Cloud Dev Lab 3

Lab Objectives

By the end of this lab, you will have built a Role Based Access Controlled (RBAC) system, and understood the need of authorization and authentication. Scenario for this lab is: In a factory, there are workers and management. The workers can view their tasks (and add/delete them) and management assigns “ToDo” tasks to the workers and can modify them.

Note: be very careful when provisioning resources in this lab, cross check your service configurations with what is available in free tier, specially RDS

Proposed Services

- ECS Fargate
- Cognito
- AWS RDS
- Cloudfront
- S3 (Simple Storage Service)
- CloudWatch

Region: us-east-1

IaaC Tool: Terraform.

Expected Workflow

Step 1: Develop a Simple To-Do Application

Start by creating a basic to-do list application using the tech stack of your choice. The application should allow users to:

- Add new to-do items
- Delete existing to-do items
- View their list of tasks

Step 2: Implement Authentication & Authorization with AWS Cognito

Set up AWS Cognito to manage user authentication and authorization. Create two user groups:

1. Simple Users – Can only **view, add,** and **delete** their **own** tasks.
2. Admins – Can **view, update,** and **delete** tasks from **all** users.

When users log in, use Cognito's authentication tokens to determine:

- Who the logged-in user is
- What actions they are authorized to perform

Unauthorized actions should be strictly denied at both the frontend and backend levels.

Step 3: Store To-Do Items in AWS RDS

Use AWS RDS as the database to store to-do items, ensuring that each item is linked to a specific user. Admins should have access to all records, while regular users should only see their own tasks.

Step 4: Deploy the Backend on AWS ECS Fargate

Containerize the backend application and deploy it on AWS ECS Fargate, ensuring it is scalable and secure. Use IAM roles and security groups to control access.

Step 5: Deploy the Frontend on AWS S3 and CloudFront

- Host the frontend as a static website on AWS S3
- Use CloudFront for content delivery to improve performance and security

Submission Guidelines:

You are required to submit:

1) Terraform and Code files In a zip folder. Remember to follow proper file naming convention for terraform files. And from last lab’s experience, remember to delete all unnecessary files created.
2) An architecture diagram of the entire process. Please incorporate feedback from lab 1 and project diagrams shared.
3) A database schema diagram of the entities you are storing.
4) Working screenshots of the deployed application.
5) Explanations of what “RBAC” is and its importance in web applications.

Add these in a pdf and rename that as **<YourRollNumber>_Lab3.pdf** and submit on the LMS assignments tab.

Note: For any topic, refer to Amazon's or Terraform documentation for detail, examples are shown also in class. Remember to look at service pricing and limits. Be sure to add relevant tags to all of the services that you use for this lab. For marking, create 2 simple users and 1 admin, login from them and demonstrate the specified actions listed in workflow.

Good luck and start early