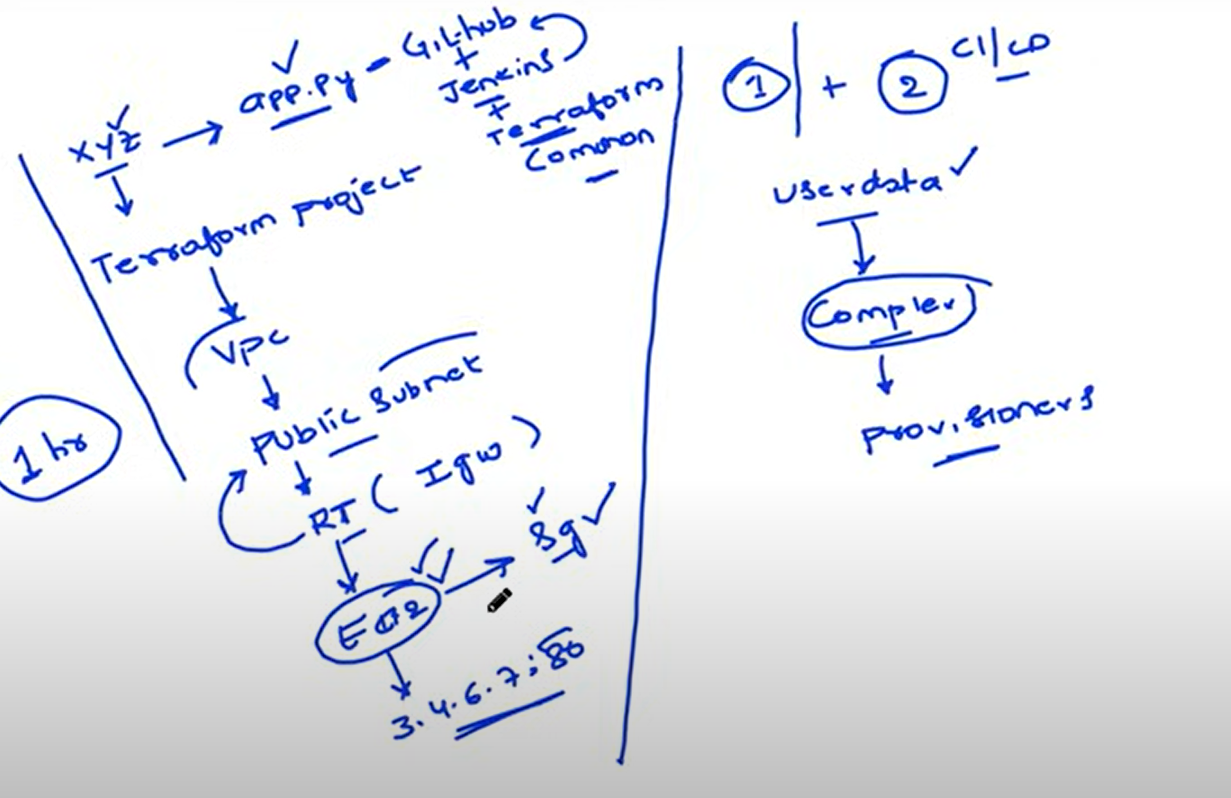
**Terraform Task-1**



The first part is Building the terraform project, app.py deployment to EC2

The second part is - integrate with CI/CD

There is a development team …they have app.py file.  
Every time they make changes to app.py file.

Devops to create terraform project

To test their changes they want to create VPC, Subnet, route table…destination to route is IGW. We have Associate RT with subnet, so that it will become public subnet.

You need to create EC2 instance, and deploy app.py and expose to external world.

Once they hit EC2 Public ip , application should be accessible.

EC2 User data- for small scripts  
When you want to execute complex scripts on EC2, you will use provisioners.

**App.py consists of app code.**

**Main.tf consists of complete terraform code.**

To create AWS key pair from terraform file, we have to generate key in our machine using below command  
**ssh-keygen -t rsa**

resource "aws\_key\_pair" "example" {

  key\_name   = "terraform-task1"  # Replace with your desired key name

  public\_key = file("~/.ssh/id\_rsa.pub")  # Replace with the path to your public key file

}

**To run the terraform project execute below commands**

Terraform init  
Terraform validate  
Terraform Plan

Terraform apply

Terraform destroy

**After creating infra and deployed EC2 using file, remote-exec provisioners, we can login to server and check.**

ssh -i ~/.ssh/id\_rsa ubuntu@pubip of created instance

**To access the app in browser**

pubip of created instance:80