Deploy a high-availability web app using CloudFormation

learn.udacity.com/nanodegrees/nd9991/parts/cd12352/lessons/dd578403-7ae7-45e7-8d64-de0d631a6762/concepts/dd578403-7ae7-45e7-8d64-de0d631a6762-project-rubric

Project: Deploy a high-availability web app using CloudFormation

The Basics

Success Criteria	Specifications
Parameters	The more the better, but an exaggerated number of parameters can be messy (say, 10 or more). 1 or 0 is definitely lacking.
Infrastructure Diagram	An infrastructure diagram is present with the required network, server, and storage resources.
Security Groups	Security Groups follow the least privilege principle.
Resources	This is the mandatory section of the script, we are looking for a LoadBalancer, Launch Template, AutoScaling group a health check, security groups and a Listener and Target Group.
Outputs	Should have URL here with the Load Balancer DNS Name and “http” in front of it.
Working Test	If the student provides a URL to verify his work is running properly, it will be a page that says “it works! Udagram, Udacity”
Scripts	Scripts allow automation of the creation and deletion of infrastructure.

Load Balancer

Success Criteria	Specifications

Success Criteria	Specifications
Target Group	The auto-scaling group needs to have a property that associates it with a target group. The Load Balancer will have a Listener rule associated with the same target group
Health Check and Listener	Port 80 should be used in Security groups, health checks and listeners associated with the load balancer

Auto-Scaling

Success Criteria	Specifications
Subnets	Students should be using PRIV-NET (private subnets) for their auto-scaling instances
Machine Specs	The machine should have 10 GB or more of disk and should be a t3.small or better. The LaunchTemplate configuration should match the server's requirements.
Importing Network IDs from Network Stack	Network output values are used as imports in the Udagram stack.

Success Criteria	Specifications
Output	Any values in the output section are a bonus
Bastion Host	Any resource of type AWS::EC2::Instance, optional, but nice to have.

Suggestions to Make Your Project Stand Out

-Students can deploy Windows Servers instead of Linux and use PowerShell scripts to showcase their Windows management skills. -Students can use AWS Parameter Store to save sensitive data, such as credentials to showcase their attention to security. -Students can use CloudWatch Alarms and CloudWatch custom metrics to showcase their performance and monitoring skills.