

## Case study :terraform

*you work as a DevOps Engineer in a leading software company.*

*You have been asked to build an infrastructure safely and efficiently. The company's requirements:*

- 1. Use AWS Cloud Provider and the software to be installed is Apache2*
- 2. Use Ubuntu AMI*

*The company wants the architecture to have the following services:*

- 1. Create a template with a VPC, 2 subnets and 1 instance in each subnet*
- 2. Attach security groups, internet gateway and network interface to the instance*

*Solution*

- 1. First creat the aws a/c setups*

*Aws account with the necessary permission for resource creation and management.*

- 2. Next stept to creat the aws vpc and cidr block 10.0.0./16*

*Two sebnet in vpc with different availability zone*

- 3. So next seep to create the security groups in the https,ports 22 number*
- 4. Let launch the internet gateway in the vpc attact in public subnets*
- 5. Lauch the ec2 instances in ubuntu ami with 3 seps to create*

```
* Management:      https://landscape.canonical.com
* Support:         https://ubuntu.com/advantage

System information as of Tue Oct 24 11:00:00 UTC 2023

System load:  0.0               Processes:            101
Usage of /:   26.0% of 7.57GB   Users logged in:     0
Memory usage: 23%              IPv4 address for eth0: 172.31.33.119
Swap usage:   0%
```

Expanded Security Maintenance for Applications is not enabled.

33 updates can be applied immediately.

23 of these updates are standard security updates.

To see these additional updates run: `apt list --upgradable`

Enable ESM Apps to receive additional future security updates.

See <https://ubuntu.com/esm> or run: `sudo pro status`

Last login: Tue Oct 24 10:50:40 2023 from 13.233.177.5

ubuntu@ip-172-31-33-119:~\$

i-0a898112afe125956 (terraform)

PublicIPs: 13.127.250.46 PrivateIPs: 172.31.33.119

```

1  sudo yum update
2  sudo apt-get update
3  sudo apt-get install docker
4  wget <a href="https://releases.hashicorp.com/terraform_1.3.5_linux_amd64.zip">https://releases.hashicorp.com/terraform_1.3.5_linux_amd64.zip</a>
5  echo $PATH
6  sudo unzip terraform_1.3.5_linux_amd64.zip -d /usr
7  ls -l /usr/local/bin
8  terraform -version
9  sudo snap install terraform
10 sudo apt-get install terraform
11 sudo yum install -y yum-utils
12 clear

```

```

16 unzip terraform_1.0.7_linux_amd64.zip
17 sudo mv terraform /usr/local/bin/
18 terraform --version
19 clear
20 kubectl get job -o json
21 sudo apt-get install kubectl
22 kubectl get job -o json
23 sudo -i
24 sudo mv terraform /usr/local/bin
25 history
ubuntu@ip-172-31-33-119:~$

```

i-0a898112afe125956 (terraform)

PublicIPs: 13.127.250.46 PrivateIPs: 172.31.33.119

So we launch the terraform successful

```

ubuntu@ip-172-31-33-119:~$ terraform --version
Terraform v1.0.7
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.6.2. You can update by downloading from https://www.terraform.io/downloads.html

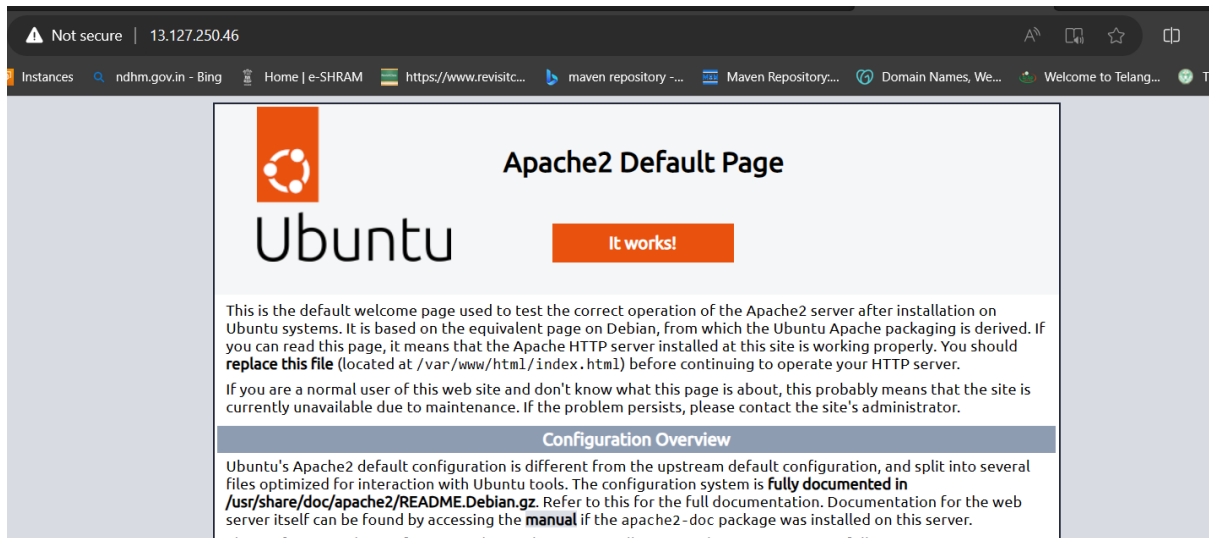
```

Now last step to create the apache install and terform in the ubuntu in the aws so  
Commands;

1. "sudo apt-get update"
2. "sudo apt-get install apache2"
3. "sudo apt-get start apache2"

Then test the public ip addres in the https hosts and dns server running correctly  
So additional configuration in the apache2 and instances in second setup pages

*Let check it work or not*



*So out put is correct and successfully to launch apache2*