

CAMPUSBITES: AWS ARCHITECTURE DESIGN & IMPLEMENTATION REPORT

Project Team:

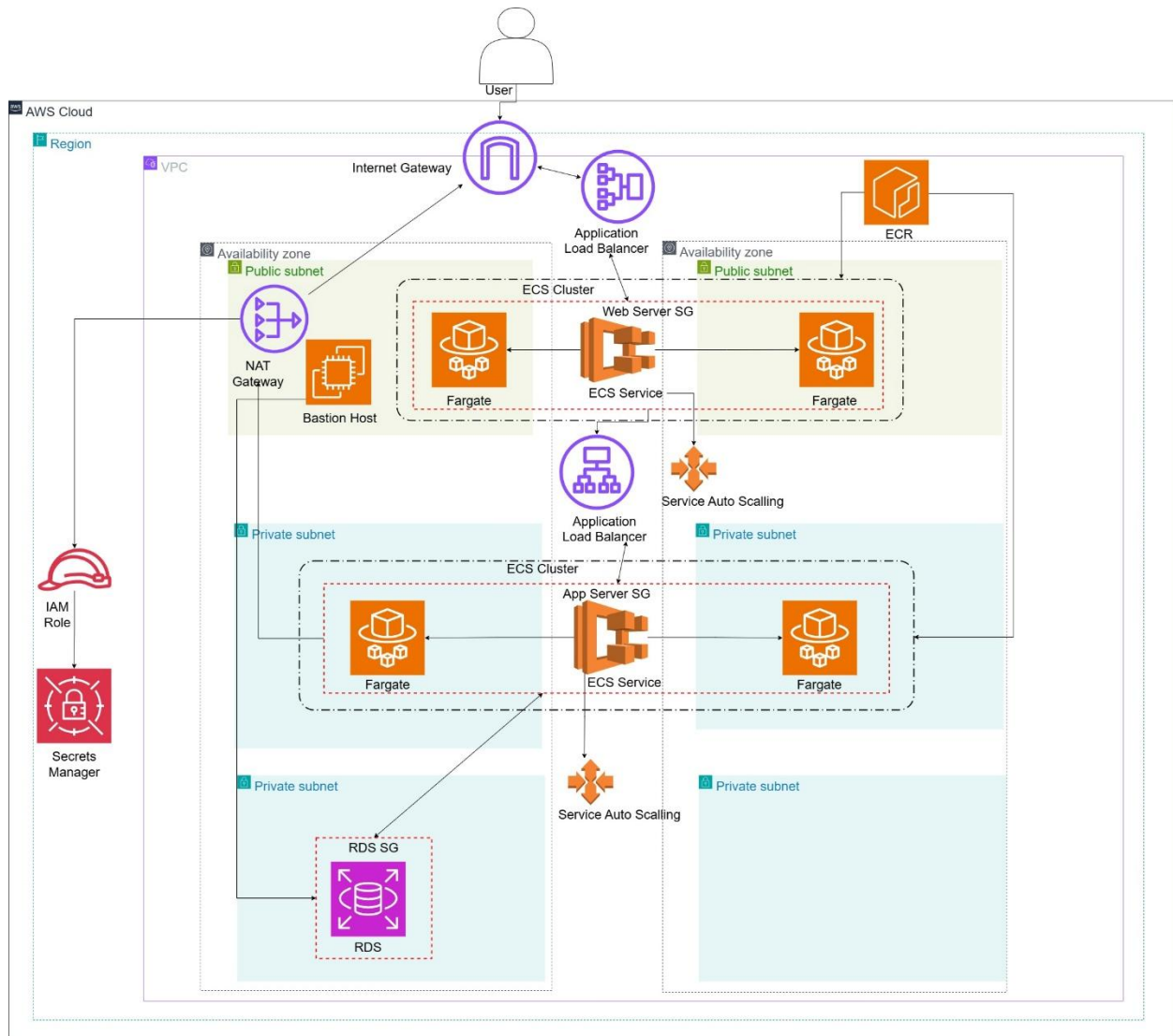
- Muhammad Samer Nisar (BSSE23113)
- Ahmad Umar Khan (BSSE23008)
- Mirza Muhammad Rehan (BSSE23021)

1. AWS ARCHITECTURE DIAGRAM

The CampusBites system utilizes a **3-Tier Serverless Architecture** designed for High Availability (Multi-AZ) and Zero-Trust Security.

1.1 Diagram Components

- **Networking:** VPC with 6 subnets (2 Public, 2 Private App, 2 Private Data).
- **Load Balancing:** Dual Application Load Balancers (External-facing for users, Internal-facing for inter-service API calls).
- **Compute:** ECS Cluster utilizing AWS Fargate (Serverless) for React.js and Node.js containers.
- **Storage/Data:** Amazon RDS (PostgreSQL) and Amazon ECR for container image versioning.
- **Security:** Bastion Host (Jump Box), AWS Secrets Manager, and IAM Role-based access.



2. STEP-BY-STEP IMPLEMENTATION

PHASE 1: FOUNDATIONAL NETWORKING (VPC & SUBNETS)

Goal: Establish an isolated environment with Multi-AZ redundancy.

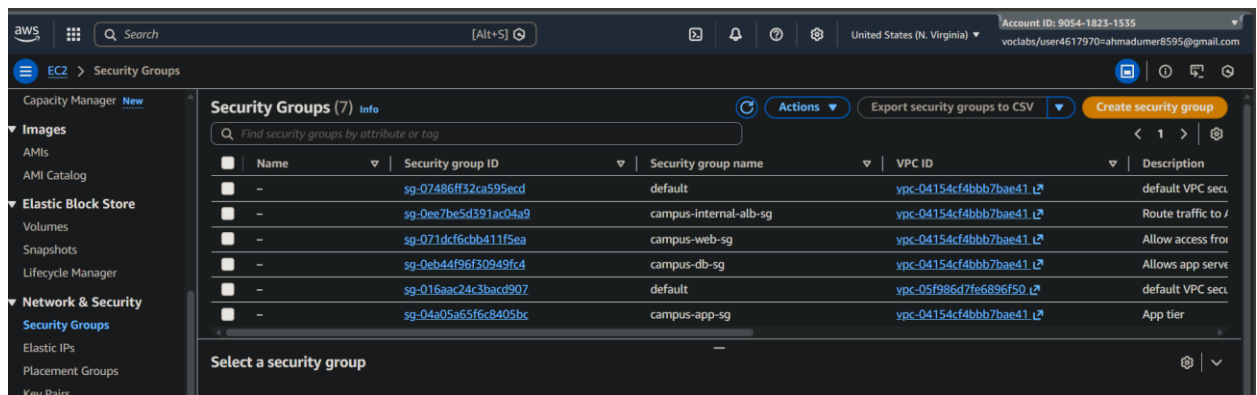
1. **VPC Creation:** Created a VPC with CIDR 10.0.0.0/16.
2. **Subnet Allocation:**
 - **Public A/B:** 10.0.1.0/24 & 10.0.2.0/24 (Hosts ALBs, NAT GW, Bastion).
 - **Private App A/B:** 10.0.3.0/24 & 10.0.4.0/24 (Hosts Backend Tasks).
 - **Private Data A/B:** 10.0.5.0/24 & 10.0.6.0/24 (Hosts PostgreSQL RDS).
3. **Gateways:** Attached an **Internet Gateway (IGW)** for public access and a **NAT Gateway** in Public Subnet A to allow private tasks to reach the internet for updates.

[SCREENSHOT: VPC Dashboard showing IPv4 CIDR and 6 Subnets] [SCREENSHOT: Route Tables showing Public (IGW) and Private (NAT-GW) routes]

PHASE 2: SECURITY & IAM (LEAST PRIVILEGE)

Goal: Implement granular access controls.

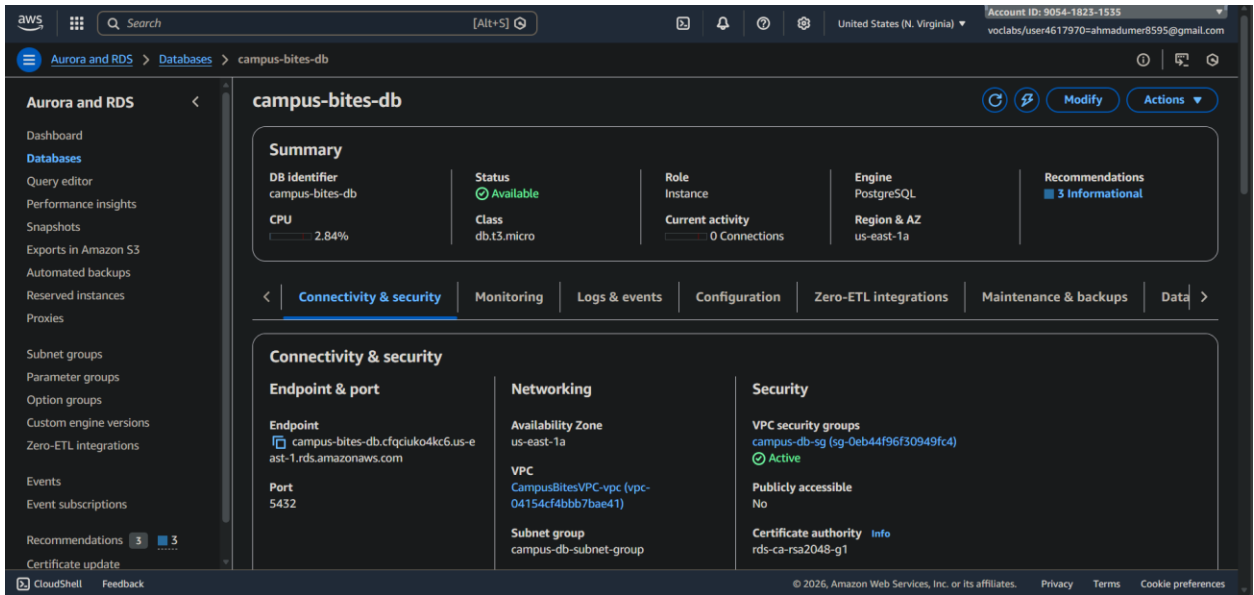
1. **IAM Task Execution Role:** Created a role with AmazonECSTaskExecutionRolePolicy and inline policies to read from **AWS Secrets Manager**.
2. **Security Group Referencing Chain:**
 - External-ALB-SG: Allow 80/443 from 0.0.0.0/0.
 - Web-Task-SG: Allow Port 80 from External-ALB-SG.
 - Internal-ALB-SG: Allow Port 80 from Web-Task-SG.
 - App-Task-SG: Allow Port 8080 from Internal-ALB-SG.
 - RDS-SG: Allow Port 5432 from App-Task-SG and Bastion-SG.



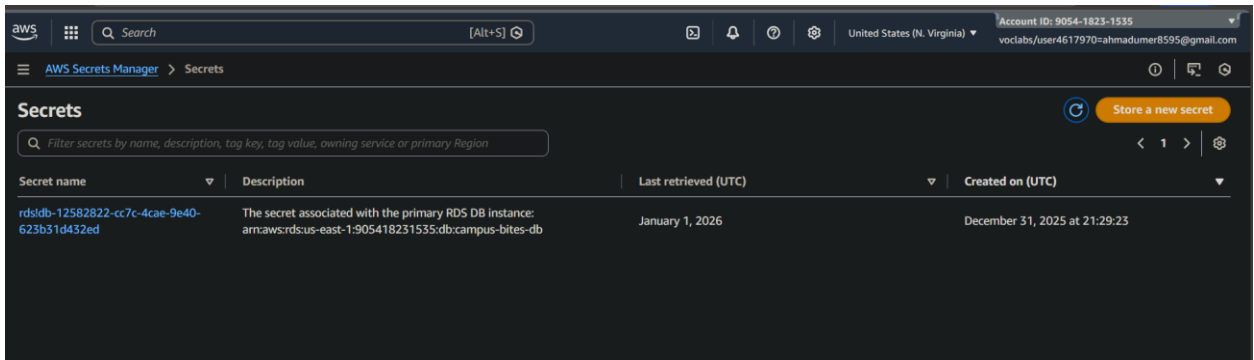
PHASE 3: DATABASE & SECRETS MANAGEMENT

Goal: Secure data persistence.

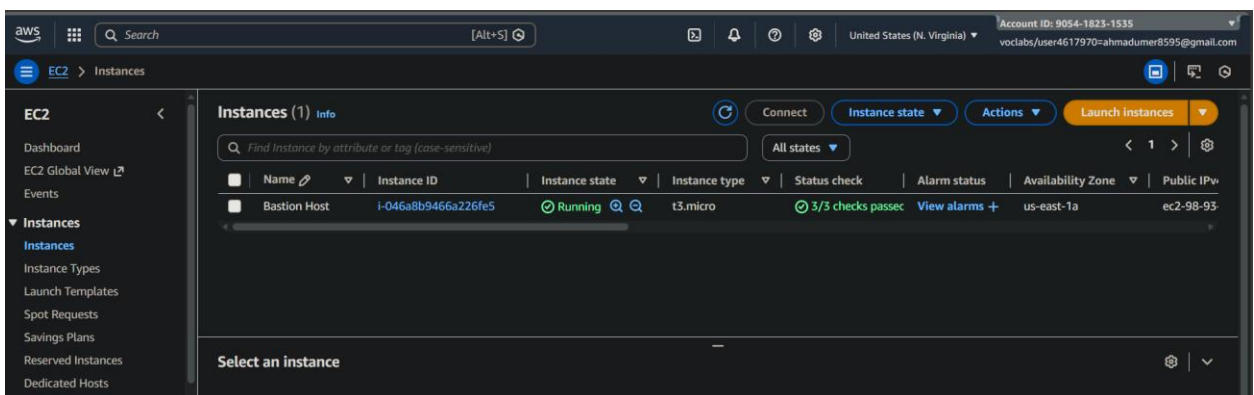
1. **RDS Provisioning:** Deployed **Amazon RDS (PostgreSQL)** in the Private Data Subnet Group.



2. **Secrets Manager:** Created a secret campus-bites/db-creds containing the master username, password, and endpoint. This prevents plaintext credentials in our code.



3. **Bastion Host:** Launched a t3.micro EC2 instance in the Public Subnet to perform initial schema migrations.



[SCREENSHOT: RDS Instance Status 'Available' in Private Subnet] [SCREENSHOT: Secrets Manager Secret Value overview]

PHASE 4: CONTAINER ORCHESTRATION (ECS & FARGATE)

Goal: Scalable compute without server management.

1. **ECR Repositories:** Created campus-bites-web and campus-bites-app repositories.
2. **Task Definitions:**
 - Configured **Fargate** tasks (0.25 vCPU for Web, 0.5 vCPU for App).
 - Mapped environment variables to Secrets Manager keys.
3. **ECS Services:**
 - **Web Service:** Connected to the External ALB.
 - **App Service:** Connected to the Internal ALB.

The image displays two screenshots from the AWS Management Console. The top screenshot shows the Amazon ECR Private registry page with two repositories: 'campus-bites-app' and 'campus-bites-web'. The bottom screenshot shows the Amazon ECS Cluster overview for a cluster named 'likable-rabbit-jmruh8', which has two services: 'campus-app-service' and 'campus-web-service'.

Amazon ECR Private repositories (2)

Repository name	URI	Created at	Tag immutability	Encryption type
campus-bites-app	905418231535.dkr.ecr.us-east-1.amazonaws.com/campus-bites-app	December 14, 2025, 22:33:13 (UTC+05)	Mutable	AES-256
campus-bites-web	905418231535.dkr.ecr.us-east-1.amazonaws.com/campus-bites-web	December 19, 2025, 16:47:15 (UTC+05)	Mutable	AES-256

Amazon ECS Cluster overview

ARN: arn:aws:ecs:us-east-1:905418231535:cluster/likable-rabbit-jmruh8

Status: **Active**

CloudWatch monitoring: **Default**

Registered container instances: **-**

Services

Draining	Active	Pending	Running
-	2	-	2

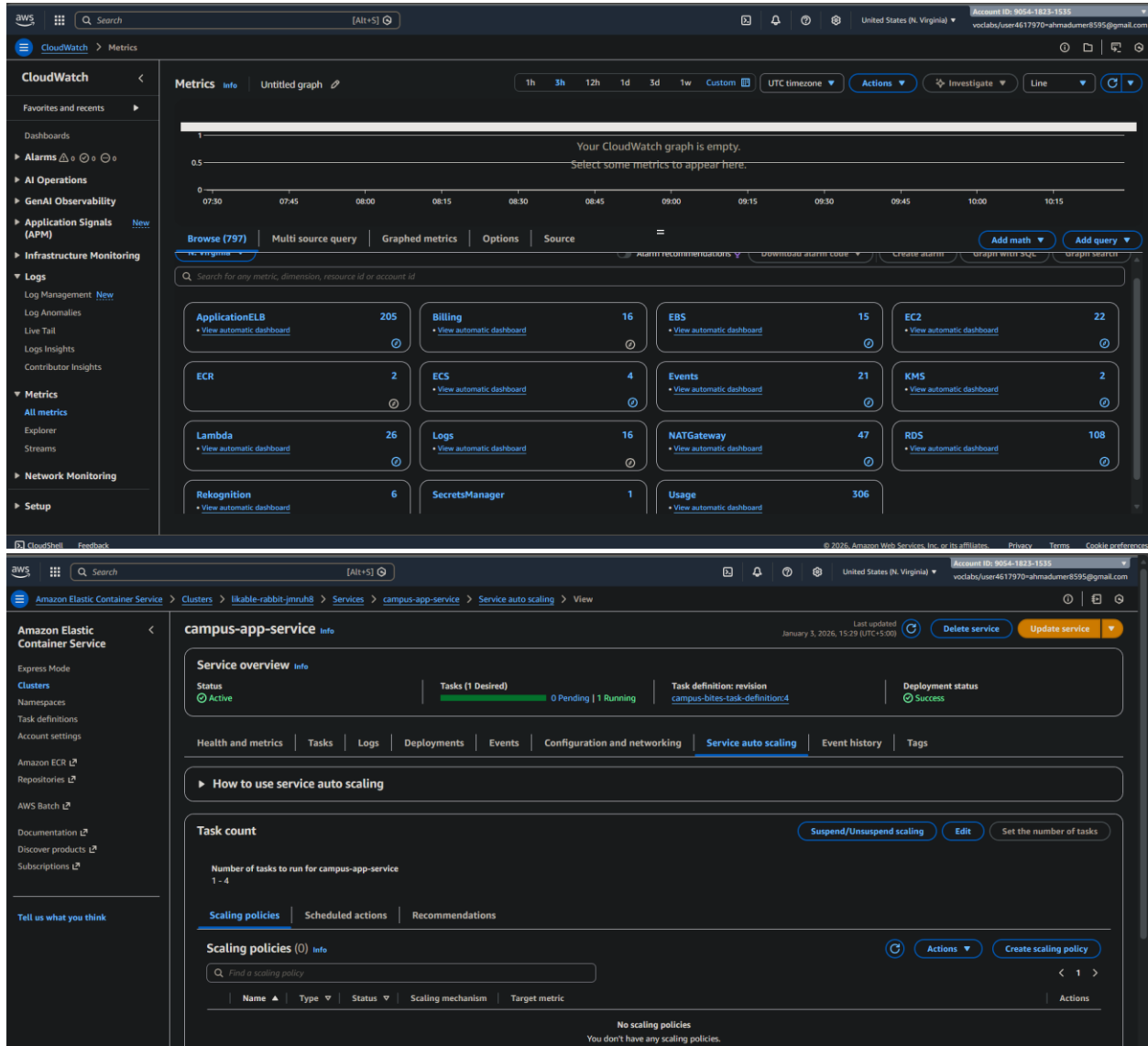
Services (2)

Service name	ARN	Status	Scheduling strategy	Launch type	Task definition	Deployments and tasks
campus-app-service	arn:aws:ecs:us-east-1:905418231535:task-definition/campus-app-service	Active	REPLICA	FARGATE	campus-bites-app	1/1 Task
campus-web-service	arn:aws:ecs:us-east-1:905418231535:task-definition/campus-web-service	Active	REPLICA	FARGATE	campus-bites-web	1/1 Task

PHASE 5: MONITORING & LOGGING (CLOUDWATCH)

Goal: Observability and automated scaling.

1. **Log Groups:** Configured awslogs driver in Task Definitions to stream application logs to CloudWatch.
2. **Auto Scaling:** Set up a Target Tracking Policy to scale the App Service when average CPU utilization exceeds **70%**.



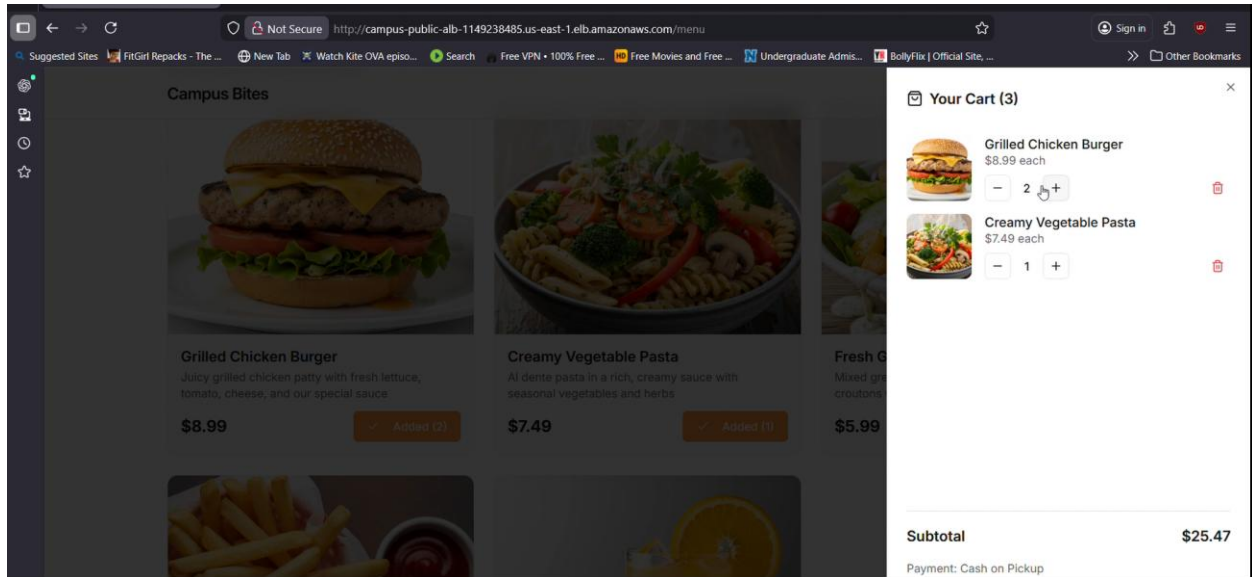
3. TESTING WORKFLOW & FINAL OUTPUT

3.1 Connectivity Test

- **Result:** Pinging the External ALB DNS name successfully loads the React frontend.
- **Internal Routing:** Frontend successfully fetches menu data from the Internal ALB endpoint.

3.2 Security Validation

- **Result:** Attempting to access the RDS instance from the public internet results in a timeout (Success: VPC isolation verified).
- **Result:** Access only possible via the Bastion Host SSH Tunnel.



4. SECURITY & COMPLIANCE CHECKLIST

- **Least Privilege:** IAM roles used instead of Root/Admin keys.
- **Encryption:** RDS storage encrypted at rest (AES-256).
- **Isolation:** All compute and data tiers reside in Private Subnets.
- **Credential Security:** No passwords hardcoded; all retrieved via Secrets Manager.