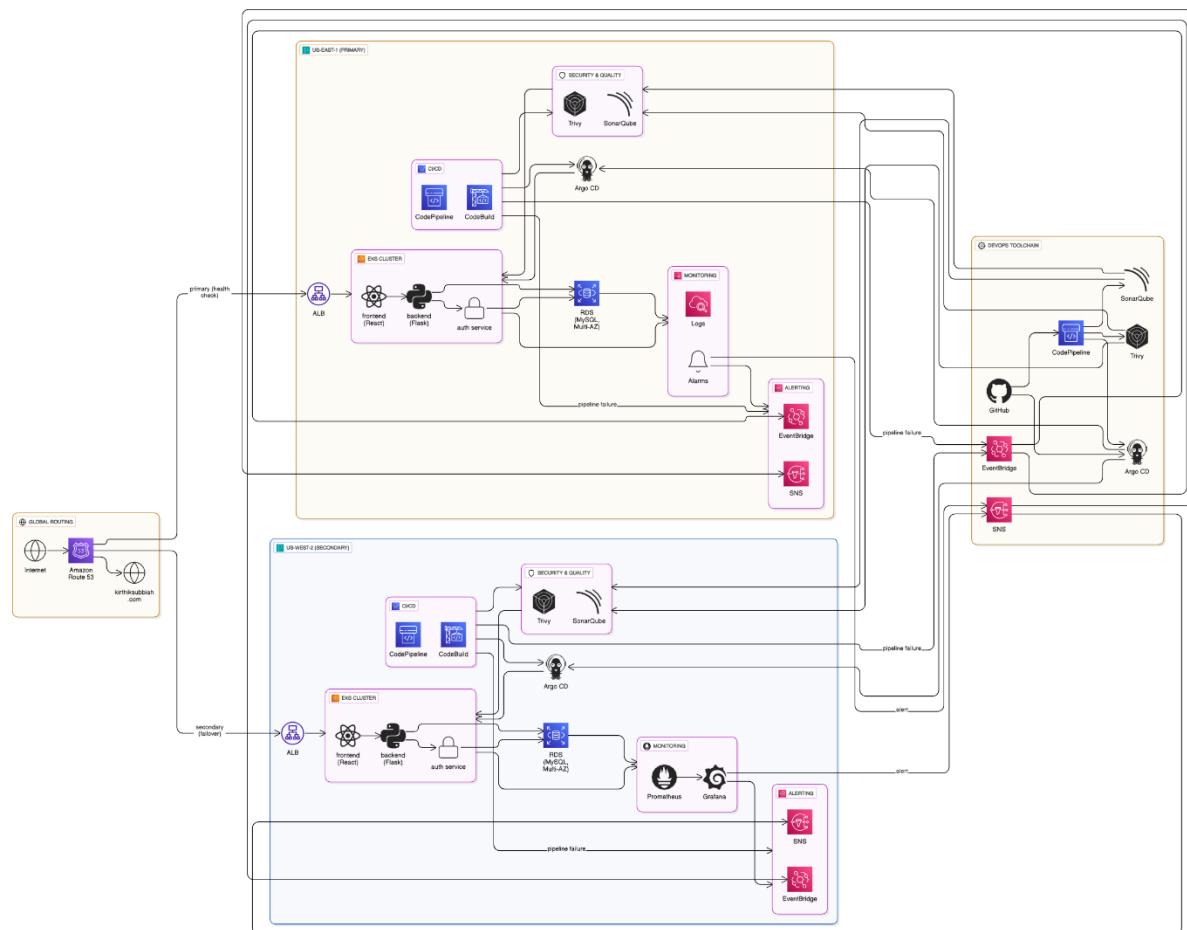


E-commerce:

AWS Multi-Region E-Commerce Microservices Deployment A full-stack, production-grade e-commerce platform deployed across multiple AWS regions. Built with Python Flask (backend), ReactJS (frontend), and powered by Amazon EKS, this solution is fully automated using CloudFormation, Terraform, and a complete CI/CD pipeline (CodePipeline, CodeBuild). It ensures high availability, auto-healing, and DNS failover with Route 53, all while following AWS best practices for security, scalability, and observability.

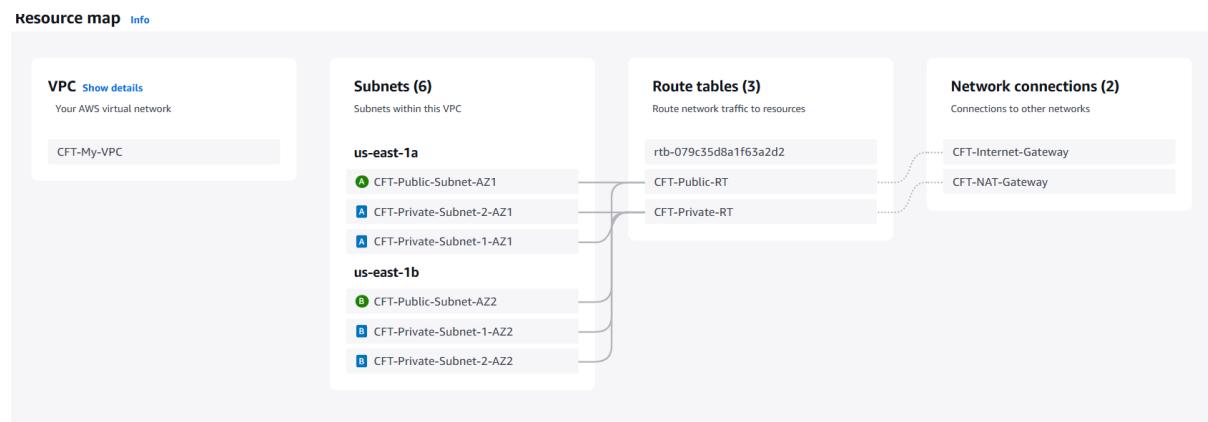
Architecture Diagram:



Project goals:

- Deploy a Containerized E-Commerce Application on AWS EKS
Build and deploy application as Docker container on an Amazon EKS cluster, ensuring scalability and resilience.
- Implement Multi-Region Infrastructure Automation
 - Use AWS CloudFormation to provision infrastructure resources in Region A.
 - Use Terraform to automate deployment of infrastructure components in Region B, enabling a multi-region architecture.
- Establish CI/CD Pipeline with AWS DevOps Services
Integrate AWS CodePipeline and CodeBuild to automate the build, test, and deployment processes for continuous integration and delivery.
- Enable Route 53 Multi-Region Failover
Configure Amazon Route 53 for DNS-based health checks and failover routing policies to ensure high availability and disaster recovery across regions.
- Integrate Monitoring and Logging with Amazon CloudWatch
Set up CloudWatch to collect logs, metrics, and alarms for real-time monitoring and alerting of application and infrastructure health.

Network:



 **Application Stack Overview**

Tier	Technology / Tool Used
Frontend	React
Backend	Flask (Python)
Auth Service	Flask (Microservice)
Database	Amazon RDS (MySQL - Multi-AZ)
Orchestration	Amazon EKS (Kubernetes)
Load Balancer	Application Load Balancer (ALB)
DNS Routing	Amazon Route 53 with Failover Routing
CI/CD	AWS CodePipeline, CodeBuild
Security	SonarQube (Code Quality), Trivy (Image Scan)
Monitoring	CloudWatch (Primary), Prometheus + Grafana (Secondary)

PHASES:

Phase 1: Application Layer Deployment

- **Tools Used:** Amazon EKS, Docker

Phase 2: CI/CD Automation

- **Tools Used:** AWS CodePipeline, AWS CodeBuild
- **Description:** Implements a fully automated CI/CD pipeline to build Docker images, push them to ECR, and deploy updates to EKS clusters in both regions seamlessly.

Phase 3: Infrastructure Provisioning

Region A – US East (N. Virginia)

- **Tools Used:** AWS CloudFormation
- **Description:** Provisions core infrastructure resources such as **VPC**, **EKS cluster**, **Subnets**, and **Security Groups** using declarative templates.

Region B – US West (Oregon)

- **Tools Used:** Terraform
- **Description:** Replicates the infrastructure setup in a second region to enable multi-region availability and redundancy.

Phase 4: DNS-Based Regional Failover

- **Tools Used:** Amazon Route 53
- **Description:** Configures health checks and sets up DNS failover routing to automatically switch traffic between regions in case of a service outage.

Phase 5: Monitoring and Notifications

- **Tools Used:** Amazon CloudWatch, Amazon SNS

- **Description:** Collects performance metrics and logs from the infrastructure and application. Integrates with SNS to send alerts and notifications for anomalies or failures.

Integration Across Phases:

Tool	Region	Purpose
ArgoCD	us-west-2	GitOps + Health monitoring
CloudWatch	us-east-1	Logs + Metrics + Alarms
Prometheus	us-west-2	Real-time pod/node monitoring
Grafana	us-west-2	Visual dashboards
SonarQube	Central (EC2)	Code quality and static analysis
Trivy	CodeBuild step	Vulnerability scanning

Project Description

E-Commerce Capstone Project is a full-stack, multi-region cloud-native solution designed to streamline digital retail operations. It provides intuitive interfaces for customers and administrators, supporting tasks like user management, product catalog browsing, cart operations, and secure order processing. With a scalable backend and GitOps-powered infrastructure, the system ensures high availability and resilience through its deployment across multiple AWS regions using Amazon EKS and RDS.

Features

- **Customer Management:** Register, login, update profiles, and view order history
- **Product Catalog:** Browse available items with descriptions and prices
- **Shopping Cart:** Add, remove, and update products in the cart
- **Order Management:** Place and track customer orders

- **Admin Panel:** Create and manage products, view users, and monitor activity
- **User Authentication:** Flask-Login-based secure user sessions
- **Role-Based Access Control:** Admin/user-based view rendering
- **CSRF-Protected Forms:** Secure form interactions using Flask-WTF
- **ArgoCD Integration:** GitOps deployment with health/status monitoring
- **CI/CD Automation:** Docker image builds and Kubernetes deployments using CodePipeline

Technologies Used

Layer	Technology
Backend	Python (Flask, SQLAlchemy, Flask-WTF)
Frontend	HTML, CSS, Bootstrap (via Jinja templates)
Database	Amazon RDS – MySQL (Multi-AZ)
DevOps Tools	Docker, Amazon ECR, CodeBuild, CodePipeline
GitOps	ArgoCD for automated app sync and health
Kubernetes	Amazon EKS
DNS Failover	Route 53 (primary/secondary routing)

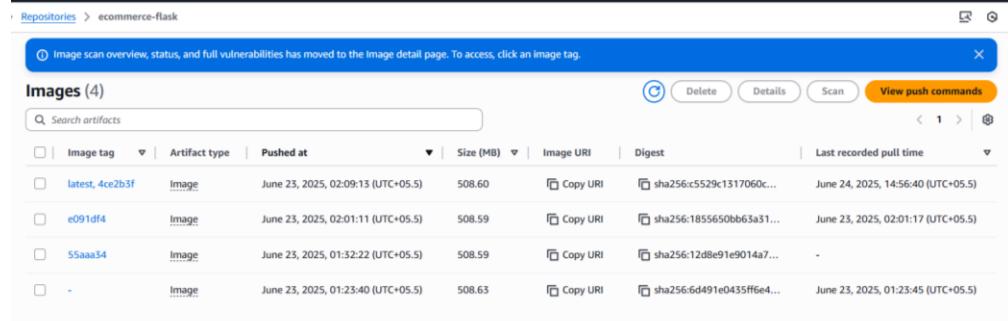
Development Tools

- **IDE:** VS Code / PyCharm
- **CI/CD:** AWS CodePipeline + CodeBuild
- **Version Control:** Git + GitHub
- **Infrastructure as Code:** Terraform (us-west-2), CloudFormation (us-east-1)

PHASE 1: Application Layer Deployment on EKS

Step 1: Prepare Docker Image for Backend

Step 2: Create Kubernetes Deployment and Service YAML

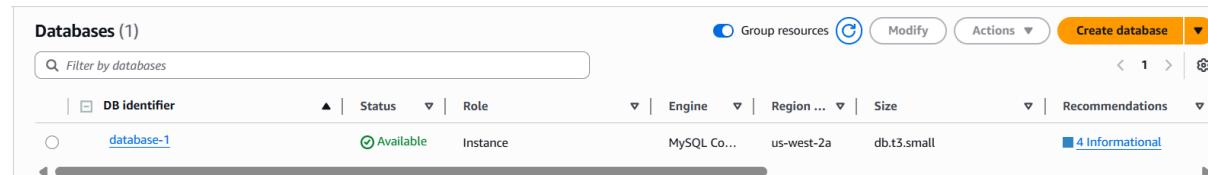


The screenshot shows the AWS Lambda console's 'Images' section. It lists four images with the following details:

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
latest, 4ce2b3f	Image	June 23, 2025, 02:09:13 (UTC+05.5)	508.60	Copy URI	sha256:c5529c1317060c...	June 24, 2025, 14:56:40 (UTC+05.5)
e091df4	Image	June 23, 2025, 02:01:11 (UTC+05.5)	508.59	Copy URI	sha256:1855650bb63a31...	June 23, 2025, 02:01:17 (UTC+05.5)
55aaa34	Image	June 23, 2025, 01:52:22 (UTC+05.5)	508.59	Copy URI	sha256:12d8e91e9014a7...	-
-	Image	June 23, 2025, 01:23:40 (UTC+05.5)	508.63	Copy URI	sha256:6d491e0435ff6e4...	June 23, 2025, 01:23:45 (UTC+05.5)

Step 3: Create an EKS cluster

Step 4: RDS database



The screenshot shows the AWS RDS console's 'Databases' section. It lists one database instance named 'database-1' with the following details:

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations
database-1	Available	Instance	MySQL Co...	us-west-2a	db.t3.small	4 Informational

Step 5: Deploy application

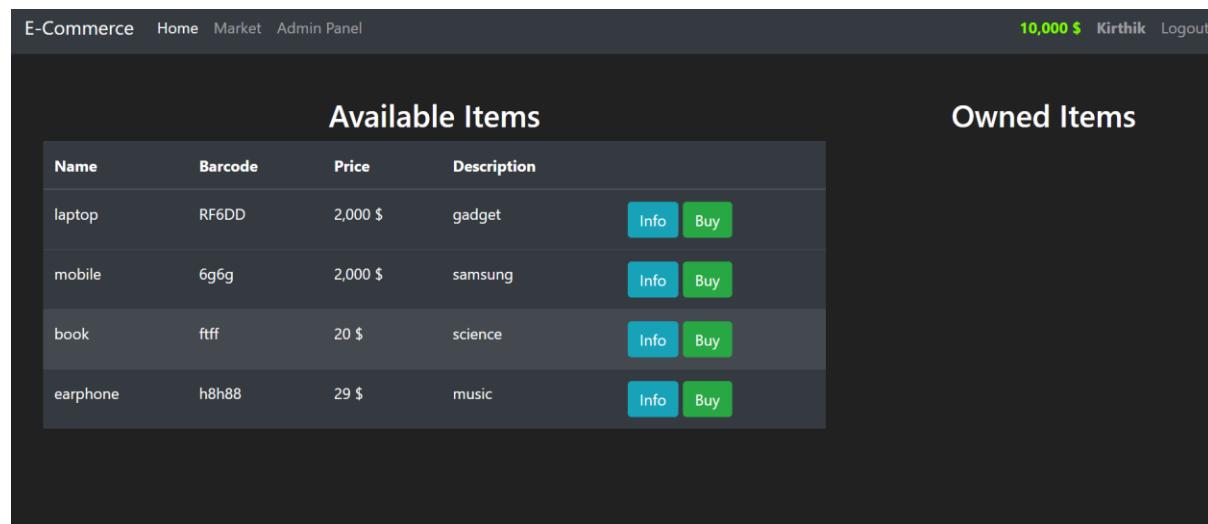
Step 6: Expose via loadBalancer

```
root@ip-172-31-18-138:/home/ubuntu# kubectl get all -o wide
NAME           READY   STATUS    RESTARTS   AGE   IP          NODE      NOMINATED-NODE   READINESS   GATES
pod/commerce-app-674c98bd4-5hav9   1/1    Running   0          107m  10.0.25.210  ip-10-0-31-156.ec2.internal  <none>        <none>
pod/commerce-app-674c98bd4-hxtp7   1/1    Running   0          107m  10.0.64.161  ip-10-0-64-77.ec2.internal  <none>        <none>

NAME            TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE   SELECTOR
service/commerce-service   LoadBalancer  172.20.70.47  a0153b01e42524244983cf04d5bee796-1254871159.us-east-1.elb.amazonaws.com  80:32539/TCP  107m  app=commerce
service/kubernetes       ClusterIP    172.20.0.1    <none>          443/TCP   2d    <none>

NAME           READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES
deployment.apps/commerce-app   2/2     2           2          107m  ecommerce   026090546401.dkr.ecr.us-east-1.amazonaws.com/commerce-flask:latest
replicaset.apps/commerce-app-64ddfd846c  2       2           2          0     ecommerce   026090546401.dkr.ecr.us-east-1.amazonaws.com/commerce-flask:latest
replicaset.apps/commerce-app-674c98bd4   2       2           2          107m  ecommerce   026090546401.dkr.ecr.us-east-1.amazonaws.com/commerce-flask:latest
```

Step 7: verify the app working through loadbalancer



The screenshot shows an E-commerce website interface. At the top, there is a navigation bar with links for 'E-Commerce', 'Home', 'Market', and 'Admin Panel'. On the right side of the header, it shows '10,000 \$' and the user 'Kirthik' with a 'Logout' link.

The main content area has two sections: 'Available Items' and 'Owned Items'.

Available Items:

Name	Barcode	Price	Description	Info	Buy
laptop	RF6DD	2,000 \$	gadget	Info	Buy
mobile	6g6g	2,000 \$	samsung	Info	Buy
book	ftff	20 \$	science	Info	Buy
earphone	h8h88	29 \$	music	Info	Buy

Owned Items:

(This section is currently empty, indicated by a large black box.)

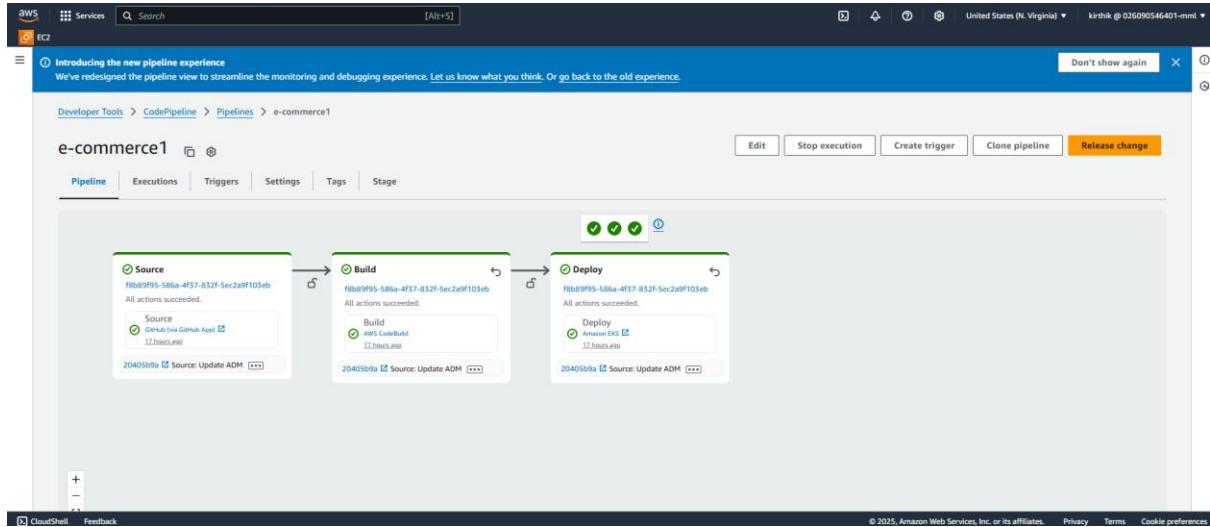
PHASE 2: AWS Core Pipeline Automation

Step 1: Create ECR Repository

Step 2: Define buildspec.yml in root of repo

Step 3: Create CodeBuild Project

Step 4: Create CodePipeline



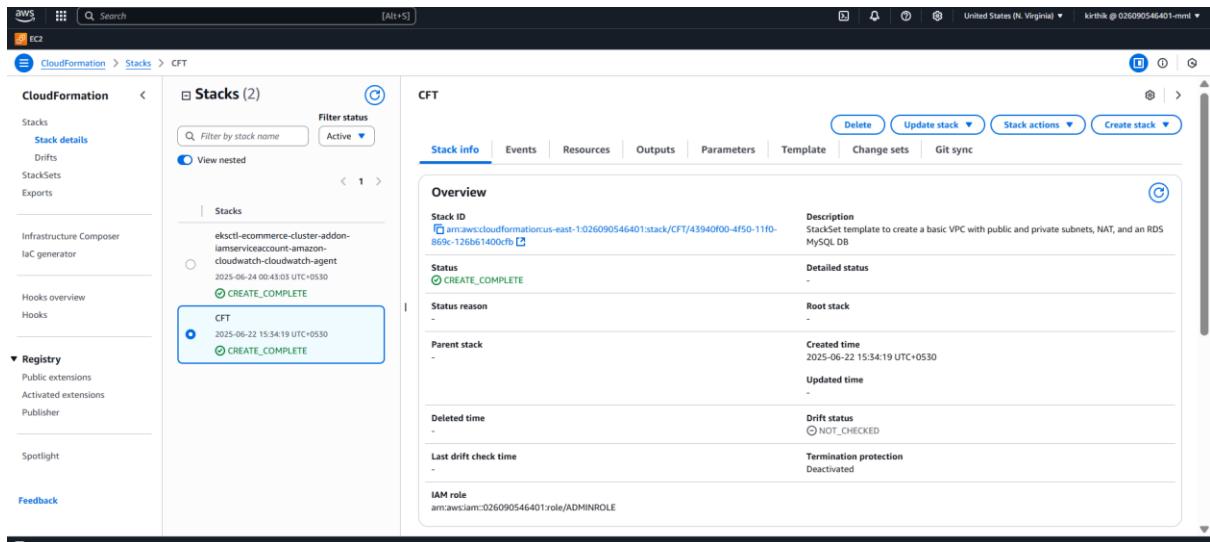
Step 5: Auto Deploy to EKS

PHASE 3: Multi-Region Infrastructure Deployment

1. CloudFormation in another (e.g., us-east-1)

2. Terraform in one region (e.g., us-west-2)

Step 1: CloudFormation Deployment in Region A



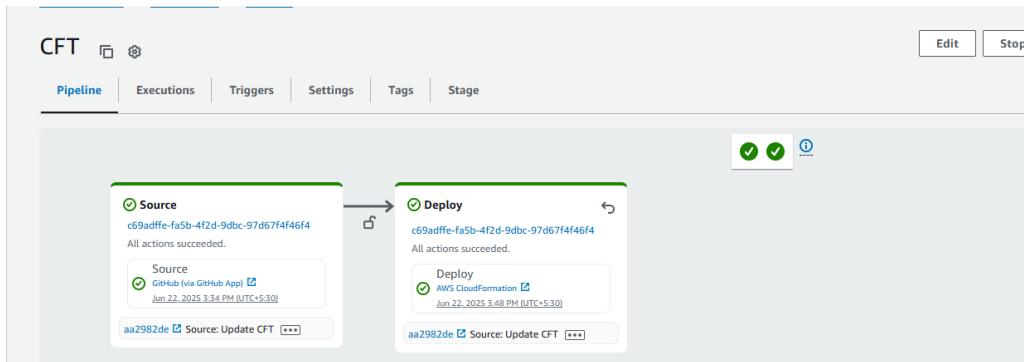
Step 2: Terraform Deployment in Region B

```
aws_eks_node_group.my_node_group: Still creating... [1m0s elapsed]
aws_db_instance.mysql_database: Still creating... [8m10s elapsed]
aws_eks_node_group.my_node_group: Still creating... [1m10s elapsed]
aws_db_instance.mysql_database: Still creating... [8m20s elapsed]
aws_eks_node_group.my_node_group: Still creating... [1m20s elapsed]
aws_db_instance.mysql_database: Still creating... [8m30s elapsed]
aws_eks_node_group.my_node_group: Still creating... [1m30s elapsed]
aws_db_instance.mysql_database: Still creating... [8m40s elapsed]
aws_eks_node_group.my_node_group: Still creating... [1m40s elapsed]
aws_eks_node_group.my_node_group: Creation complete after 1m47s [id=eks-cluster-2:terraform-2025062218211310590000000d]
aws_db_instance.mysql_database: Still creating... [8m50s elapsed]
aws_db_instance.mysql_database: Still creating... [9m0s elapsed]
aws_db_instance.mysql_database: Still creating... [9m10s elapsed]
aws_db_instance.mysql_database: Still creating... [9m20s elapsed]
aws_db_instance.mysql_database: Still creating... [9m30s elapsed]
aws_db_instance.mysql_database: Still creating... [9m40s elapsed]
aws_db_instance.mysql_database: Still creating... [9m50s elapsed]
aws_db_instance.mysql_database: Still creating... [10m0s elapsed]
aws_db_instance.mysql_database: Still creating... [10m10s elapsed]
aws_db_instance.mysql_database: Still creating... [10m20s elapsed]
aws_db_instance.mysql_database: Still creating... [10m30s elapsed]
aws_db_instance.mysql_database: Still creating... [10m40s elapsed]
aws_db_instance.mysql_database: Creation complete after 10m42s [id=db-6WCVCS3GE7MPDF4KE5354477YH]
```

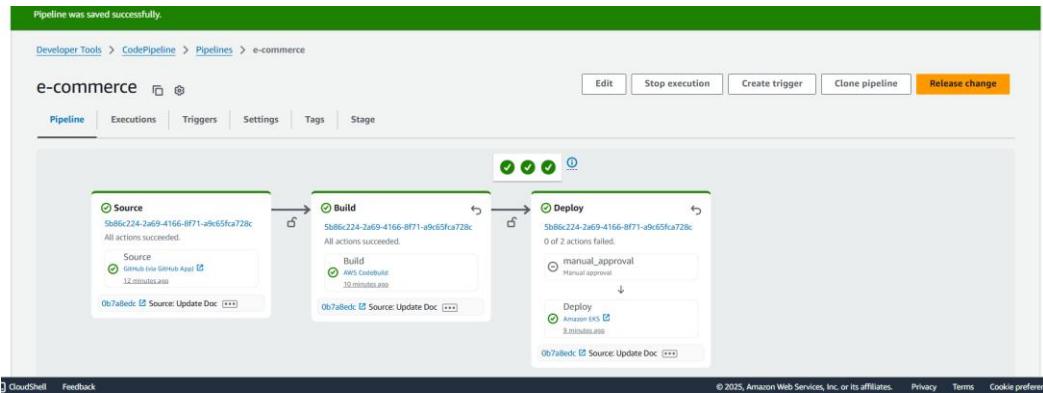
Apply complete! Resources: 37 added, 0 changed, 0 destroyed.

Step 3: CodePipeline

1. For cloudformation region



2. For terraform region



Trivy:

2106	Python (python-pkg)						
2107	<hr/>						
2108	Total: 3 (HIGH: 3, CRITICAL: 0)						
2109							
2110							
2111							
2112	Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
2113	setup tools (METADATA)	CVE-2022-40897	HIGH	fixed	58.1.0	65.5.1	pypa-setuptools: Regular Expression Denial of Service (ReDoS) in package_index.py https://avd.aquasec.com/nvd/cve-2022-40897
2114		CVE-2024-6345				70.0.0	pypa/setuptools: Remote code execution via download functions in the package_index module in... https://avd.aquasec.com/nvd/cve-2024-6345
2115		CVE-2025-47273				78.1.1	setup tools: Path Traversal Vulnerability in setuptools PackageIndex https://avd.aquasec.com/nvd/cve-2025-47273
2116							
2117							
2118							
2119							
2120							
2121							
2122							
2123							
2124							
2125							
2126							
2127	[Container] 2025/06/24 10:08:27.774390 Running command echo "Updating Kubernetes deployment..."						

Total: 178 (HIGH: 169, CRITICAL: 9)

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
icu-devtools	CVE-2025-5222	HIGH	affected	72.1-3		icu: Stack buffer overflow in the SRBRoot::addTag function https://avd.aquasec.com/nvd/cve-2025-5222
libaom3	CVE-2023-6879	CRITICAL	will_not_fix	3.6.0-1+deb12u1		aom: heap-buffer-overFlow on frame size change https://avd.aquasec.com/nvd/cve-2023-6879
	CVE-2023-39616	HIGH				ADMedia v3.0.0 to v3.5.0 was discovered to contain an invalid read mem... https://avd.aquasec.com/nvd/cve-2023-39616
libbluetooth-dev	CVE-2023-44431	fix_deferred		5.66-1+deb12u2		bluez: AVRCP stack-based buffer overflow remote code execution vulnerability https://avd.aquasec.com/nvd/cve-2023-44431
	CVE-2023-51596					bluez: phone book access profile heap-based buffer overflow remote code execution vulnerability... https://avd.aquasec.com/nvd/cve-2023-51596
libbluetooth3	CVE-2023-44431					bluez: AVRCP stack-based buffer overflow remote code execution vulnerability https://avd.aquasec.com/nvd/cve-2023-44431
	CVE-2023-51596					bluez: phone book access profile heap-based buffer overflow remote code execution vulnerability... https://avd.aquasec.com/nvd/cve-2023-51596
libc-bin	CVE-2025-4802	affected		2.36-9+deb12u10		glibc: static setuid binary dlopen may incorrectly search LD_LIBRARY_PATH https://avd.aquasec.com/nvd/cve-2025-4802
libc-dev-bin						
libc6						
libc6-dev						
libexpat1	CVE-2023-52425		will_not_fix	2.5.0-1+deb12u1		expat: parsing large tokens can trigger a denial of service https://avd.aquasec.com/nvd/cve-2023-52425
	CVE-2024-8176					libexpat: expat: Improper Restriction of XML Entity Expansion Depth in libexpat https://avd.aquasec.com/nvd/cve-2024-8176
libexpat1-dev	CVE-2023-52425	affected				expat: parsing large tokens can trigger a denial of service https://avd.aquasec.com/nvd/cve-2023-52425
	CVE-2024-8176					libexpat: expat: Improper Restriction of XML Entity Expansion Depth in libexpat https://avd.aquasec.com/nvd/cve-2024-8176
libharfbuzz0b	CVE-2023-25193	affected		6.0.0+dfsg-3		harfbuzz: allows attackers to trigger O(n^2) growth via consecutive marks https://avd.aquasec.com/nvd/cve-2023-25193
libicu-dev	CVE-2025-5222			72.1-3		icu: Stack buffer overflow in the SRBRoot::addTag function https://avd.aquasec.com/nvd/cve-2025-5222
	libicu72					
libldap-2.5-0	CVE-2023-2953			2.5.13+dfsg-5		openldap: null pointer dereference in ber_malloc_x function https://avd.aquasec.com/nvd/cve-2023-2953

Argo CD for deployment:

The screenshot shows the Argo CD interface for managing the 'ecommerce-app'. On the left, a sidebar provides navigation links for Applications, Settings, User Info, and Documentation. It also includes resource filters for NAME, KINDS, and SYNC STATUS, with a count of 2 Synced resources. The main panel displays the 'Applications' section for 'ecommerce-app'. Key metrics shown include APP HEALTH (Healthy), SYNC STATUS (Synced to HEAD 1fdebc8), and LAST SYNC (Sync OK to 1fdebc8). A detailed sync history table is present, showing a successful sync from 'Create main.tf' at 15:12:14 on June 24, 2025. Below this is a deployment graph illustrating the flow from the source code repository through various stages (ecommerce-service, ecommerce-app, ecommerce-app-674c98bdc4..., pod) to the final running pods.

PHASE 4: Setup Route 53 Failover

1. Go to AWS Route 53 → Hosted Zones

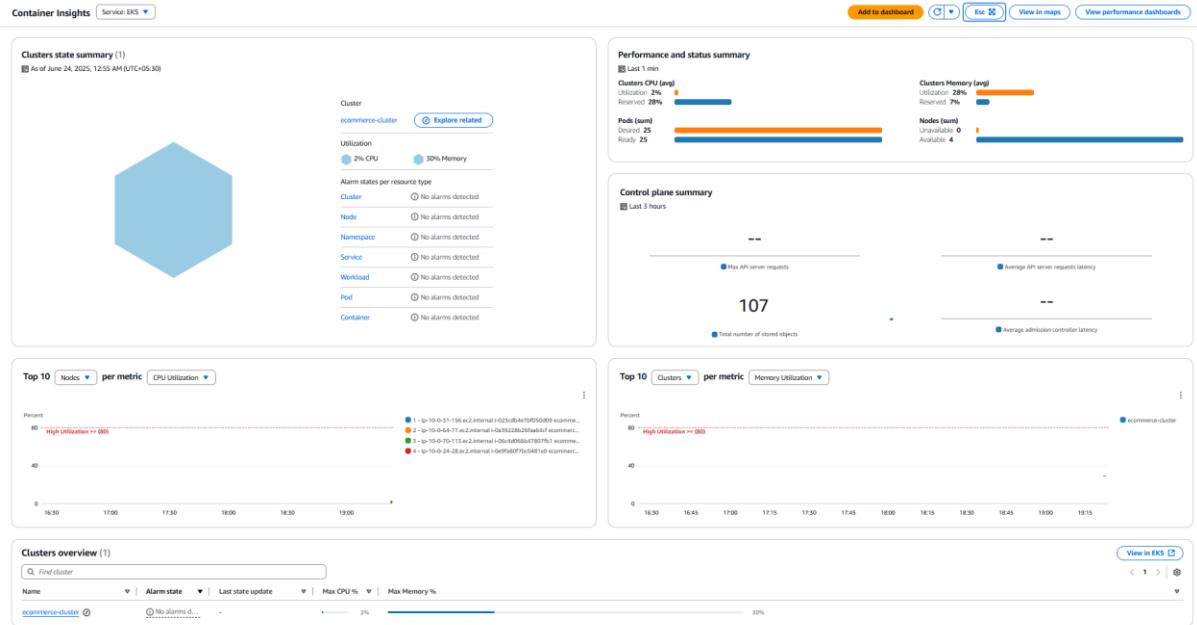
The screenshot shows the 'Hosted zones' page in the AWS Route 53 console. It lists one hosted zone named 'kirthiksubbiah.com', which is Public and was created by 'Route 53'. The page includes a search bar, a table with columns for Hosted zone name, Type, Created by, and Record count, and a 'Create hosted zone' button.

2. Create 2 records

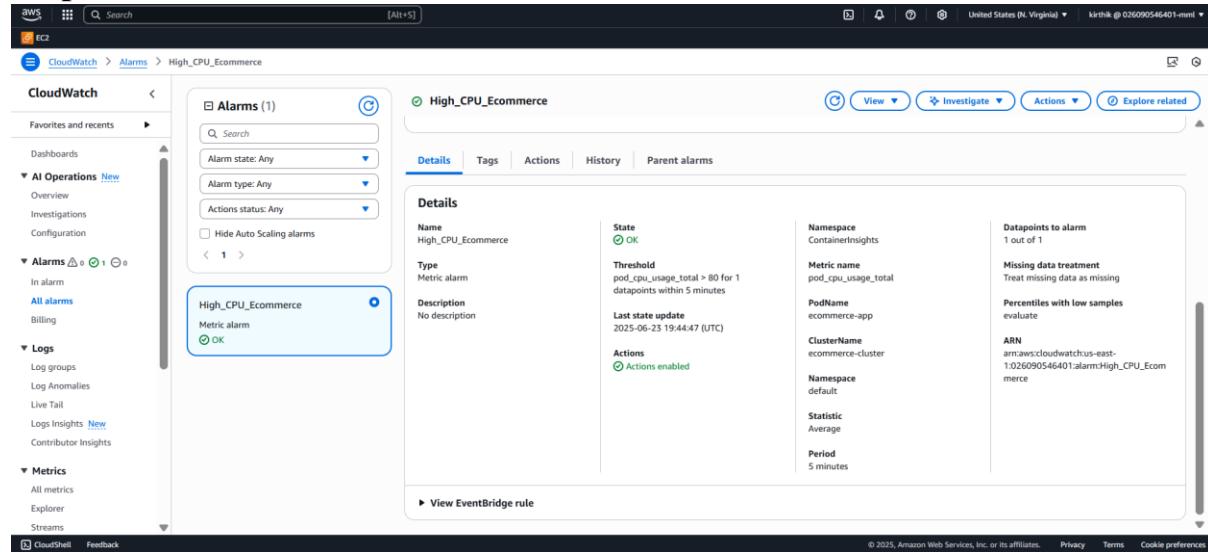
The screenshot shows the 'Health checks' page in the AWS Route 53 console. It lists two health checks: 'primary-frontend1' and 'secondary-front...'. Both checks are currently healthy and have no alarms. The page includes a search bar, a table with columns for ID, Name, Details, Status in last 24 hours, Current s..., Alarm, and Actions, and a 'Create health check' button.

PHASE 5: CloudWatch Monitoring Setup

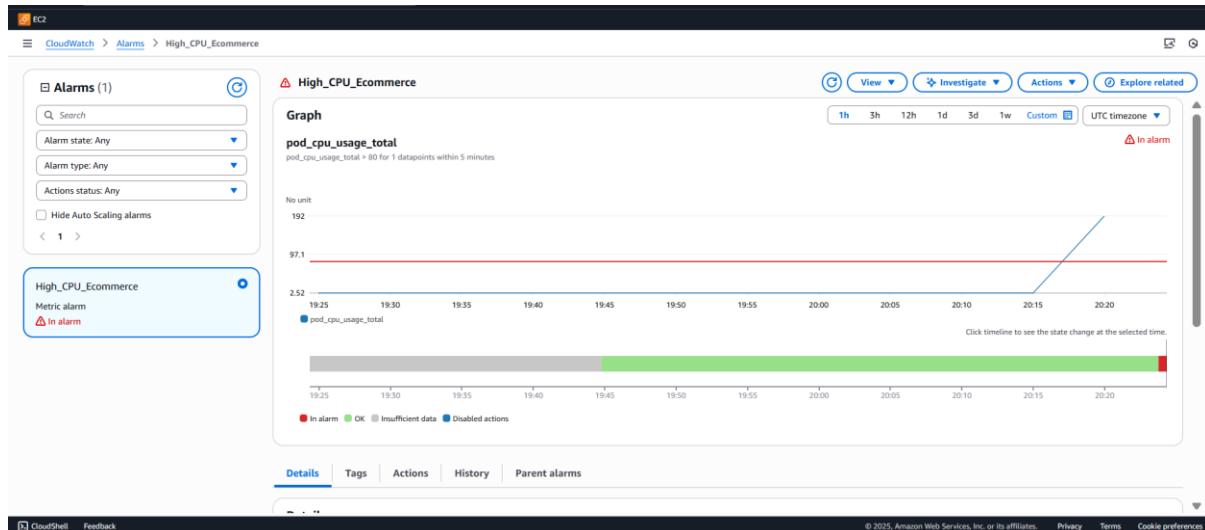
Step 1: Create a CloudWatch



Step 2: Alarms



Alarm:



Triggered Mail:

ALARM: "High_CPU_Ecommerce" in US East (N. Virginia)

You are receiving this email because your Amazon CloudWatch Alarm "High_CPU_Ecommerce" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [153.9168366494024 (23/06/25 20:18:00)] was greater than the threshold (80.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Monday 23 June, 2025 20:23:47 UTC".

View this alarm in the AWS Management Console:
[https://urldefense.com/v3/_https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1&alarmsV2:alarm*High_CPU_Ecommerce_!y8!!PdM5GIU!vdF-E0d7baihNf2xvoRta-qgK6scCrSLDNRDgi3f4UpalVyC_kp5lPu1jLu276g4v74oryqZ8_nQr57UwHozsQM\\$](https://urldefense.com/v3/_https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1&alarmsV2:alarm*High_CPU_Ecommerce_!y8!!PdM5GIU!vdF-E0d7baihNf2xvoRta-qgK6scCrSLDNRDgi3f4UpalVyC_kp5lPu1jLu276g4v74oryqZ8_nQr57UwHozsQM$)

Alarm Details:

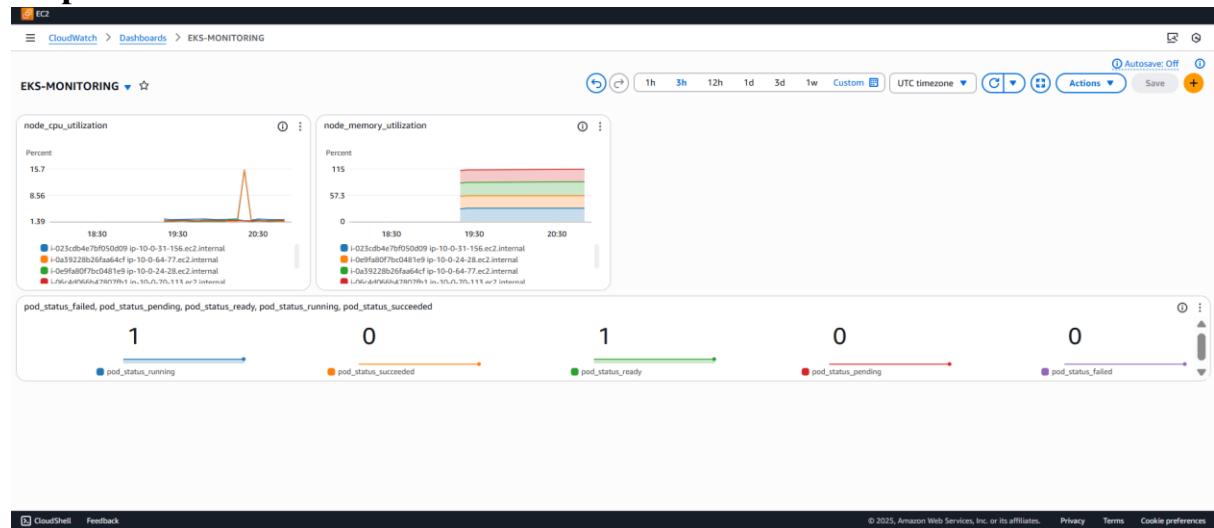
- Name: High_CPU_Ecommerce
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [153.9168366494024 (23/06/25 20:18:00)] was greater than the threshold (80.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Monday 23 June, 2025 20:23:47 UTC
- AWS Account: 026090546401
- Alarm Arn: arn:aws:cloudwatch:us-east-1:026090546401:alarm:High_CPU_Ecommerce

Threshold:

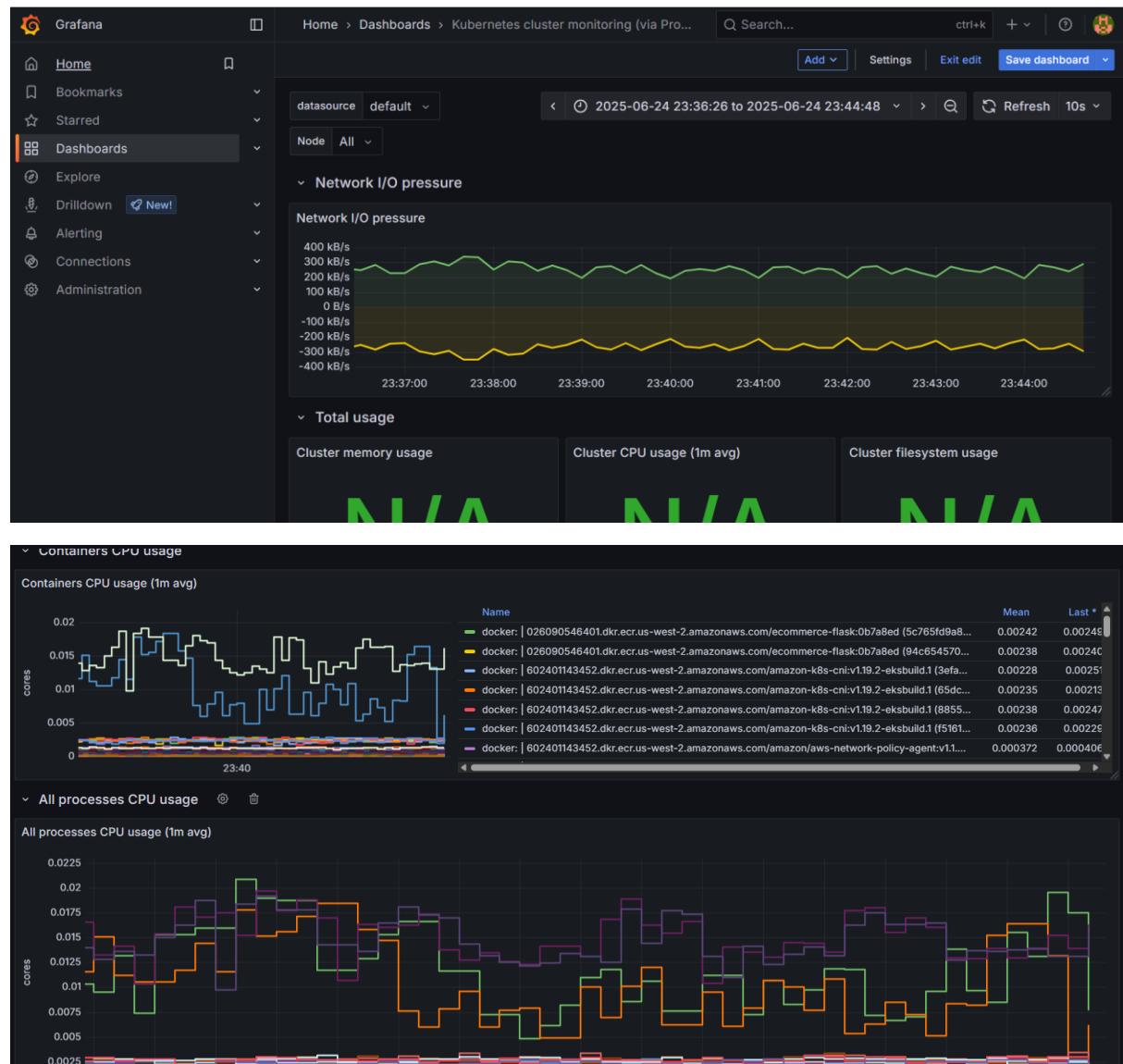
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 80.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:

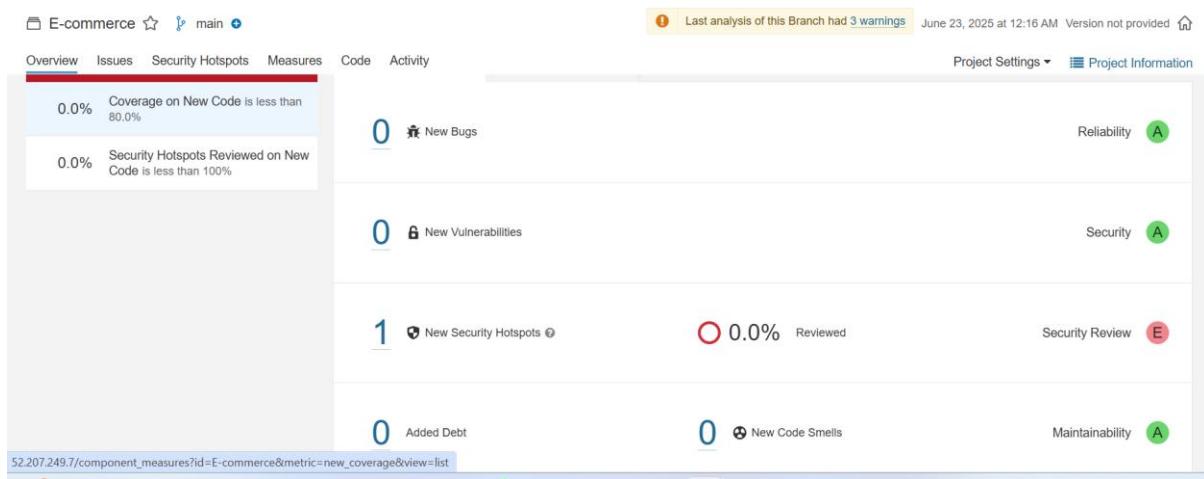
Step 3: CloudWatch dashboards



Grafana



Sonar Qube:



Application:

The screenshot shows a login page for an 'E-Commerce' application. The header includes links for Home, Market, Admin Panel, Login, and Register. The main title is 'Account | Login'. There are two input fields: 'Username..' and 'Password..'. Below the fields is a link 'Don't have an account? [Create Account](#)'. A large yellow button labeled 'Sign In' is centered at the bottom.

Added item: TV

Welcome to Admin Panel

Control Products				
ID	Name	Price	Owner	Control
1	laptop	2,000 \$	None	<button>Update</button> <button>Delete</button>
2	mobile	200 \$	None	<button>Update</button> <button>Delete</button>
3	TV	2,600 \$	None	<button>Update</button> <button>Delete</button>

Control Users				
ID	Username	Budget	Cart	Control
1	kireeti	10,000 \$	Empty	<button>Delete</button>
2	admin	10,000 \$	Empty	<button>Delete</button>

Add New Product

Item Name

Price

Barcode

Description

Add Item

Not secure kirthiksubbiah.com/market

E-Commerce Home Market Admin Panel Admin Logout

Available Items

Name	Barcode	Price	Description	Actions
laptop	RF6DD	2,000 \$	gadget	<button>Info</button> <button>Buy</button>
mobile	6g6g	2,000 \$	samsung	<button>Info</button> <button>Buy</button>
book	ftff	20 \$	science	<button>Info</button> <button>Buy</button>
earphone	h8h88	29 \$	music	<button>Info</button> <button>Buy</button>

Owned Items

|--|--|--|--|--|

