Devops Exercises

Please pick ***just 2*** of the exercises below and complete them. Once you have completed your submission, place it in a github repo that you share.

We will schedule a time to review your solution with you after we have had a chance to review it internally.

AWS VPC Exercise

You have been assigned a project to create a new terraform module for creating a staging VPC in AWS. The following requirements have been given to you.

- The network supernet of 172.16.0.0/16
- The VPC should provide an endpoint for AWS Systems Manager (SSM), internally.
- The VPC should provide an internal endpoint for S3.
- The VPC should have two different availability zones, with two private subnets and two public subnets for each availability zone.
- The VPC should contain appropriate routing tables.
- The VPC should include appropriate NAT gateways and an internet gateway.

Create module code to fulfill this request utilizing terragrunt/terraform. Show how you would use this module code to bring up the given architecture.

A Small EC2 App

Using the following technologies:

- Packer
- Ansible
- Terraform/terragrunt.

- Utilizing a pack/fry idiom in ansible, create an Amazon Machine Image (AMI) which runs a small nginx webserver. (A static page loading here is just fine).
    - It should be based on the latest Amazon Linux 2023 AMI.
    - Ensure there are both pack and fry roles and that the appropriate variables needed for the ami are injectable via userdata in the frying role. (if applicable)
- Create a terraform stack which creates the following configuration:
    - An autoscaling group of two Ec2 Instances
    - Launch template
        - Instance profile role along with manageable policy to be attached to the EC2 instances.

- An application load balancer (ALB)
    - A target group where you will place the ec2 instances.
- Security groups for the load balancer and ec2 instances with the proper rules to ensure ssh access to the instances and communication between the instances and the load balancer.
- Place the autoscaling group on a private subnet.
- Bonus points if you add transport layer security support via an SSL cert through AWS Certificate Manager (ACM) but not necessary for this exercise.

Deploying An Application

Helm

- Create a simple templated helm chart that loads up an nginx server and an appropriate ingress. (assume either an nginx or alb controller is just fine)
- Create a values.yml file that fulfills the values for this helm chart.
- Show a command that dumps the template generated.

Github Actions

- Create an example github action that would use Github's Open ID Connect (OIDC) provider as a trusted AWS Identity to deploy the helm chart above into a staging EKS cluster.