

BATCH3.0 START DATE – MAY 27TH 2023 [2 MONTHS COURSE]

Link to enroll - <https://bit.ly/ULTIMATEDEVOPS3>

[COUPON CODE - **LIVE30**] Replace the existing code with new code for the biggest discount till date, Code valid for 24 hours

Register now seats are getting filled fast

- 1) WEEKEND LIVE ONLINE CLASSES – SATURDAY AND SUNDAY [3 HOURS EACH SESSION]
- 2) 12+ REAL TIME HANDS-ON PROJECTS
- 3) EXTRA DOUBT CLEARING SESSIONS
- 4) EXTRA PROJECTS WITH DOCUMENTS
- 5) JOB SUPPORT VIA FEW EXTRA SESSIONS AFTER COMPLETION OF COURSE
- 6) JOB SUPPORT VIA RESUME TO HR DIRECT CONNECT
- 7) RESUME REVIEWS / MOCK INTERVIEWS / LIVE FEEDBACK SHARE TO STUDENTS / INTERVIEW QUESTIONS TOOL WISE DISCUSSION

PROJECT HANDSON TERRAFORM WITH AWS






CODE - https://github.com/DEVOPS-WITH-WEB-DEV/Terraform_AWS_Session7.git

Step 1 – Create the AWS CLOUD9 Environment in **us-west-1**

AWS Cloud9 > Environments

Environments (1) Delete View details Open in Cloud9 Create environment

My environments ▼ < 1 > ⚙

	Name ▲	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
	Terraform session7	 Open	EC2 instance	AWS Systems Manager (SSM)	Owner	 arn:aws:iam::600735812827

Step2 – Go into your cloud9 environment and configure AWS CLI

Step 3 – Run aws configure command

```
python2 - "ip-172-31-12-5i x"
ec2-user:~/environment $ aws configure
AWS Access Key ID [*****P3DB]: AKIAI7XXV1BTH4DBLERD7L
AWS Secret Access Key [*****3tVv]: lmcub4ngat745x4W56320N15qmE0
Default region name [us-west-1]:
Default output format [None]:
ec2-user:~/environment $
```

Step 3.1 – Create IAM user if you don't have and get the access and secret key

Specify user details

User details

User name

☐ Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

ⓘ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

Step 4 – Create the folder structure

```
ec2-user:~/environment $ mkdir terraform
ec2-user:~/environment $ cd terraform/
ec2-user:~/environment/terraform $ mkdir project1
ec2-user:~/environment/terraform $ cd project1/
ec2-user:~/environment/terraform/project1 $ pwd
/home/ec2-user/environment/terraform/project1
ec2-user:~/environment/terraform/project1 $ █
```

Step 5 – Create a main.tf file to create EC2 instance and go to EC2-> AMI CATALOG -> Select the AMI from AMAZON LINUX 2023

vim main.tf and add below commands:

```
#Configure the AWS Provider
provider "aws" {
  region = "us-west-1"
}
#Create EC2 Instance
resource "aws_instance" "instance1" {
  ami           = "ami-051ed863837a0b1b6"
  instance_type = "t2.micro"
  vpc_security_group_ids = [aws_security_group.jenkins_sg.id]
  tags = {
    Name = "jenkins_instance"
  }
}

#Bootstrap Jenkins installation and start [ AUTOMATION ]
user_data = <<-EOF
#!/bin/bash
sudo yum update
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum upgrade
amazon-linux-extras install epel -y
sudo dnf install java-11-amazon-corretto -y
sudo yum install jenkins -y
sudo systemctl enable jenkins
sudo systemctl start jenkins
EOF

user_data_replace_on_change = true
}
```

The `<<-EOF` and `EOF` are Terraform's *heredoc* syntax and allows you to create multi-line strings without having to use the `\n` characters. The `user_data_replace_on_change = true` parameter means that if the `user_data` parameter is changed and you run `terraform apply`, Terraform will terminate the current instance and launch a new one to reflect the new user data script.

Step 6 – Create a Security Group

```
#Create security group
resource "aws_security_group" "jenkins_sg" {
  name      = "jenkins_sg"
  description = "Open ports 22, 8080, and 443"

  #Allow incoming TCP requests on port 22 from any IP
  ingress {
    description = "Incoming SSH"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  #Allow incoming TCP requests on port 8080 from any IP
  ingress {
    description = "Incoming 8080"
    from_port   = 8080
    to_port     = 8080
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  #Allow incoming TCP requests on port 443 from any IP
  ingress {
    description = "Incoming 443"
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  #Allow all outbound requests
```

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

tags = {
  Name = "jenkins_sg"
}
}
```

Step 6 – Assign SG to Ec2 instance

Step 7 – Create a S3 bucket for Jenkins Artifacts

```
#Create S3 bucket for Jenkins artifacts
resource "aws_s3_bucket" "jenkins-artifacts" {
  bucket = "jenkins-artifacts-12345abc"
  tags = {
    Name = "jenkins_artifacts"
  }
}
```

Step 8 – Run the terraform command

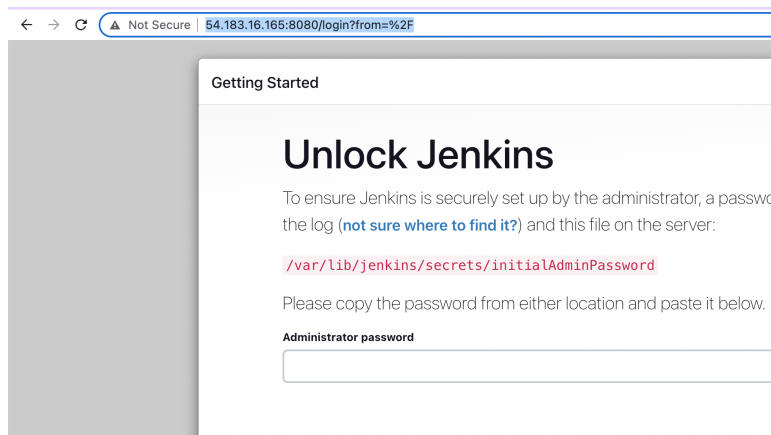
```
terraform init
```

```
terraform validate
```

```
terraform plan -out plan.out
```

```
terraform apply "plan.out"
```

Step 9 – Hit the endpoint for Jenkins



<http://<EC2IP>:8080/login?from=%2F>

Step 10 - terraform destroy

Step 11 – Delete cloud9 environment

