Test Project Day three

Cloud Computing

Submitted by: Taze Miller US

Contents

This Test Project consists of the following documentation/files:

- WS2019_TP53_Main_document
- WSC2019_TP53_DayOne_actual
- WSC2019_TP53_DayTwo_actual
- **WSC2019_TP53_DayThree_actual**
- WSC2019_TP53_DayFour_actual

Description of project and tasks

Today's test project is six hours.

The goal this test project is further test specific cloud computing skills through a series of unique modules. There will be 4 tasks for you to complete in any order that you choose.

Tasks

1. Log into the AWS Provided testing platform using the same credentials you've used previously
2. Read the documentation thoroughly (Outlined below)
3. Continue until the test project day has completed (Six hours)

AWS Jam - High Availability

This challenge introduces the student to the concept of chaos testing by providing an opportunity to re-architect a critical web service to be highly-available.

Summary

Re-Architect a critical web service for High Availability

Background

You are a new operations engineer working for a major financial services company, responsible for running a critical web service for your organization.

The service is publically accessible, read-only, and the architecture consists of an ELB, a Python app running on a single EC2 instance in us-east-2b,and a MySQL RDS instance, all inside a VPC.

The webapp connects to the DB via an internal Route53 record. DNS is managed by an external team who CNAME the ELB, and clients point to a '/data' endpoint behind that DNS name.

This service is critical to the business and must be resilient to a major network outage or misconfiguration (must continue to serve data without manual intervention)

Inventory

- VPC w/ 4 subnets (2 public/2 private)
- MySQL RDS Instanc
- EC2 Instance running web service
- EC2 Instance running validator application
- ELB
- Route53 Private Zone

Challenge

Your supervisor has asked you to identify and remediate any single-points-of-failure so that the service is resilient to a major network outage or misconfiguration.

The automated testing team has deployed a validator app on EC2 to test your architecture, you can access it by browsing to /validate. This will simulate a network issue that isolates all resources running in us-east-2b while attempting to pull data from the /data. If it passes, you have passed the challenge. Good luck!

AWS Jam - Perfect World

Your customer runs a hotel company that stores PCI data. They want to lock down their PCI environment to prevent all SSH logins.

Take the provided script and implement a method that will automatically quarantine a server if it is logged into.

The SSH key has been provided for your usage with this challenge.

The answer to this challenge can be found if you browse to the public IP of the EC2 instance in your browser (http).

Details

- The login script can be found at: https://s3-us-west-2.amazonaws.com/aws-jam-challenge-resources/perfectworld/init-script.sh
- The EC2 instance notifies SQS of a login
- The SQS triggers the Lambda on receipt of the message

Hint

Did you put the script on the EC2, but still no solution? Is there also a problem in the lambda function? Maybe a problem updating an attribute.

AWS Jam - Too Many Secrets:

Rumor has it that one of your developers was sloppy and may have stored some "secrets" in the source code of a project they were working on. Find it before your entire VPC is mining crypto! Oh yeah... they also quit and took their ssh key with them.

Summary

Find the secrets that were left behind.

A bit of background

The developer quit without passing on any details or documentation about what he did. Although we have the instance he was developing on, nobody knows what happened to the key. We suspect that there may be credentials placed inside the source code.

Inventory

An EC2 instance

Challenge

The answer is the 20 uppercase alphanumeric characters that you should be worrying about.

AWS Jam - How to Automate Incident Response and Detection

This challenge revolves around how to build automated response systems for incidents at the infrastructure layer.

An EC2 instance has been found to be in communication with a known command and control server.

The findings can be seen at the Amazon GuardDuty console. As a security engineer, you have been instructed to automate the following steps to respond to such incidents:

Take a snapshot of the volume.

Quarantine the instance in its own Security Group.

You discover that the Lambda function which isolates the instance into its own Security Group is buggy. Your task is to fix this Lambda function.

Upon completing the challenge, a tag of ForensicsCase is added to the EC2 instance, and the answer to this challenge is the value of the tag.

Please refer to the **WS2019_TP53_Main_document** document for more details.