

PROJECT SPECIFICATION

Deploying High Availability Infrastructure

SLO/SLI

CRITERIA	MEETS SPECIFICATIONS
Create a SLO/SLI document	Students will use the provided <code>slo_sli_template.md</code> template to fill in the SLI column with a description of what the given SLO could use to achieve the objective. This will describe the indicator (SLI) that could be used to meet the objective.
Create an SLO/SLI dashboard for provided sample infrastructure	<ul style="list-style-type: none">Students will show 4 panels - availability (availability), remaining error budget (error budget), successful requests per second (throughput), and 90th percentile requests finish in this time (latency).Students will submit the Prometheus query. Each panel will contain a Prometheus query in the backend that demonstrates their ability to query Prometheus to get SLI metrics that can translate into an SLO. The dashboard itself is a demonstration of displaying the SLO in a meaningful way.

DR Document

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Define IT assets and ensure their structure is setup for resilience and HA	<p>Students will need to complete the following sections in the <code>dr_template.md</code> template:</p> <ul style="list-style-type: none">• AWS Zones• Servers and Clusters: Table 1.1 Summary• Descriptions
Define a DR plan using the asset list and infrastructure	<p>Students will list pre-steps for DR as well as high-level DR steps to fail this infrastructure over to an alternate region. Students will need to complete the following sections in the template:</p> <ul style="list-style-type: none">• DR Plan - > Pre-Steps• DR Plan -> Steps

Infrastructure deployment

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Create and apply infrastructure using infrastructure as code (Terraform) to demonstrate automation and consistency	<p>Modify the Terraform code in the starter code for the following for the <code>us-east-2</code> zone in AWS. Students will deploy the starter code, then make changes to meet the following criteria in the Terraform code:</p> <ul style="list-style-type: none">• 3 EC2 instances of the Ubuntu-Web VM• 2 EKS nodes for the udacity-cluster EKS cluster• The VPC has IPs in multiple availability zones• ALB (application load balancer) module was added to the modules folder under the zone1 folder. This will deploy a load balancer for the <code>Ubuntu-Web</code> VM group that they will deploy <p>The final submission will be the final code folder in Github and a screenshot of the final successful Terraform run deploying the code.</p>

CRITERIA	MEETS SPECIFICATIONS
Implement the DR plan using Terraform	<p>Modify the Terraform code in the starter code for the following for the <code>us-west-1</code> zone in AWS (zone2 folder). Students will deploy the starter code, then make changes to meet the following criteria in the Terraform code:</p> <ul style="list-style-type: none">• 3 EC2 instances of the Ubuntu-Web VM• 2 EKS nodes for the udacity-cluster EKS cluster• Multiple availability zones (students will need to add availability zones to VPC)• ALB module was added to the modules folder under the zone1 and zone2 folder. This will deploy a load balancer for the <code>Ubuntu-Web</code> VM group that students will deploy <p>The final submission will be the final code folder in Github and a screenshot of the final successful Terraform run deploying the code.</p>

CRITERIA	MEETS SPECIFICATIONS
Implement basic SQL replication and establish backups	<p>Modify the Terraform code in the starter code for the following for the <code>us-west-1</code> zone in AWS but note that this will be executed and changed from the zone1 folder. The zone1 folder is where you will do your modifications to deploy your RDS cluster to the other datacenter</p> <ul style="list-style-type: none"> • 2 RDS instance nodes • RDS backup window set to 5 days • RDS in zone2 will be set up as a replica from zone1. <p>Students will need to define this in the Terraform code</p> <p>The final submission will be the final code folder in Github and a screenshot of the final successful Terraform run deploying the code</p>
Destroy it all	<p>Students destroy all the Terraform resources in AWS using the given command for both <code>zone1</code> and <code>zone2</code></p> <p>Students will submit a screenshot of the final output from Terraform showing the destroyed resources</p>

Suggestions to Make Your Project Stand Out!

1. Perform a failover of their application load balancer to their secondary region using route 53 DNS
2. Failover the RDS instance to the secondary region so it becomes the primary target and the first region becomes the replica
3. Create an additional AWS module to provision another piece of infrastructure not discussed in the project