Building a Highly Available, Scalable Web Application

Phase 1: Planning the design and estimating cost (day 1)

In this phase, I will plan the design of the architecture. First, I will create an architecture diagram.
Next, I will estimate the cost of the proposed solution, and present the estimate. An important first step for any solution is to plan the design and estimate the cost. As necessary, review the various components in the architecture to adjust the estimated cost. Cost is an important factor when building a solution because cost can help to determine the components and architecture pattern to use.

Task 1.1: Creating an architectural diagram

Create an architectural diagram to illustrate what you plan to build.

Deliverables: Architecture diagram

Task 1.2: Developing a cost estimate

Develop a cost estimate that shows the cost to run the solution in the us-east-1 Region for 12 months.

Deliverables: Cost assessment report

Phase 2: Creating a basic functional web application (day 2)

In this phase, I will start to build the solution. The objective of this phase is to have a functional web application that works on a single virtual machine in a virtual network. By the end of this phase, I will have a POC to demonstrate hosting the application on the AWS Cloud.

Task 2.1: Creating a virtual network

Create a virtual network to host the web application.

Deliverables: Configuration documentation of deployed VPC

Task 2.2: Creating a virtual machine

Create a virtual machine in the cloud to host the web application.

Deliverables: Configuration documentation of deployed EC2s

Task 2.3: Testing the deployment

Test the deployment of the web application to ensure it is accessible from the internet and functional. Perform a few tasks, such as viewing, adding, deleting, or modifying records.

Deliverables: Configuration documentation of the website

Phase 3: Decoupling the application components (day 3)

In this phase, I will continue building. The objective is to separate the database and the web server infrastructure so that they run independently. The web application should run on a separate virtual machine, and the database should run on the managed service infrastructure.

Task 3.1: Changing the VPC configuration

Update or re-create the virtual network components that are necessary to support hosting the database separately from the application.

Deliverables: Configuration documentation of deployed VPC

Task 3.2: Creating and configuring the Amazon RDS database

Create an Amazon Relational Database Service (Amazon RDS) database that runs a MySQL engine.

Deliverables: Configuration documentation of deployed RDS

Task 3.3: Configuring the development environment

Provision an AWS Cloud9 environment to run AWS Command Line Interface (AWS CLI) commands in later tasks.

Deliverables: Configuration documentation of deployed Cloud9

Task 3.4: Provisioning Secrets Manager

Use AWS Secrets Manager to create a secret to store the database credentials, and configure the web application to use Secrets Manager.

Deliverables: Configuration documentation of deployed Secret manager

Task 3.5: Provisioning a new instance for the web server

Create a new virtual machine to host the web application.

Deliverables: Configuration documentation of deployed Web site

Task 3.6: Migrating the database

Migrate the data from the original database, which is on an EC2 instance, to the new Amazon RDS database.

Deliverables: Configuration documentation of deployed database

Task 3.7: Testing the application

Access the application and perform a few tasks to test it. For example, view, add, delete, and modify student records.

Deliverables: Configuration documentation of deployed Web site

Phase 4: Implementing high availability and scalability (day 4)

In this phase, I will complete the design and fulfill the remaining solution requirements. The objective is to use the key components that I created in earlier phases to build a scalable and highly available architecture.

Task 4.1: Creating an Application Load Balancer

Launch a load balancer. The endpoint will be used to access your web application.

Deliverables: Configuration documentation of deployed Load balancer

Task 4.2: Implementing Amazon EC2 Auto Scaling

Create a new launch template, and use an Auto Scaling group to launch the EC2 instances that host the web application.

Deliverables: Configuration documentation of deployed Autoscaling group

Task 4.3: Accessing the application

Access the application and perform a few tasks to test it. For example, view, add, delete, and modify student records.

Deliverables: Configuration documentation of deployed Web site

Task 4.4: Load testing the application

Perform a load test on the application to monitor scaling.

Deliverables: report containing test outcomes

Securing and Monitoring Resources with AWS

Phase 1: Securing data in Amazon S3 (day 1)

In this phase, I will be implementing security settings in the AWS account. I have been asked to secure customer PII data that is stored in Amazon S3. The leadership team of AnyCompany Financial has heard about recent data breaches at other companies and wants to protect customer data from unauthorized access. The company wants to do the following:

- Limit access to buckets to certain account managers, who are in the Account Manager group.
- Enable versioning on S3 buckets and all objects in them.

- Enable object logging in all S3 buckets.
- Encrypt all buckets by using server-side encryption with Amazon S3 managed keys (SSE-S3).
- Implement Amazon S3 Inventory to keep a running inventory of all files that are stored in Amazon S3.

Task 1.1: Create a bucket, apply a bucket policy, and test access

In this task, I will create a bucket, apply a bucket policy to it, and then test whether Paulo and Mary can access the bucket.

Deliverables: Configuration documentation of deployed Buckets

Task 1.2: Enable versioning and object-level logging on a bucket

In this task, I will enable versioning and object-level logging on the data-bucket. With versioning enabled, you can track all changes to objects that are stored in the bucket and revert any object to a previous version if needed. Object-level logging creates a detailed audit trail of the objects that are stored in a bucket, which helps you to detect security incidents quickly.

Deliverables: Configuration documentation of deployed S3

Task 1.3: Implement the S3 Inventory feature on a bucket

In this task, I will enable the S3 Inventory feature to monitor changes to objects that are stored in an S3 bucket. S3 Inventory provides a scheduled report of the metadata and object-level changes to your S3 objects and buckets. By using the feature, you can track changes to the stored objects and detect potential security incidents.

Deliverables: Configuration documentation of deployed S3

Task 1.4: Confirm that versioning works as intended

In this task, I will access the AWS account as the paulo user and upload an object to the data-bucket. Then, I will confirm that versioning is enabled on the object. I also test access as the mary user. In the next task, I will

analyze the object-level logs to see the actions that I took as different users in this task.

Deliverables: Configuration documentation of deployed Bucket

Task 1.5: Confirm object-level logging and query the access logs by using Athena

In this task, I will confirm the S3 object-level logging I enabled earlier is successfully writing log data to S3. I will also use Athena to query these logs.

Deliverables: Configuration documentation of deployed S3

Task 1.6: Cost assessment

I will calculate the cost assessment to secure Amazon S3

Deliverables: Cost assessment report

Phase 2: Securing VPCs (day 2,3)

After securing the data in Amazon S3, the leadership team for AnyCompany Financial wants me to focus on securing the network in the AWS Cloud that hosts the company's critical applications. They are aware of recent network security incidents and want to ensure that their network is protected from unauthorized access and attacks. My task is to secure the virtual private clouds (VPCs) for the company's web servers.

An inexperienced employee of AnyCompany Financial created the LabVPC and the WebServer instance that pre-exist in the lab project environment. The employee made some mistakes and the result is that the network is not properly configured. In tasks 2.1 through 2.4 I will analyze the existing configurations and make updates to correct the network configuration

Task 2.1: Review LabVPC and its associated resources

In this task, I will review the resources that already exist in the lab environment.

Deliverables: Configuration documentation of deployed VPC

Task 2.2: Create a VPC flow log

In this task, I will create a VPC flow log for LabVPC. The VPC Flow Logs feature can help you understand how inbound and outbound traffic flows through the VPC. The feature also provides monitoring information that will provide insights for how to secure the web server, subnets, and VPC in later tasks.

Deliverables: Configuration documentation of deployed VPC flow log

Task 2.3: Access the WebServer instance from the internet and review VPC flow logs in CloudWatch

In this task, I will use the web browser to test access to the WebServer EC2 instance over port 80 (HTTP). I also will test access to port 22 (SSH) by using the netcat command, which will test whether inbound traffic is allowed. Then, I will review the VPC flow log to see how these attempts were recorded.

Deliverables: Configuration documentation of deployed CloudWatch

Task 2.4: Configure route table and security group settings

In this task, I will create a route for traffic from the internet to access the WebServerSubnet through an internet gateway. This will allow inbound HTTP traffic to be directed to the WebServer instance. I will also modify the security group that is associated with the WebServer instance to allow inbound traffic on ports 22 (SSH) and 80 (HTTP).

Deliverables: Configuration documentation of route tables

Task 2.5: Secure the WebServerSubnet with a network ACL

In this task, I will configure a network access control list (ACL) to secure the subnet where the web server is running. The network ACL will provide an additional layer of security beyond the security group that I already configured.

Deliverables: Configuration documentation of deployed ACLs

Task 2.6: Review NetworkFirewallVPC and its associated resources

In this phase so far, I worked to secure LabVPC by updating a route table, network ACL, and security group.

In the remaining tasks in this phase, I will work to secure a different VPC, named NetworkFirewallVPC. I will secure the VPC by using AWS Network Firewall. AWS Network Firewall features provide us another tool for securing our network. In this project I will use it to achieve a result similar to how I secured LabVPC using security groups and NACLs.

Deliverables: Configuration documentation of deployed NetworkFirewallVPC

Task 2.7: Create a network firewall

In this task, I will create a network firewall for the NetworkFirewallVPC, which hasn't yet used in this project.

Deliverables: Configuration documentation of deployed firewall settings

Task 2.8: Create route tables

In this task, I will create and configure three new route tables, including one for each subnet in the NetworkFirewallVPC and one to handle inbound (ingress) traffic for the internet gateway in NetworkFirewallVPC.

Deliverables: Configuration documentation of deployed route tables

Task 2.9: Configure logging for the network firewall

In this task, I will configure logging for the network firewall so that we can analyze the details of network traffic requests.

Deliverables: Configuration documentation of deployed firewall settings

Task 2.10: Configure the firewall policy and test access

In this task, I will define and add a stateful rule group to the network firewall's policy. With this policy, traffic to and from the internet and the NetworkFirewallVPC will be monitored and managed. Access over specific ports will be allowed, and access over other ports will be specifically denied.

Deliverables: Configuration documentation of deployed firewall policy settings

Task 2.11: Cost assessment

I will calculate the cost estimate to secure a VPC with a network firewall

Deliverables: Cost assessment report

Phase 3: Securing AWS resources by using AWS KMS (day 4)

The legal department at AnyCompany Financial received notice from the U.S. Federal Deposit Insurance Corporation (FDIC) that the company needs to encrypt sensitive information, such as PII, credit card numbers, and social security numbers. The legal department contacted my manager, the director of IT, and they tasked me to implement encryption and tokenization to meet regulatory compliance standards.

Task 3.1: Create a customer managed key and configure key rotation

In this task, I will create an AWS KMS customer managed key. I will then configure automatic key rotation on the key. In later tasks, I will use this key to protect data that is stored in the account.

Deliverables: Configuration documentation of deployed KMS settings

Task 3.2: Update the AWS KMS key policy and analyze an IAM policy

In this task, I will modify the policy of the AWS KMS key that I created so that the sofia user will be authorized to use the key. I will analyze the IAM policy that controls what the sofia user can do in the AWS account.

Deliverables: Configuration documentation of deployed IAM policy settings

Task 3.3: Use AWS KMS to encrypt data in Amazon S3

In this task, I will use the AWS KMS key that I created to encrypt an object in the data-bucket S3 bucket. I will then test access to the object.

Deliverables: Configuration documentation of deployed S3 settings

Task 3.4: Use AWS KMS to encrypt the root volume of an EC2 instance

In this task, I will use the AWS KMS key again, but now I will use it to encrypt the root volume of a new EC2 instance.

Deliverables: Configuration documentation of deployed root volume settings

Task 3.5: Use AWS KMS envelope encryption to encrypt data in place

In this task, I will use the AWS Command Line Interface (AWS CLI) to encrypt data in place by using the AWS KMS key. I will see how to decrypt the encrypted data.

Deliverables: Configuration documentation of deployed KMS envelope settings

Task 3.6: Use AWS KMS to encrypt a Secrets Manager secret

In this task, I will create a key-value pair (a secret), which I will encrypt with my AWS KMS key and store in Secrets Manager. I will verify that I can retrieve the secret by using the AWS CLI.

Deliverables: Configuration documentation of deployed Secrets Manager settings

Task 3.7: Cost assessment

I will calculate the cost assessment for using AWS KMS

Deliverables: Cost assessment report

Phase 4: Monitoring and logging (day 5)

The leadership team at AnyCompany Financial received reports of a new security breach at one of its biggest competitors. The company wants to ensure that it's prepared to detect and respond to any future security incidents. The director of IT has asked me to implement a monitoring and

logging solution to detect security incidents so that the company can respond promptly.
The solution needs to do the following:

- Track all API calls to S3 buckets.
- Monitor application logs.
- Notify team members in case of security incidents.
- Monitor AWS resource configurations and automatically modify configurations that are out of compliance.

Task 4.1: Use CloudTrail to record Amazon S3 API calls

In this task, I will use CloudTrail to record API calls that are made to Amazon S3 buckets. This information will provide an audit trail to track when S3 objects are created, modified, or read.

Deliverables: Configuration documentation of deployed CloudTrail settings

Task 4.2: Use CloudWatch Logs to monitor secure logs

AnyCompany Financial has been working with a vendor to design and manage their website. The vendor has requested SSH access to design and manage the website. Company policy doesn't allow direct access through SSH to the production web servers, but access can be provided to a development web server named EncryptedInstance.
I have been asked to create a solution to monitor access to EncryptedInstance. In this task, I will configure CloudWatch Logs to monitor SSH access to the instance so that your company can understand who accesses the server, where they access it from, when they access it, and what actions they take.

Deliverables: Configuration documentation of deployed CloudWatch settings

Task 4.3: Create a CloudWatch alarm to send notifications for security incidents

AnyCompany Financial wants security team members in IT to be notified when attempts to access the EncryptedInstance through SSH are denied. In this task, I will create a CloudWatch alarm to notify these team members when such an incident occurs.

Deliverables: Configuration documentation of deployed CloudWatch alarm settings

Task 4.4: Configure AWS Config to assess security settings and remediate the configuration of AWS resources

AnyCompany Financial would like to have a mechanism to automatically configure AWS resources based on company standards. One standard is to have object logging enabled on S3 buckets.

In this task, I will use AWS Config to report on whether object logging is configured on the S3 buckets in the AnyCompany Financial account. I will also configure an automation script to remediate noncompliance.

Deliverables: Configuration documentation of deployed AWS Config settings

Task 4.5: Cost assessment

I will calculate the cost assessment for monitoring and logging

Deliverables: Cost assessment report