

Indian Institute of Information Technology Vadodara



DevOps Capstone Project

Aman Gangwar
202051020
Semester 6

DevOps

Capstone Project

Name - Aman Gangwar

Student ID - 202051020

Instance Configuration

Total Instances - 3

	jenkins-capstone	test-capstone	prod-capstone
Type	t2.medium	t2.micro	t2.micro
OS	Ubuntu 20.04	Ubuntu 20.04	Ubuntu 20.04
CPUs	2	1	1
RAM	4GB	1GB	1GB
Security Group	jenkins-capstone-sg	test-capstone-sg	prod-capstone-sg

Security Groups Configuration

Ingress (Inbound Rules)

Service\SG		jenkins-capstone-sg	test-capstone-sg	prod-capstone-sg
Jenkins	Port	8080	-	-
	Access	Anywhere	-	-
SSH	Port	22	22	22
	Access	My IP	My IP	My IP
HTTP	Port	-	80	80
	Access	-	Anywhere	Anywhere
Everything	Port	-	All	All
	Access	-	jenkins-capstone-sg	jenkins-capstone-sg

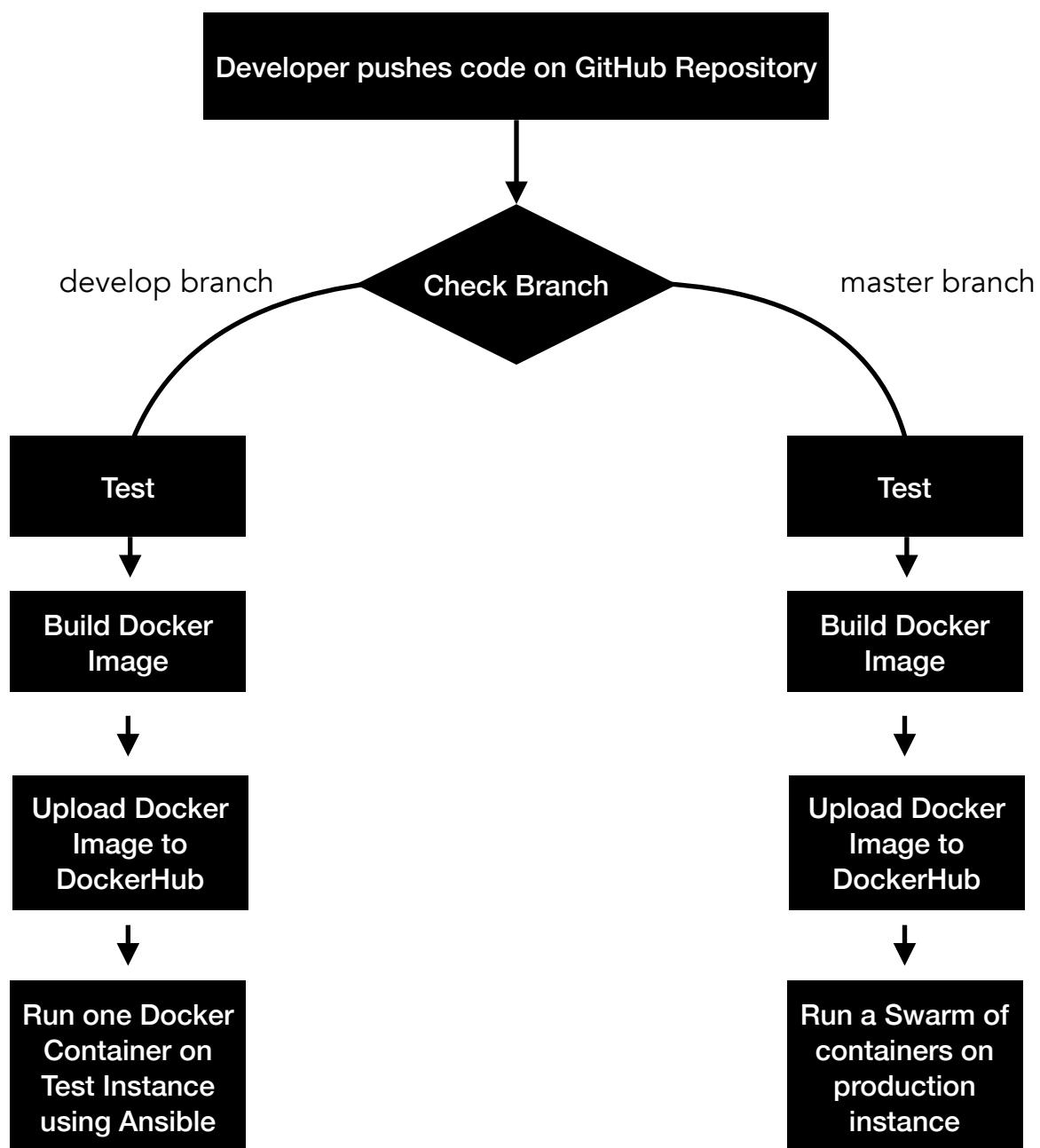
Egress (Outbound Rules)

Service\SG		jenkins-capstone-sg	test-capstone-sg	prod-capstone-sg
Everything	Port	All	All	All
	Access	Anywhere	Anywhere	Anywhere

Tools Used

1. Terraform
2. Git
3. Ansible
4. Jenkins
5. Docker
6. DockerHub
7. Docker Swarm (in place of Kubernetes)

Project Flow



IaC (Infrastructure as Code)

Terraform Script -

```
main.tf > 1 resource "aws_key_pair" "capstone-key" > 2 public_key
1 provider "aws" {
2   region = "us-east-1"
3   access_key = "<access-key>"
4   secret_key = "<access-key>"}
5 }
6
7 # Defining external data source to get the IP address of the machine running Terraform
8 data "external" "myipaddr" {
9   program = ["bash", "-c", "curl -s 'https://ipinfo.io/json'"]
10 }
11
12 resource "aws_key_pair" "capstone-key" {
13   key_name   = "capstone-key"
14   public_key = "<public-key>"}
15 }
16
17
18 # Creating security group for Jenkins Server
19 # SG allows SSH from my IP
20 # SG allows port 8080 from anywhere
21 resource "aws_security_group" "jenkins-capstone-sg" {
22   name = "jenkins-capstone-sg"
23   description = "Security Group for Jenkins Server for Capstone Project"
24
25   ingress {
26     description = "Allow SSH from my IP"
27     from_port = 22
28     to_port = 22
29     protocol = "tcp"
30     cidr_blocks = [ "${data.external.myipaddr.result.ip}/32" ]
31   }
32
33   ingress {
34     description = "Allow port 8080 from anywhere"
35     from_port = 8080
36     to_port = 8080
37     protocol = "tcp"
38     cidr_blocks = [ "0.0.0.0/0" ]
39   }
40
41   egress {
42     from_port = 0
43     to_port = 0
44     protocol = "-1"
45     cidr_blocks = [ "0.0.0.0/0" ]
46   }
47
48   tags = {
49     Name = "jenkins-capstone-sg"
50     Project = "Capstone"
51   }
52 }
53
54 # Creating security group for Test Server
55 # SG allows SSH from my IP
56 # SG allows port 80 from anywhere
57 # SG allows all traffic from Jenkins Server
58 resource "aws_security_group" "test-capstone-sg" [
59   name = "test-capstone-sg"
60   description = "Security Group for Test Server for Capstone Project"
61
62   ingress {
63     description = "Allow SSH from my IP"
64     from_port = 22
65     to_port = 22
66     protocol = "tcp"
67     cidr_blocks = [ "${data.external.myipaddr.result.ip}/32" ]
68   }
69
70   ingress {
71     description = "Allow port 80 from anywhere"
72     from_port = 80
73     to_port = 80
74     protocol = "tcp"
75     cidr_blocks = [ "0.0.0.0/0" ]
76   }
77
78   ingress {
79     description = "Allow all traffic from Jenkins Server"
80     from_port = 0
81     to_port = 0
82     protocol = "-1"
83     security_groups = [ aws_security_group.jenkins-capstone-sg.id ]
84   }
85
86   egress {
87     from_port = 0
88     to_port = 0
89     protocol = "-1"
90     cidr_blocks = [ "0.0.0.0/0" ]
91   }
92 }
```

△ Kubernetes

Ln 14, Col 29 Spaces: 4 UTF-8 LF {} Terraform

```
main.tf > 1 resource "aws_security_group" "test-capstone-sg"
2
3
4 tags = {
5   Name = "jenkins-capstone-sg"
6   Project = "Capstone"
7 }
8
9
10 # Creating security group for Test Server
11 # SG allows SSH from my IP
12 # SG allows port 80 from anywhere
13 # SG allows all traffic from Jenkins Server
14 resource "aws_security_group" "test-capstone-sg" [
15   name = "test-capstone-sg"
16   description = "Security Group for Test Server for Capstone Project"
17
18   ingress {
19     description = "Allow SSH from my IP"
20     from_port = 22
21     to_port = 22
22     protocol = "tcp"
23     cidr_blocks = [ "${data.external.myipaddr.result.ip}/32" ]
24   }
25
26   ingress {
27     description = "Allow port 80 from anywhere"
28     from_port = 80
29     to_port = 80
30     protocol = "tcp"
31     cidr_blocks = [ "0.0.0.0/0" ]
32   }
33
34   ingress {
35     description = "Allow all traffic from Jenkins Server"
36     from_port = 0
37     to_port = 0
38     protocol = "-1"
39     security_groups = [ aws_security_group.jenkins-capstone-sg.id ]
40   }
41
42   egress {
43     from_port = 0
44     to_port = 0
45     protocol = "-1"
46     cidr_blocks = [ "0.0.0.0/0" ]
47   }
48 }
```

△ Kubernetes

Ln 61, Col 1 Spaces: 4 UTF-8 LF {} Terraform

```

main.tf > ` resource "aws_security_group" "test-capstone-sg"
91    }
92
93    tags = {
94      Name = "test-capstone-sg"
95      Project = "Capstone"
96    }
97  }
98
99 # Creating security group for Production Server
100 # SG allows SSH from my IP
101 # SG allows port 80 from anywhere
102 # SG allows all traffic from Jenkins Server
103 resource "aws_security_group" "prod-capstone-sg" {
104   name = "prod-capstone-sg"
105   description = "Security Group for Production Server for Capstone Project"
106
107   ingress {
108     description = "Allow SSH from my IP"
109     from_port = 22
110     to_port = 22
111     protocol = "tcp"
112     cidr_blocks = [ "${data.external.myipaddr.result.ip}/32" ]
113   }
114
115   ingress {
116     description = "Allow port 80 from anywhere"
117     from_port = 80
118     to_port = 80
119     protocol = "tcp"
120     cidr_blocks = ["0.0.0.0/0"]
121   }
122
123   ingress {
124     description = "Allow all traffic from Jenkins Server"
125     from_port = 0
126     to_port = 0
127     protocol = "-1"
128     security_groups = [aws_security_group.jenkins-capstone-sg.id]
129   }
130
131   egress {
132     from_port = 0
133     to_port = 0
134     protocol = "-1"
135     cidr_blocks = ["0.0.0.0/0"]
136   }
△ Kubernetes
main.tf > ` resource "aws_security_group" "test-capstone-sg"
137
138   tags = {
139     Name = "prod-capstone-sg"
140     Project = "Capstone"
141   }
142
143
144 # Creating Jenkins Server
145 # Properties -
146 #   - Instance type: t2.medium
147 #   - AMI: Ubuntu 20.04 (LTS)
148 #   - Security Group: jenkins-capstone-sg
149 #   - Key Pair: capstone-key
150 resource "aws_instance" "jenkins-capstone" {
151   ami = "ami-0aa2b772dc1b5612"
152   instance_type = "t2.medium"
153   key_name = "capstone-key"
154   security_groups = [aws_security_group.jenkins-capstone-sg.name]
155   user_data = <<-EOF
156   #!/bin/bash
157
158   # Install Jenkins
159   sudo apt update
160   sudo apt install git openjdk-11-jdk -y
161   curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
162   /usr/share/keyrings/jenkins-keyring.asc > /dev/null
163   echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
164   https://pkg.jenkins.io/debian-stable binary | sudo tee \
165   /etc/apt/sources.list.d/jenkins.list > /dev/null
166   sudo apt update
167   sudo apt-get update
168   sudo apt-get install jenkins -y
169
170   # Install Docker
171   sudo apt-get update
172   sudo apt-get install -y \
173     ca-certificates \
174     curl \
175     gnupg
176   sudo install -m 0755 -d /etc/apt/keyrings
177   curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
178   sudo chmod a+r /etc/apt/keyrings/docker.gpg
179   echo \
180   "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
181   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
△ Kubernetes

```

Ln 61, Col 1 Spaces: 4 UTF-8 LF {} Terraform ⚙️ 🔍 🗃

```

main.tf > ` resource "aws_security_group" "test-capstone-sg"
181     "$!. /etc/os-release && echo \"$VERSION_CODENAME\" stable" | \
182     sudo tee /etc/apt/sources.list.d/docker.list >/dev/null
183     sudo apt-get update
184     sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
185
186     # Install Ansible
187     sudo apt update
188     sudo apt install software-properties-common -y
189     sudo apt-add-repository --yes --update ppa:ansible/ansible
190     sudo apt-get install ansible -y
191     EOF
192
193     tags = {
194         Name = "jenkins-capstone"
195         Project = "Capstone"
196     }
197 }
198
199 # Creating Test Server
200 # Properties -
201 #   - Instance type: t2.micro
202 #   - AMI: Ubuntu 20.04 (LTS)
203 #   - Security Group: test-capstone-sg
204 #   - Key Pair: capstone-key
205 resource "aws_instance" "test-capstone" {
206     ami = "ami-0aa2b772dc1b5612"
207     instance_type = "t2.micro"
208     key_name = "capstone-key"
209     security_groups = [aws_security_group.test-capstone-sg.name]
210     user_data = <<-EOF
211     #!/bin/bash
212
213     # Install Docker
214     sudo apt-get update
215     sudo apt-get install -y \
216         ca-certificates \
217         curl \
218         gnupg
219     sudo apt install git python3 python3-pip -y
220     sudo install -m 0755 -d /etc/apt/keyrings
221     curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
222     sudo chmod a+r /etc/apt/keyrings/docker.gpg
223     echo \
224     "deb [arch=\"$dpkg --print-architecture\"] signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
225     \"$!. /etc/os-release && echo \"$VERSION_CODENAME\" stable" | \
226
227     EOF
228
229     tags = {
230         Name = "test-capstone"
231         Project = "Capstone"
232     }
233 }
234
235
236 # Creating Production Server
237 # Properties -
238 #   - Instance type: t2.micro
239 #   - AMI: Ubuntu 20.04 (LTS)
240 #   - Security Group: prod-capstone-sg
241 #   - Key Pair: capstone-key
242 resource "aws_instance" "prod-capstone" {
243     ami = "ami-0aa2b772dc1b5612"
244     instance_type = "t2.micro"
245     key_name = "capstone-key"
246     security_groups = [aws_security_group.prod-capstone-sg.name]
247     user_data = <<-EOF
248     #!/bin/bash
249
250     # Install Docker
251     sudo apt-get update
252     sudo apt-get install -y \
253         ca-certificates \
254         curl \
255         gnupg
256     sudo apt install git python3 python3-pip -y
257     sudo install -m 0755 -d /etc/apt/keyrings
258     curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
259     sudo chmod a+r /etc/apt/keyrings/docker.gpg
260     echo \
261     "deb [arch=\"$dpkg --print-architecture\"] signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
262     \"$!. /etc/os-release && echo \"$VERSION_CODENAME\" stable" | \
263     sudo tee /etc/apt/sources.list.d/docker.list >/dev/null
264     sudo apt-get update
265     sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
266
267     # # Install Minikube
268     # curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
269     # sudo dpkg -i minikube_latest_amd64.deb
270
271

```

```
main.tf > resource "aws_security_group" "test-capstone-sg"
271
272  # # Install Kubernetes
273  # sudo apt-get update
274  # sudo apt-get install -y ca-certificates curl
275  # sudo curl -fsSlo /etc/apt/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
276  # echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
277  # sudo apt-get update
278  # sudo apt-get install -y kubectl
279 EOF
280
281 tags = {
282   Name = "prod-capstone"
283   Project = "Capstone"
284 }
285 }
```

Kubernetes

Ln 61, Col 1 Spaces: 4 UTF-8 LF {} Terraform 8 🔍 🔒

What is on GitHub?

- Code files
- Ansible Playbook named “test-deploy.yml” in ./scripts directory in repository
- Jenkinsfile named “Jenkinsfile” in root of repository
- Dockerfile to build docker image

What is happening?

- Developer pushes the code on a branch
- Jenkins checks which branch the code is pushed to
- Go to that branch
 - Search for a Jenkinsfile in the repository
 - Run the pipeline mentioned in the Jenkinsfile

Jenkinsfiles

In develop branch

- Fetches the code of develop branch of <https://github.com/iamangangwar/iiitv-devops-capstone.git> (this repository is forked from the given repository, i.e. <https://github.com/hshar/website.git>)
- Performs testing
- Builds image using Dockerfile in the repository
 - Images built are
 - iamangangwar/iiitv-devops-capstone-test:\${BUILD_NUMBER}
 - iamangangwar/iiitv-devops-capstone-test:latest
- Uploads the images to DockerHub to the corresponding repositories
- Search for ansible playbook named "test-deploy.yml" in scripts directory in repository
 - Ansible playbook does -
 - Remotely pull the image iamangangwar/iiitv-devops-capstone-test:latest from DockerHub
 - Run the container from the pulled image
 - If a container is already running
 - Remove the previous container
 - Run new container

In master branch

- Fetches the code of develop branch of <https://github.com/iamangangwar/iiitv-devops-capstone.git> (this repository is forked from the given repository, i.e. <https://github.com/hshar/website.git>)
- Performs testing
- Builds image using Dockerfile in the repository
 - Images built are
 - iamangangwar/iiitv-devops-capstone-prod:\${BUILD_NUMBER}
 - iamangangwar/iiitv-devops-capstone-prod:latest
- Uploads the images to DockerHub to the corresponding repositories
- If a swarm already doesn't exist
 - Create a swarm
- Else
 - Update existing swarm

Jenkinsfile (in develop branch)

```
minikube default △ Kubernetes Jenkinsfile
1 pipeline {
2     agent any
3
4     environment {
5         DOCKERHUB_CREDENTIALS = credentials('dockerhub')
6     }
7     stages [
8         stage('Fetch Code') {
9             steps {
10                 git branch: 'develop', url: "https://github.com/iamangangwar/iiity-devops-capstone.git"
11             }
12         }
13
14         stage('Test Dockerfile') {
15             steps {
16                 sh "docker build -t iamangangwar/iiity-devops-capstone-test:${BUILD_NUMBER} ."
17             }
18         }
19
20         stage('Build Image') {
21             steps {
22                 script {
23                     sh "docker build -t iamangangwar/iiity-devops-capstone-test:${BUILD_NUMBER} ."
24                     sh "docker build -t iamangangwar/iiity-devops-capstone-test:latest ."
25                 }
26             }
27         }
28
29         stage('Upload Image') {
30             steps {
31                 script {
32                     sh "docker login -u ${DOCKERHUB_CREDENTIALS_USR} -p ${DOCKERHUB_CREDENTIALS_PSW}"
33                     sh "docker push iamangangwar/iiity-devops-capstone-test:${BUILD_NUMBER}"
34                     sh "docker push iamangangwar/iiity-devops-capstone-test:latest"
35                 }
36             }
37         }
38
39         stage('Deploy on Test Server') {
40             stages {
41                 stage('Run Ansible Playbook') {
42                     steps {
43                         ansiblePlaybook credentialsId: 'ansible-master', sudoUser:'root', disableHostKeyChecking: true, installation: 'ansible'
44                     }
45                 }
46             }
47         }
48
49
50         post {
51             always {
52                 sh "docker logout"
53             }
54         }
55     }
56 }
```

Ln 13, Col 1 Spaces: 4 UTF-8 LF Groovy ⚙️ 🔍

```
minikube default △ Kubernetes Jenkinsfile
45
46
47
48
49
50
51
52
53
54
55 }
```

Ln 13, Col 1 Spaces: 4 UTF-8 LF Groovy ⚙️ 🔍

Jenkinsfile (in master branch)

```
② Jenkinsfile
 1 pipeline {
 2     agent any
 3
 4     environment {
 5         DOCKERHUB_CREDENTIALS = credentials('dockerhub')
 6     }
 7     stages {
 8         stage('Fetch Code') {
 9             steps {
10                 git branch: 'master', url: 'https://github.com/iamangangwar/iiitv-devops-capstone.git'
11             }
12         }
13
14         stage('Test Dockerfile') {
15             steps {
16                 sh "docker build -t iamangangwar/iiitv-devops-capstone-prod:${BUILD_NUMBER} ."
17             }
18         }
19
20         stage('Build Image') {
21             steps {
22                 script {
23                     sh "docker build -t iamangangwar/iiitv-devops-capstone-prod:${BUILD_NUMBER} ."
24                     sh "docker build -t iamangangwar/iiitv-devops-capstone-prod:latest ."
25                 }
26             }
27         }
28
29         stage('Upload Image') {
30             steps {
31                 script {
32                     sh "docker login -u ${DOCKERHUB_CREDENTIALS_USR} -p ${DOCKERHUB_CREDENTIALS_PSW}"
33                     sh "docker push iamangangwar/iiitv-devops-capstone-prod:${BUILD_NUMBER}"
34                     sh "docker push iamangangwar/iiitv-devops-capstone-prod:latest"
35                 }
36             }
37         }
38
39         stage('Deploy on Production Server') {
40             steps {
41                 script {
42                     try {
43                         sh "docker service create \
44                             --name site \
45                             --publish published=80,target=80 \
46                             --replicas 3 \
47                             iamangangwar/iiitv-devops-capstone-prod:latest"
48                     }
49                     catch(e) {
50                         sh "docker service update \
51                             --image iamangangwar/iiitv-devops-capstone-prod:latest \
52                             site"
53                     }
54                 }
55             }
56         }
57     }
58     post {
59         always {
60             sh "docker logout"
61         }
62     }
63 }
```

Ln 18, Col 10 Spaces: 4 UTF-8 LF Groovy ⚙️ 🔍

```
② Jenkinsfile
 1 pipeline {
 2     agent any
 3
 4     environment {
 5         DOCKERHUB_CREDENTIALS = credentials('dockerhub')
 6     }
 7     stages {
 8         stage('Fetch Code') {
 9             steps {
10                 git branch: 'master', url: 'https://github.com/iamangangwar/iiitv-devops-capstone.git'
11             }
12         }
13
14         stage('Test Dockerfile') {
15             steps {
16                 sh "docker build -t iamangangwar/iiitv-devops-capstone-prod:${BUILD_NUMBER} ."
17             }
18         }
19
20         stage('Build Image') {
21             steps {
22                 script {
23                     sh "docker build -t iamangangwar/iiitv-devops-capstone-prod:${BUILD_NUMBER} ."
24                     sh "docker build -t iamangangwar/iiitv-devops-capstone-prod:latest ."
25                 }
26             }
27         }
28
29         stage('Upload Image') {
30             steps {
31                 script {
32                     sh "docker login -u ${DOCKERHUB_CREDENTIALS_USR} -p ${DOCKERHUB_CREDENTIALS_PSW}"
33                     sh "docker push iamangangwar/iiitv-devops-capstone-prod:${BUILD_NUMBER}"
34                     sh "docker push iamangangwar/iiitv-devops-capstone-prod:latest"
35                 }
36             }
37         }
38
39         stage('Deploy on Production Server') {
40             steps {
41                 script {
42                     try {
43                         sh "docker service create \
44                             --name site \
45                             --publish published=80,target=80 \
46                             --replicas 3 \
47                             iamangangwar/iiitv-devops-capstone-prod:latest"
48                     }
49                     catch(e) {
50                         sh "docker service update \
51                             --image iamangangwar/iiitv-devops-capstone-prod:latest \
52                             site"
53                     }
54                 }
55             }
56         }
57     }
58     post {
59         always {
60             sh "docker logout"
61         }
62     }
63 }
```

Ln 18, Col 10 Spaces: 4 UTF-8 LF Groovy ⚙️ 🔍

Dockerfile

```
🐳 Dockerfile > ...
1   FROM hshar/webapp
2
3   # Set the working directory to /var/www/html
4   WORKDIR /var/www/html
5
6   # Copy the source code into the container
7   COPY . .
8
9   # Expose port 80 for HTTP traffic
10  EXPOSE 80
11
12  # Start Apache web server when the container starts
13  CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

Ansible Playbook

```
scripts > / test-deploy.yml > {} 0 > [ ] tasks > {} 1 > [ ] rescue > {} 1 > {} docker_container > [ ] ports > [ ] 0
1   - hosts: test
2     tasks:
3       - name: Pull Docker Image
4         docker_image:
5           name: iamangangwar/iiitv-devops-capstone-test:latest
6           state: present
7           source: pull
8       - name: Run Container
9         block:
10        - name: Run Container
11          docker_container:
12            name: test
13            image: iamangangwar/iiitv-devops-capstone-test:latest
14            state: started
15            ports:
16              - "80:80"
17       rescue:
18         - name: Remove Previous Container
19           docker_container:
20             name: test
21             state: absent
22         - name: Run Container
23           docker_container:
24             name: test
25             image: iamangangwar/iiitv-devops-capstone-test:latest
26             state: started
27             ports:
28               - "80:80"
```

minikube default △ Kubernetes Ln 28, Col 20 Spaces: 2 UTF-8 LF YAML ⚙ No JSON Schema 🔍 ↻

Instance Screenshots

jenkins-capstone

The screenshot shows the AWS EC2 Instances page for the instance `i-0e81aaadf20f0c9c3 (jenkins-capstone)`. The instance is currently running. Key details include:

- Instance ID: `i-0e81aaadf20f0c9c3 (jenkins-capstone)`
- Public IPv4 address: `18.208.127.113`
- Private IP4 address: `172.31.85.237`
- Instance state: `Running`
- Private IP DNS name (IPv4 only): `ip-172-31-85-237.ec2.internal`
- Instance type: `t2.medium`
- VPC ID: `vpc-0003ad0d233c45c37`
- Subnet ID: `subnet-01f8eec9299c5e8e7`
- Platform: `Ubuntu (Inferred)`
- AMI ID: `ami-0aa2b7722dc1b5612`
- AMI name: `ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20230328`

prod-capstone

The screenshot shows the AWS EC2 Instances page for the instance `i-0554f0dad032c92a6 (prod-capstone)`. The instance is currently running. Key details include:

- Instance ID: `i-0554f0dad032c92a6 (prod-capstone)`
- Public IPv4 address: `18.212.76.94`
- Private IP4 address: `172.31.20.64`
- Instance state: `Running`
- Private IP DNS name (IPv4 only): `ip-172-31-20-64.ec2.internal`
- Instance type: `t2.micro`
- VPC ID: `vpc-0003ad0d233c45c37`
- Subnet ID: `subnet-088eb83d04fb70df3`
- Platform: `Ubuntu (Inferred)`
- AMI ID: `ami-0aa2b7722dc1b5612`
- AMI name: `ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20230328`

test-capstone

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Images, AMIs, and more. The main content area displays the instance summary for the instance i-0a30703c304302cb0. The summary includes details such as Instance ID, Public IPv4 address (34.224.66.153), Private IPv4 addresses (172.31.29.14), Instance state (Running), and VPC ID (vpc-0003ad0d233c45c37). Below the summary, there are tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the Details tab, there's a sub-section for Instance details with fields for Platform (Ubuntu (Inferred)), AMI ID (ami-0aa2b7722dc1b5612), and Monitoring (disabled). The status bar at the bottom indicates "© 2023, Amazon Web Services India Private Limited or its affiliates." and provides links for CloudShell, Feedback, Language, Privacy, Terms, and Cookie preferences.

Security Group Screenshots

The screenshot shows the AWS Security Groups page. The sidebar includes links for Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content shows the security group sg-03e34445364e2dee4 - jenkins-capstone-sg. It displays the Details section with fields for Security group name (jenkins-capstone-sg), Security group ID (sg-03e34445364e2dee4), Description (Security Group for Jenkins Server for Capstone Project), VPC ID (vpc-0003ad0d233c45c37), Owner (583571788342), Inbound rules count (2 Permission entries), and Outbound rules count (1 Permission entry). Below the Details section, there are tabs for Inbound rules, Outbound rules, and Tags. A message box says "You can now check network connectivity with Reachability Analyzer" and has a "Run Reachability Analyzer" button. The Inbound rules table shows two entries: one for Custom TCP on port 8080 from 0.0.0.0/0 allowing port 8080, and another for SSH on port 22 from 103.81.93.134/32 allowing SSH from my IP. The status bar at the bottom indicates "© 2023, Amazon Web Services India Private Limited or its affiliates." and provides links for CloudShell, Feedback, Language, Privacy, Terms, and Cookie preferences.

Screenshot of the AWS Management Console showing the details of a Security Group named "sg-032a07d46c71c0f42 - prod-capstone-sg".

Details:

Security group name prod-capstone-sg	Security group ID sg-032a07d46c71c0f42	Description Security Group for Production Server for Capstone Project	VPC ID vpc-0003ad0d233c45c37
Owner 583571788342	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules (3):

Type	Protocol	Port range	Source	Description
HTTP	TCP	80	0.0.0.0/0	Allow port 80 from an...
SSH	TCP	22	103.81.93.134/32	Allow SSH from my IP
All traffic	All	All	sg-03e34445364e2de...	Allow all traffic from J...

sg-07f21dae990e18114 - test-capstone-sg

Details:

Security group name test-capstone-sg	Security group ID sg-07f21dae990e18114	Description Security Group for Test Server for Capstone Project	VPC ID vpc-0003ad0d233c45c37
Owner 583571788342	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules (3):

Type	Protocol	Port range	Source	Description
HTTP	TCP	80	0.0.0.0/0	Allow port 80 from an...
SSH	TCP	22	103.81.93.134/32	Allow SSH from my IP
All traffic	All	All	sg-03e34445364e2de...	Allow all traffic from J...

Setup Ansible Result

```
● ● ● capstone — root@ip-172-31-85-237: ~ — ssh -i capstone-key ubuntu@4...
[root@ip-172-31-85-237:~# ansible all -m ping -v
Using /etc/ansible/ansible.cfg as config file
The authenticity of host '172.31.29.14 (172.31.29.14)' can't be established.
ECDSA key fingerprint is SHA256:TQgVqiBhYhrH2po2GWrabU/0yIFhtT2z8T+zMGuTDwc.
The authenticity of host '172.31.20.64 (172.31.20.64)' can't be established.
ECDSA key fingerprint is SHA256:vAls7Niv1t0GgmG8sAoblrHQoJd95RR4UgMqDW3GTTo.
[Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
[Please type 'yes', 'no' or the fingerprint: yes
[Please type 'yes', 'no' or the fingerprint: yes
test1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[yes
prod1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[root@ip-172-31-85-237:~# ansible test -m ping -v
Using /etc/ansible/ansible.cfg as config file
test1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[root@ip-172-31-85-237:~# ansible prod -m ping -v
Using /etc/ansible/ansible.cfg as config file
prod1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[root@ip-172-31-85-237:~# _
```

Docker Swarm Setup Result

Leader is **jenkins-capstone** instance, and worker is **prod-capstone** instance

```
[root@ip-172-31-85-237:~# docker node ls
ID           HOSTNAME   STATUS  AVAILABILITY  MANAGER STATUS  ENGINE VERSION
pzkrqiw95zf0dhssz4lxssp  ip-172-31-20-64  Ready   Active        Leader        23.0.4
uf7nih98fq0ws9rybm2d3ejuz * ip-172-31-85-237  Ready   Active        Active        23.0.4
root@ip-172-31-85-237:~# ]
```

Jenkins Multibranch Pipeline Configuration

The screenshot shows the Jenkins Multibranch Pipeline configuration page for the 'iiitv-devops-capstone' project. The 'Branch Sources' tab is selected. Under 'Branch Sources', there is a 'GitHub' section with a dropdown menu set to 'none'. A red 'Add' button is available to add credentials. A warning message states 'Credentials are recommended'. Below this, a radio button is selected for 'Repository HTTPS URL', with the URL 'https://github.com/iamangangwar/iiitv-devops-capstone.git' entered into the input field. A 'Validate' button is present. There is also an option for 'Repository Scan - Deprecated Visualization'. At the bottom of the 'Branch Sources' section are 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins Multibranch Pipeline configuration page for the 'iiitv-devops-capstone' project. The 'Build Configuration' tab is selected. Under 'Mode', the dropdown is set to 'by Jenkinsfile'. In the 'Script Path' field, 'Jenkinsfile' is specified. Under 'Scan Repository Triggers', the 'Scan by webhook' checkbox is checked, and the 'Trigger token' field contains 'capstone-token'. At the bottom of the 'Build Configuration' section are 'Save' and 'Apply' buttons.

Github Webhook

The screenshot shows the GitHub repository settings for 'iiitv-devops-capstone'. The 'Webhooks' tab is selected. The payload URL is set to 'http://3.86.167.173:8080/multibranch-webhook-trigger/invoke?c'. The content type is 'application/json'. The secret field is empty. Under 'Which events would you like to trigger this webhook?', the 'Just the push event' radio button is selected. The 'Active' checkbox is checked. At the bottom, there are 'Update webhook' and 'Delete webhook' buttons.

Jenkins Pipeline Screenshots

The screenshot shows the Jenkins dashboard for the 'iiitv-devops-capstone' project. The left sidebar includes options like Status, Configure, Scan Repository Now, Scan Repository Log, Multibranch Pipeline Events, Delete Multibranch Pipeline, People, Build History, Project Relationship, Check File Fingerprint, GitHub, Rename, Pipeline Syntax, and Credentials. The main area displays the project name 'iiitv-devops-capstone'. A message states it is the repository of the capstone project of the 6th Semester DevOps course. It shows two branches: 'develop' (last success 2 hr 43 min ago, build #4) and 'master' (last success 2 hr 57 min ago, build #2). Buttons for 'Disable Multibranch Pipeline' and 'Branches (2)' are visible. At the bottom, there are links for Atom feed for all, Atom feed for failures, and Atom feed for just latest builds.

Develop branch -

Jenkins

Dashboard > iiitv-devops-capstone > develop >

Status

Branch develop

Full project name: iiitv-devops-capstone/develop

Changes

Build Now

View Configuration

Full Stage View

GitHub

Pipeline Syntax

Build History trend

Filter builds...

#4 Apr 23, 13:04 1 commit

#3 Apr 23, 12:45 1 commit

#2 Apr 23, 11:59 1 commit

#1 Apr 23, 11:25 No Changes

Atom feed for all Atom feed for failures

Average stage times: (Average full run time: ~2min 42s)

	Declarative: Checkout SCM	Fetch Code	Test Dockerfile	Build Image	Upload Image	Deploy on Test Server	Run Ansible Playbook	Declarative: Post Actions
#4	289ms	239ms	781ms	1s	3s	33ms	4s	315ms
#3	239ms	244ms	575ms	1s	3s	35ms	4s	313ms
#2	347ms	205ms	867ms	1s	3s	32ms	4s	316ms
#1	323ms	303ms	852ms	1s	4s	32ms	5s	317ms
	248ms	206ms	831ms	1s	3s	34ms	4s	314ms

Stage View

Permalinks

- Last build (#4), 2 hr 44 min ago
- Last stable build (#4) 2 hr 44 min ago

Master branch -

Jenkins

Dashboard > iiitv-devops-capstone > master >

Status

Branch master

Full project name: iiitv-devops-capstone/master

Changes

Build Now

View Configuration

Full Stage View

GitHub

Pipeline Syntax

Build History trend

Filter builds...

#2 Apr 23, 12:51 1 commit

#1 Apr 23, 11:59 No Changes

Atom feed for all Atom feed for failures

Average stage times: (Average full run time: ~5min 30s)

	Declarative: Checkout SCM	Fetch Code	Test Dockerfile	Build Image	Upload Image	Deploy on Production Server	Declarative: Post Actions
#2	232ms	248ms	835ms	1s	3s	48s	320ms
#1	227ms	262ms	838ms	1s	3s	48s	314ms
	237ms	235ms	833ms	1s	3s	48s	327ms

Stage View

Permalinks

- Last build (#2), 2 hr 57 min ago
- Last stable build (#2), 2 hr 57 min ago
- Last successful build (#2), 2 hr 57 min ago
- Last completed build (#2), 2 hr 57 min ago

Docker Containers running

test-capstone instance

```
[root@ip-172-31-29-14:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
20c4a4fc997e iamangangwar/iiitv-devops-capstone-test:latest "/usr/sbin/apache2ct..." 23 minutes ago
Up 23 minutes 0.0.0.0:80->80/tcp test]
```

Docker Swarm running

```
[root@ip-172-31-85-237:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
371b6755b10 iamangangwar/iiitv-devops-capstone-prod:latest "/usr/sbin/apache2ct..." 42 minutes ago Up 42 minutes 80/tcp site.1.rzcc7lrd75htvux3ml3u7hj2b
630438a102e8 iamangangwar/iiitv-devops-capstone-prod:latest "/usr/sbin/apache2ct..." 3 hours ago Up 3 hours 80/tcp site.3.ruphx3t4p35a3uc4u7paytzc
f977e41f8306 iamangangwar/iiitv-devops-capstone-prod:latest "/usr/sbin/apache2ct..." 3 hours ago Up 3 hours 80/tcp site.2.qad5es95iqgvvqds083eyvkcb
root@ip-172-31-85-237:~#]
```

Running on Production Instance



GitHub

Instance: i-0554f0dad032c92a6 (prod-capstone)

Details Security Networking Storage Status checks Monitoring Tags

▼ Instance summary Info

Instance ID i-0554f0dad032c92a6 (prod-capstone)	Public IPv4 address 204.236.198.248 open address	Private IPv4 addresses 172.31.20.64
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-204-236-198-248.compute-1.amazonaws.com open address

Creating status.txt in /home/ubuntu/config-management

Shell Script -

```
capstone — root@ip-172-31-20-64: ~ — ssh -i capstone-key ubuntu@204.236.198.248 — 98...
root@ip-172-31-20-64:~# vim /var/apache2-check.sh
root@ip-172-31-20-64:~# cat /var/apache2-check.sh
#!/bin/bash

mkdir -p /home/ubuntu/config-management
touch /home/ubuntu/config-management/status.txt

sudo systemctl restart apache2

if [ $? == 0 ]
then
    echo "Apache is installed on this system" > /home/ubuntu/config-management/status.txt
else
    echo "Apache is not installed on this system" > /home/ubuntu/config-management/status.txt
fi
root@ip-172-31-20-64:~# crontab -e
No modification made
root@ip-172-31-20-64:~# crontab -e
crontab: installing new crontab
root@ip-172-31-20-64:~# chmod +x /var/apache2-check.sh
root@ip-172-31-20-64:~#
```

Files in all 3 instances -

```
[root@ip-172-31-29-14:~# cat /home/ubuntu/config-management/status.txt
Apache is not installed on this system
root@ip-172-31-29-14:~# ]
```

```
[root@ip-172-31-85-237:~# cat /home/ubuntu/config-management/status.txt
Apache is not installed on this system
root@ip-172-31-85-237:~# ]
```

```
[root@ip-172-31-20-64:~# cat /home/ubuntu/config-management/status.txt
Apache is not installed on this system
root@ip-172-31-20-64:~# ]
```

Apache is not installed on any of the systems because I have deployed docker containers containing apache2 server, hence apache2 needs not be installed in the virtual machine.

All the scripts I used are present in my GitHub repository :)

<https://github.com/iamangangwar/iiitv-devops-capstone-config.git>