Tasks

The candidate will pick **two missions** to finish and can apply for a dev machine if needed.

Each mission can have multiple tasks, and finish as much as possible.

Upload all the files, including supporting documents (e.g., README), to a GitHub repository, and invite the following users to the repository:

- https://github.com/annihilatopia
- https://github.com/antonio-altr
- https://github.com/Exubient
- https://github.com/revolution1

Mission 1: Terraform: EKS

We have all of our infrastructure on AWS, and it is imperative that we handle our infrastructure in a stateful manner via tools such as Terraform. Your objective here is to show a basic understanding of deploying simple components to AWS.

We also maintain a EKS cluster on AWS to handle all our app deployments. Thus, we need to use Terraform to deploy to EKS.

Tasks:

1. Write a Terraform script to deploy EKS.
    1. Use S3 to store the Terraform state file.

    2. Please include in your Terraform scripts the deployment of any additional required components.

        Hint: Please consider all AWS components an EKS deployment requires.

References:

https://registry.terraform.io/providers/hashicorp/aws/latest/docs

https://aws.amazon.com/s3/

https://aws.amazon.com/eks/

Mission 2: CICD

After deployment of the EKS cluster, we also want to make sure all our apps are deployed in a stateful manner, via GitOps.

Tasks:

1. Show how you would install FluxCD onto the EKS cluster created via Mission 1.

References:

https://fluxcd.io/

Mission 3: EKS Monitoring

After deployment of the EKS cluster, we need to implement some sort of monitoring tool onto the cluster. There are many solutions out there, but we want to deploy specifically kube-prometheus-stack onto the cluster.

Tasks:

1. Deploy kube-prometheus-stack via FluxCD.

References:

https://github.com/prometheus-community/helm-charts/tree/main/charts/kube-prometheus-stack

Mission 4: Sample App Deployment

You now have a functioning cluster to deploy apps on. Your next objective is to deploy a simple Hello World web app on EKS.

Tasks:

1. Dockerize your Hello World web app.
2. Create a helm chart for it.
3. Deploy it via FluxCD.

Mission 5: Stress Testing

As with all applications deployed, we need to stress test it.

Tasks:

1. Create an account on https://www.artillery.io/
2. Run load test on your web app deployed on EKS

References:

https://www.artillery.io/