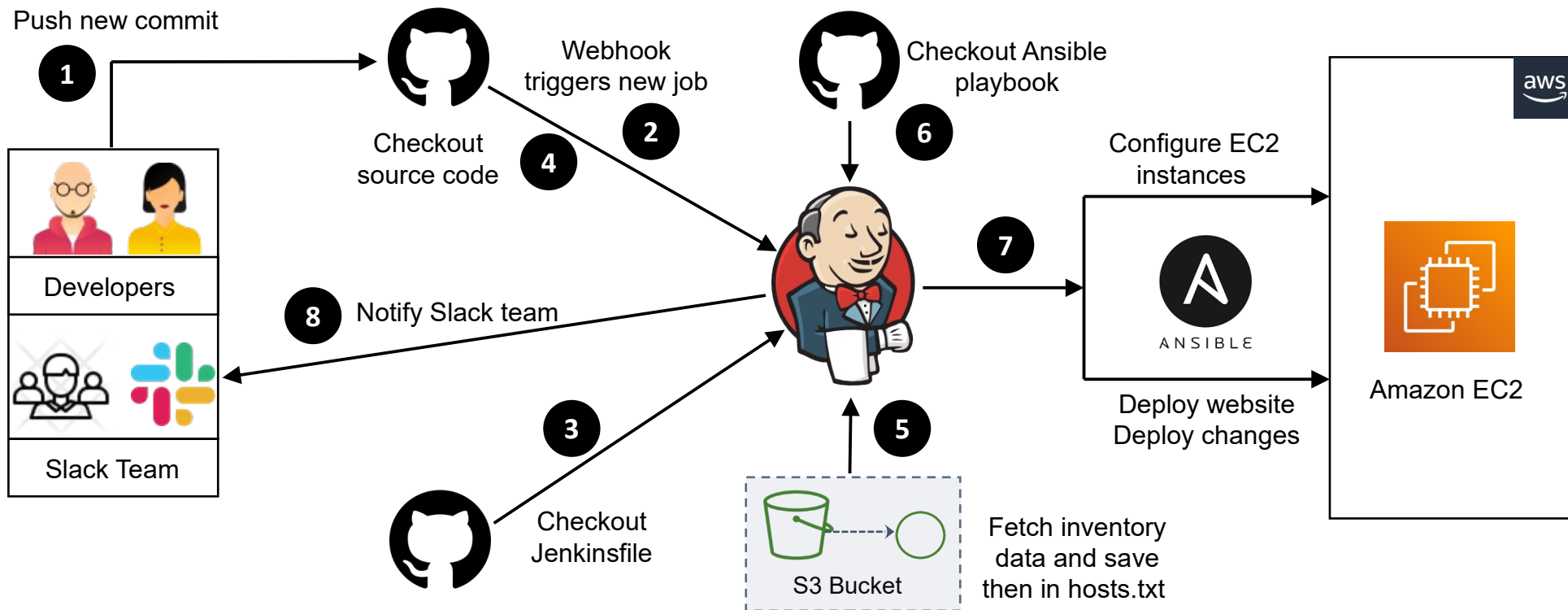
```
ec2_elb_hosts.tf M X
aws_modules > instances > ec2_elb_hosts.tf > ...
147 #####
148 # Collect inventory data and save in s3 #
149 #####
150
You, 5 hours ago | 1 author (You)
151 locals {
152     group_ips = join("\n", [
153         for a in aws_instance.webservers[*].private_ip : "${a} ansible_user=${var.user}"
154     ])
155     group_data = "[${var.env}]\n${local.group_ips}\n"
156 }
157
You, 6 hours ago | 1 author (You)
158 resource "aws_s3_bucket_object" "hosts" {
159     bucket = "terraform-state-cicd"
160     key    = "hosts/${var.env}_hosts.txt"
161     acl    = "private"
162     content = local.group_data
163     depends_on = [aws_instance.webservers]
164 }

backend.tf X
backend.tf > ...
You, 4 weeks ago | 1 author (You)
1 terraform {
2     You, 4 weeks ago | 1 author (You)
3     backend "s3" {
4         bucket = "terraform-state-cicd"
5         key    = "terraform/backend"
6         region = "eu-central-1"
7     }
8 }
```

Inventory data

Terraform remote state

BUILD 2: Configuration and Provisioning | triggered by webhook



BUILD 2: Webhook trigger

Webhooks / Manage webhook

Settings

Recent Deliveries

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, etc). More information can be found in [our developer documentation](#).

Payload URL *

`http://3.73.63.113:8080/github-webhook/`

Jenkins server public IP address;
mutable after restart unless elastic
IP was attached to the server

Content type

`application/json`

Which events would you like to trigger this webhook?

☒ Just the push event.

BUILD 2: Inventory

Amazon S3 > Buckets > terraform-state-cicd > hosts/

Objects (2)

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	dev_hosts.txt	txt
<input type="checkbox"/>	prod_hosts.txt	txt

1. Use AWS CLI to fetch from s3 bucket the inventory files with information about webserver created in dev and/or prod environment.

2. Save collected inventory data in a single hosts.txt file

```
stage('Copy Inventory File'){
  steps{
    sh 'aws s3 ls s3://terraform-state-cicd/hosts/'
    sh 'aws s3 cp s3://terraform-state-cicd/hosts/ ./ansible/ --recursive'
    sh 'cat ./ansible/*.txt > ./ansible/hosts.txt'
    sh 'cat ./ansible/hosts.txt'
  }
}
```

Console Output

```
+ cat ./ansible/hosts.txt
```

```
[dev]
10.0.10.65 ansible_user=ubuntu
10.0.10.82 ansible_user=ubuntu
[prod]
10.1.10.209 ansible_user=ec2-user
10.1.11.109 ansible_user=ec2-user
```

BUILD 2: Ansible playbook: configuration

Flexible configure of apache web server depending on Linux distributive:

- blocks
- conditions
- handlers

```
10 tasks:
11   - name: Check and Print Linux Version
12     debug: var=ansible_os_family
13
14   - block: # ===== Block for Amazon Linux =====
15     - name: Install Apache Web Server and Rsync Utility for Amazon Linux
16       yum:
17         name:
18           - httpd
19           - rsync
20         state: latest
21
22     - name: Synchronize Website code
23
24     - name: Start Web Server and Make It Enable on Boot for Amazon Linux
25       service: name=httpd state=started enabled=yes
26
27   when: ansible_os_family == "RedHat"
28
29   - block: # ===== Block for Ubuntu =====
30     - name: Install Apache Web Server for Ubuntu
31       apt: name=apache2 state=latest
32
33     - name: Synchronize Website code
34
35     - name: Start Web Server and Make It Enable on Boot for Ubuntu
36       service: name=apache2 state=started enabled=yes
37
38   when: ansible_os_family == "Debian"
39
40 handlers: # ===== Handlers =====
```

BUILD 2: Ansible playbook: deploy

Use synchronize module:

- faster than copy module
- removes older and unused files in the destination folder

```
vars:
```

```
source_dir: ../my_website/
```

```
destination_dir: /var/www/html/
```

```
- name: Synchronize Website code
```

```
synchronize:
```

```
src: "{{ source_dir }}"
```

```
dest: "{{ destination_dir }}"
```

```
delete: yes
```

```
recursive: yes
```

```
notify: Restart Apache on Amazon Linux
```

BUILD 2: Slack notification

Pipeline snippets:

```
def COLOR_MAP = [  
    'SUCCESS': 'good',  
    'FAILURE': 'danger',  
]
```

```
post {  
    always {  
        echo 'Slack Notifications.'  
        slackSend channel: '#someproject',  
                  color: COLOR_MAP[currentBuild.currentResult],  
                  message: "**${currentBuild.currentResult}:* Job '${env.JOB_NAME}' build #${env.BUILD_NUMBER} \n  
                             More info at: ${env.BUILD_URL}console"  
    }  
}
```

dev_team ▾

- Channels
 - # general
 - # random
 - # someproject
 - + Add channels
- Direct messages
 - gamolchanov you

someproject ▾



Today ▾

gamolchanov 4:33 PM
joined #someproject.

gamolchanov 4:37 PM
added an integration to this channel: jenkins

Message from Jenkins

SUCCESS: Job '2_configure_and_deploy' build #20
More info at: http://3.73.63.113:8080/job/2_configure_and_deploy/20/console

Now

DOCKER

Earlier experience with docker and docker - compose in my Python/Django project : FakeCSV

- Project on GitHub (with full instruction how to run dockerized application in readme file):
<https://github.com/CodePathfinder/fake-csv/blob/master/README.md>
- Running application on Heroku:
<https://fake--csv.herokuapp.com/>
- Docker image in DockerHub repository:
<https://hub.docker.com/repository/docker/gamdckr/fakecsv>
- Docker-compose file reference:
https://github.com/CodePathfinder/fake-csv/blob/master/readme_docs/docker-compose.yml

CONCLUSIONS

Some ideas for further improvement of the project:

- 1) create dynamic inventory file for Ansible and apply autoscaling group
- 2) use dynamic application with DB connection, DB replication and backup
- 3) refactor terraform code to get more specific and reusable modules
- 4) refactor Ansible code to make it more reusable (roles)
- 5) add custom dns name and ssl certificate
- 6) create bastion host to reach instances placed in private subnets

THANK YOU!