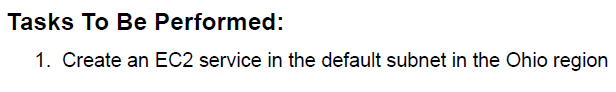
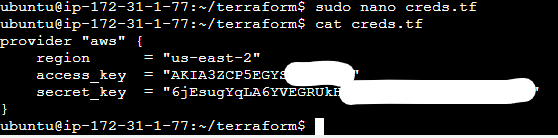
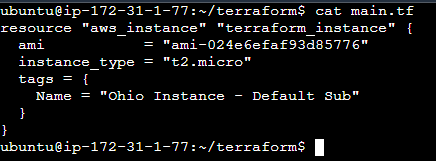
**Terraform Assignment – 1**



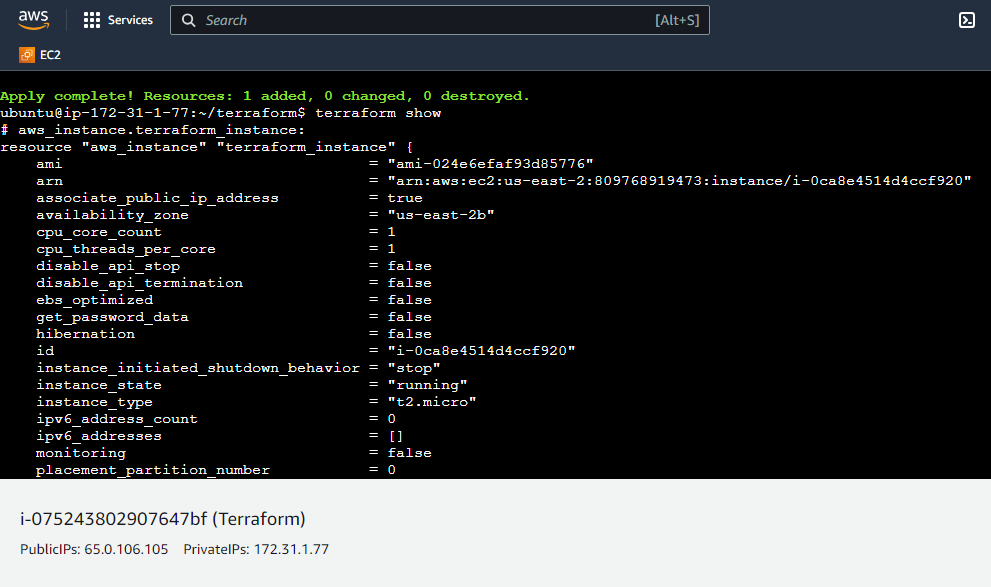
1. Setup Terraform Locally / Cloud. I am using AWS Instance to setup Terraform.
2. Create a file as “creds.tf” which will store the default region and the Credentials (i.e Access and Secret Access Key)

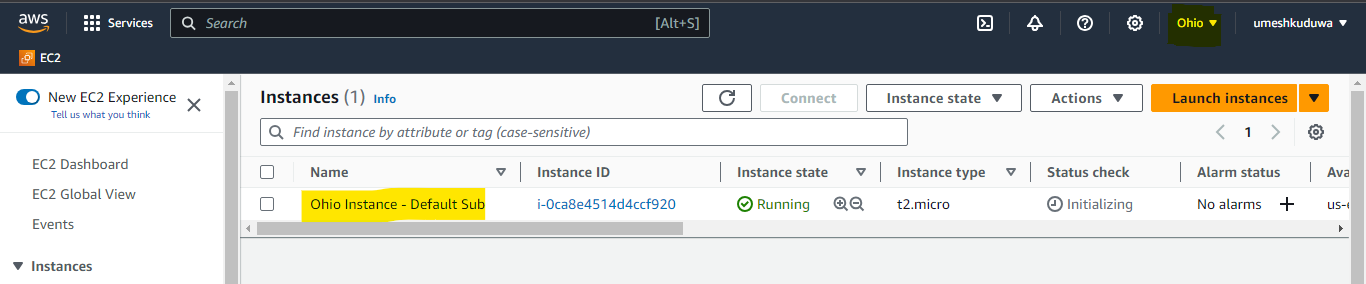


1. Create another file with “.tf” extension which will have the infrastructure code. Requirement is to create a EC2 Instance in Ohio Region.

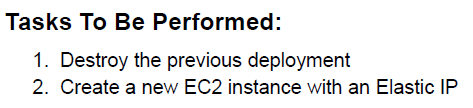


1. Plan and Apply the by using terraform plan followed by terraform apply. To verify, we can use terraform show command or go to Management Console and verify.

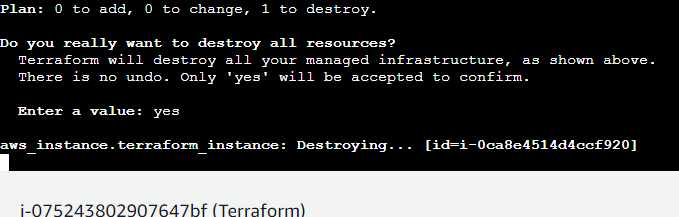




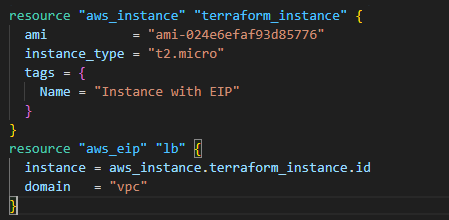
**Terraform Assignment – 2**



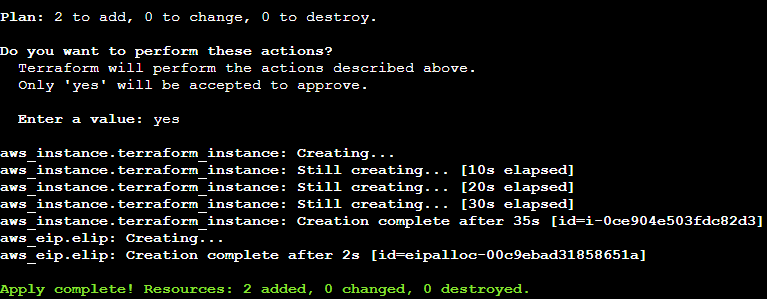
1. To destroy the previous deployment, type the terraform command : terraform destroy.



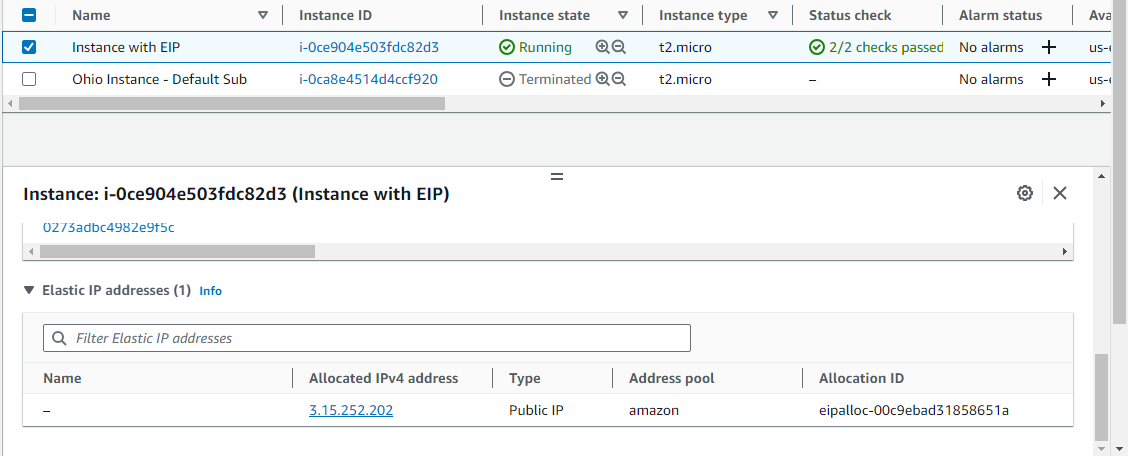
1. You can use the same template and modify the script to add in Elastic IP.

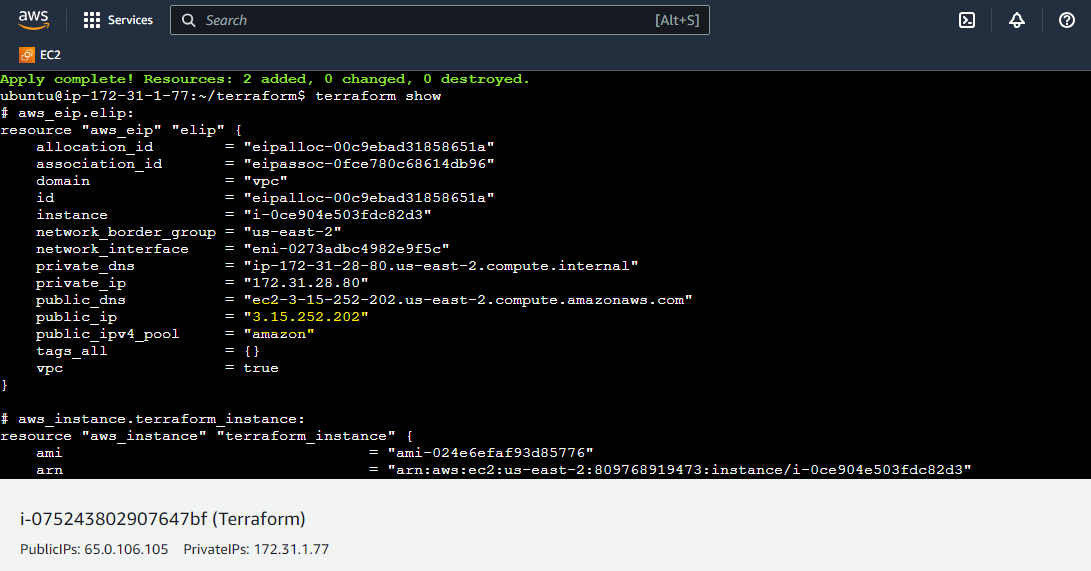


1. Save the script, and proceed to terraform plan. Review the changes and apply it by terraform apply command.

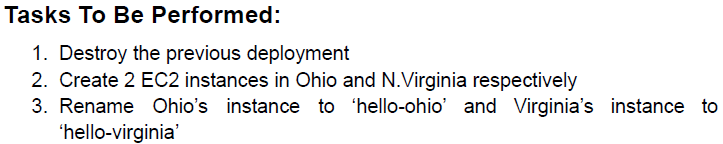


1. Again, verify the details by going to management console / terraform show.

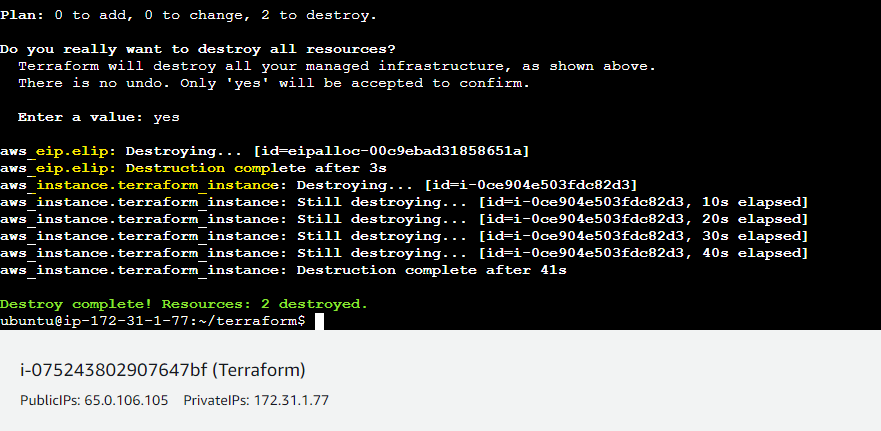




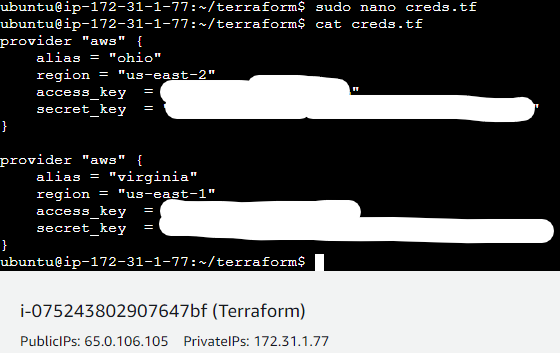
**Terraform Assignment – 3**



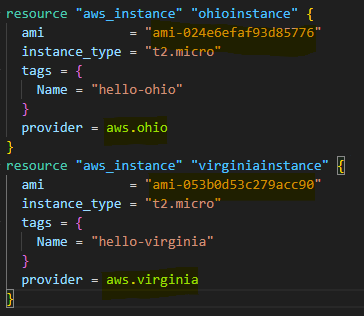
1. Destroy the previous deployment by using terraform destroy command.



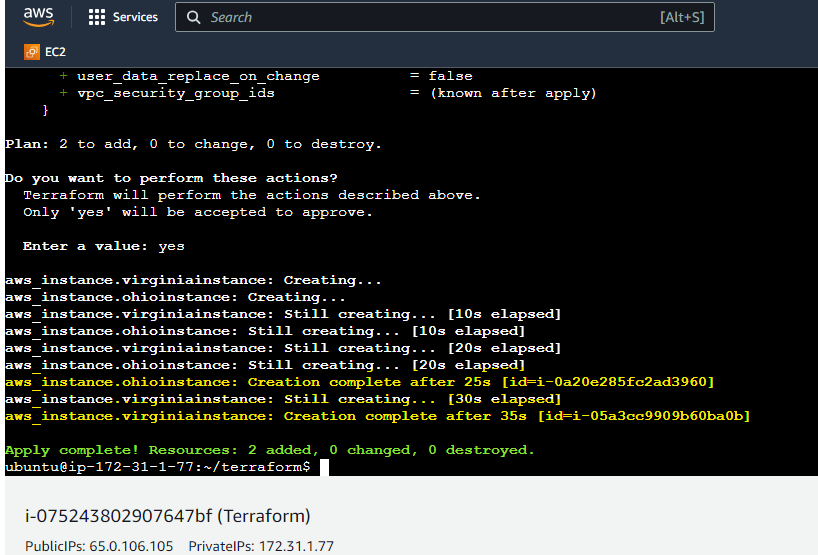
1. Modify the creds.tf to add in alias such that it would pick inside the resource to which region it should deploy the resources to.



1. Modify the main.tf script to satisfy assignment requirements.



1. If you observe the above script, in AWS AMI is Region Dependent. That’s why you need to search for AMI in each region and input in the script. Also the provider block inside resource would call the provider block from creds.tf where we had mentioned 2 region and aliased with virginia and ohio respectively.
2. Plan and Apply the code and observe the result



1. Verify it via management console.

