Kimai Cloud Migration Project

Angeline Hephzibah J

GitHub Repository

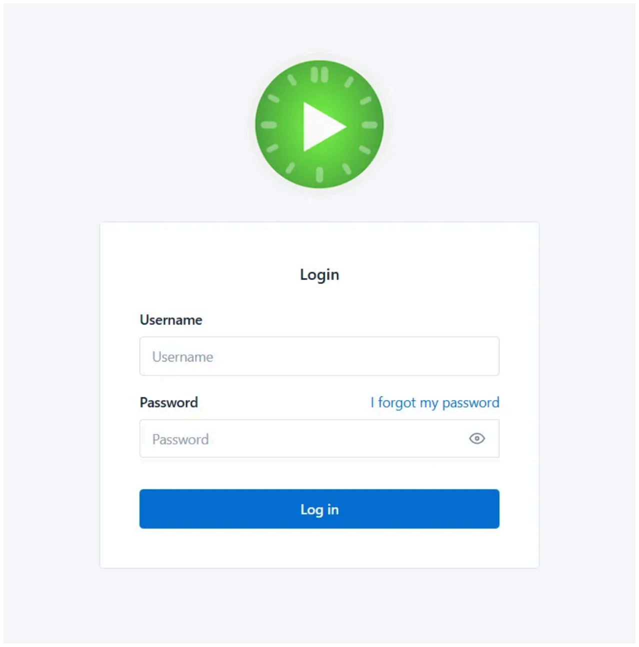
This project is maintained on GitHub:

Repo Name: `Cloud-Migration-Project`

Link: https://github.com/angelinedev/Cloud-Migration-Project

It contains:

- `terraform/` — Infrastructure-as-Code setup (modular with S3 backend)
- `kimai/` — Dockerfile and Docker Compose setup for Kimai
- `Jenkinsfile` — Jenkins Pipeline as Code
- `docs/` — High-Level Design (HLD) and Low-Level Design (LLD)
- `README.md` — Setup guide, architecture documentation, and cost estimation



http://13.48.47.64:8001/

http://kimai-alb-653895671.eu-north-1.elb.amazonaws.com/en/login

Tech Stack

Component	Tech
Backend	PHP (Symfony Framework)
Frontend	HTML/CSS/JS (Bundled)
DB	MariaDB
Server	Nginx
Runtime	PHP-FPM
OS	Amazon Linux 2 (EC2)
IaC	Terraform
Containerization	Docker
CI/CD	Jenkins
Monitoring	CloudWatch, Grafana
Logging	CloudWatch Logs
Access Control	AWS IAM, Bastion Host

Deployment Architecture

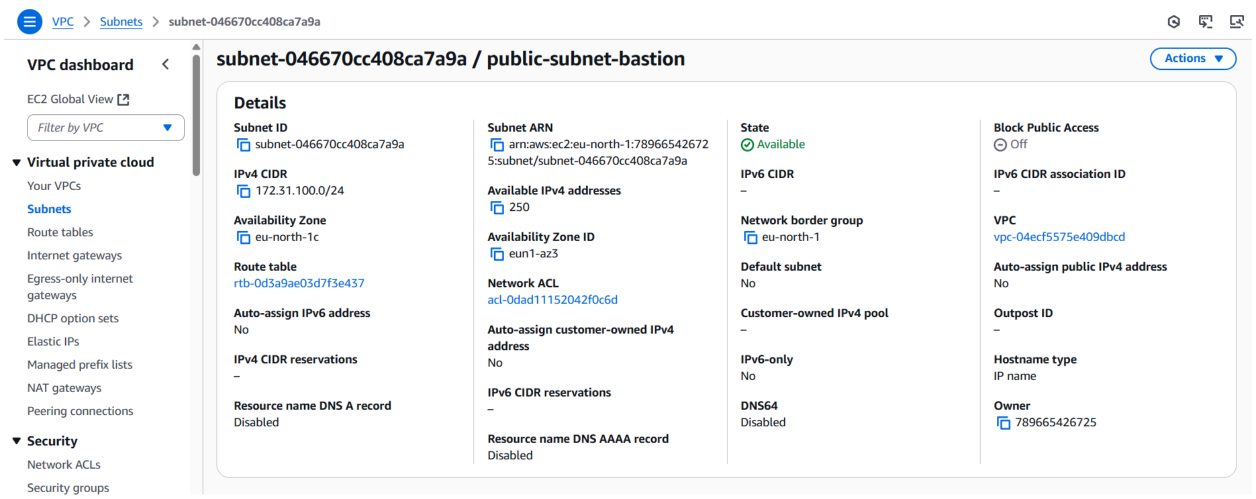
Textual Architecture Diagram

```
[ User ]
   |
[ Load Balancer (Public) + WAF ]
   |
[ Bastion Host (Public Subnet) ]
```

```
  |
[ VPC ]
  ├── EC2 (Kimai + Docker) [Private Subnet]
  └── RDS (MySQL DB)      [Private Subnet]
```

Security Highlights:

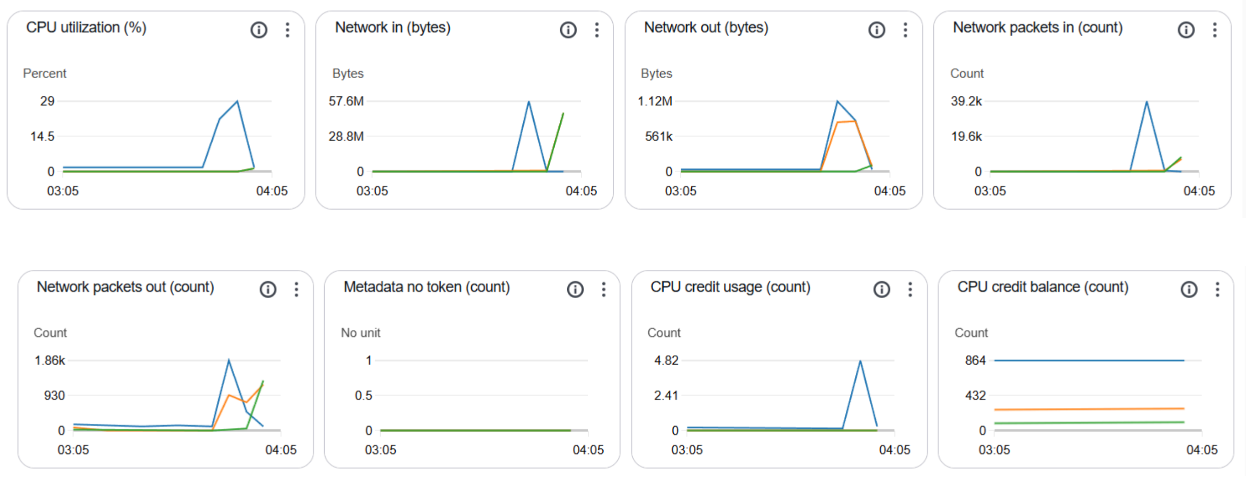
- Bastion host for secure SSH
- IAM with least privilege
- Security Groups with only required ports open
- ALB protected by AWS WAF



subnet-046670cc408ca7a9a / public-subnet-bastion

Details

Subnet ID subnet-046670cc408ca7a9a	**Subnet ARN** arn:aws:ec2:eu-north-1:789665426725:subnet/subnet-046670cc408ca7a9a	**State** Available	**Block Public Access** Off
IPv4 CIDR 172.31.100.0/24	**Available IPv4 addresses** 250	**IPv6 CIDR** –	**IPv6 CIDR association ID** –
Availability Zone eu-north-1c	**Availability Zone ID** eun1-az3	**Network border group** eu-north-1	**VPC** vpc-04ecf5575e409dbcd
Route table rtb-0d3a9ae03d7f3e437	**Network ACL** acl-0dad11152042f0c6d	**Default subnet** No	**Auto-assign public IPv4 address** No
Auto-assign IPv6 address No	**Auto-assign customer-owned IPv4 address** No	**Customer-owned IPv4 pool** –	**Outpost ID** –
IPv4 CIDR reservations –	**IPv6 CIDR reservations** –	**IPv6-only** No	**Hostname type** IP name
Resource name DNS A record Disabled	**Resource name DNS AAAA record** Disabled	**DNS64** Disabled	**Owner** 789665426725



Repository Structure

```
Cloud-Migration-Project/
├─ terraform/
│   ├─ main.tf
│   ├─ variables.tf
│   ├─ outputs.tf
│   └─ modules/
│     └─ ec2/
│       ├─ main.tf
│       ├─ variables.tf
│       └─ outputs.tf
├─ Jenkinsfile
├─ kimai/
│     ├─ Dockerfile
│     └─ docker-compose.yml
├─ docs/
│   ├─ HLD.md
│   └─ LLD.md
└─ README.md
```

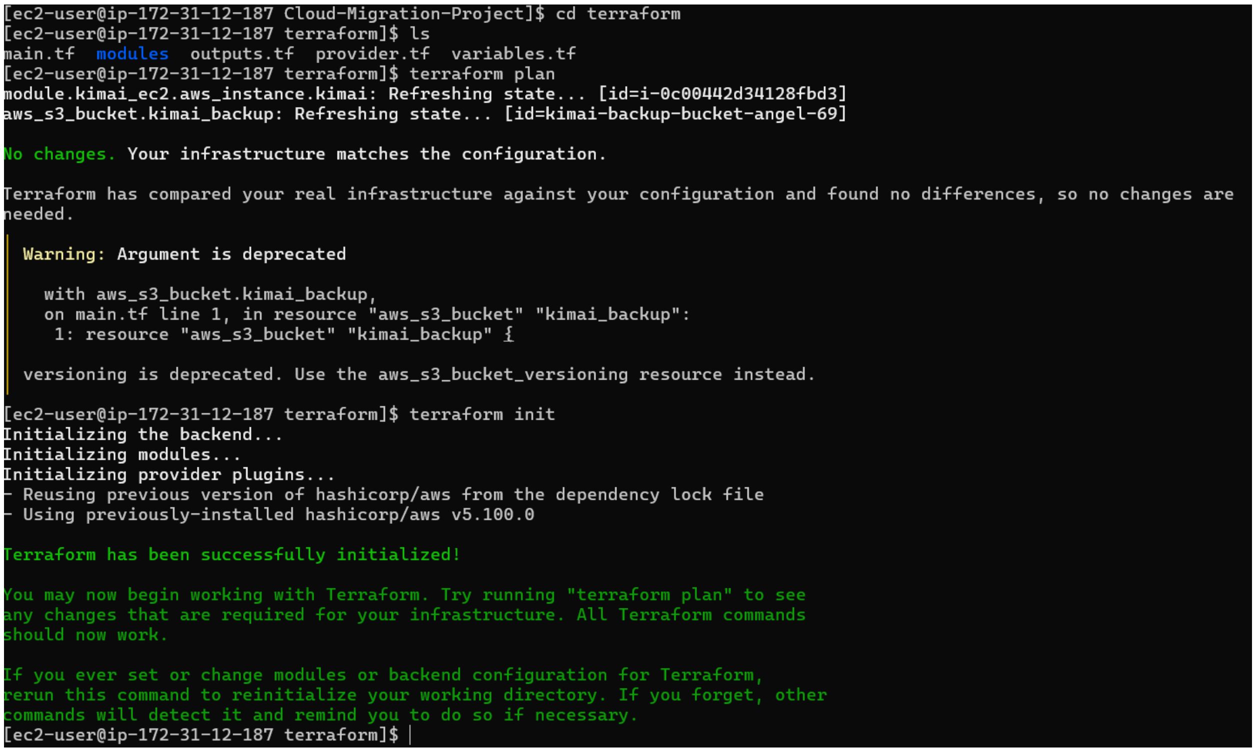
Terraform Setup

All Terraform modules and configuration files are located in the `terraform/` directory inside the GitHub repository.

- Modular setup for reusability
- Remote state managed in S3
- Uses t3.large for Kimai EC2 and t3.micro for Bastion Host
- Outputs public/private IPs, instance IDs

To deploy:

```
cd terraform
terraform init
terraform apply -auto-approve
```



```
[ec2-user@ip-172-31-12-187 Cloud-Migration-Project]$ cd terraform
[ec2-user@ip-172-31-12-187 terraform]$ ls
main.tf  modules  outputs.tf  provider.tf  variables.tf
[ec2-user@ip-172-31-12-187 terraform]$ terraform plan
module.kimai_ec2.aws_instance.kimai: Refreshing state... [id=i-0c00442d34128fbd3]
aws_s3_bucket.kimai_backup: Refreshing state... [id=kimai-backup-bucket-angel-69]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are
needed.
│
│ Warning: Argument is deprecated
│
│   with aws_s3_bucket.kimai_backup,
│   on main.tf line 1, in resource "aws_s3_bucket" "kimai_backup":
│    1: resource "aws_s3_bucket" "kimai_backup" {
│
│ versioning is deprecated. Use the aws_s3_bucket_versioning resource instead.
│
[ec2-user@ip-172-31-12-187 terraform]$ terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.100.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[ec2-user@ip-172-31-12-187 terraform]$ |
```

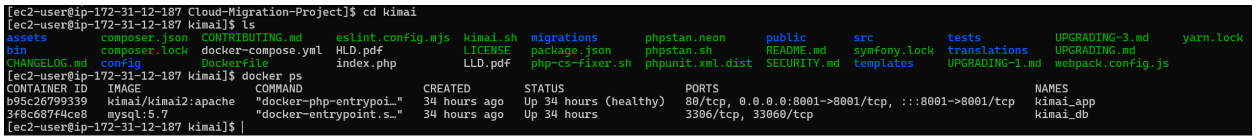
Docker Setup

Kimai is containerized with a multi-stage Dockerfile for optimized builds. The container is:

- Non-root
- Has healthcheck
- Ready for production

To build locally:

```
docker build -t kimai-app .
docker run -p 80:8001 kimai-app
```



```
[ec2-user@ip-172-31-12-187 Cloud-Migration-Project]$ cd kimai
[ec2-user@ip-172-31-12-187 kimai]$ ls
assets         composer.json  CONTRIBUTING.md     eslint.config.mjs  kimai.sh  migrations       phpstan.neon      public     src           tests            UPGRADING-3.md     yarn.lock
bin            composer.lock  docker-compose.yml  HLD.pdf            LICENSE   package.json     phpstan.sh        README.md  symfony.lock  translations     UPGRADING.md
CHANGELOG.md   config         Dockerfile          index.php          LLD.pdf   php-cs-fixer.sh  phpunit.xml.dist  SECURITY.md  templates   UPGRADING-1.md   webpack.config.js
[ec2-user@ip-172-31-12-187 kimai]$ docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED        STATUS                  PORTS                                              NAMES
b95c26799339   kimai/kimai2:apache   "docker-php-entrypoi…"   34 hours ago   Up 34 hours (healthy)   80/tcp, 0.0.0.0:8001->8001/tcp, :::8001->8001/tcp   kimai_app
3f8c687f4ce8   mysql:5.7             "docker-entrypoint.s…"   34 hours ago   Up 34 hours             3306/tcp, 33060/tcp                                kimai_db
[ec2-user@ip-172-31-12-187 kimai]$ |
```

CI/CD Pipeline (Jenkins)

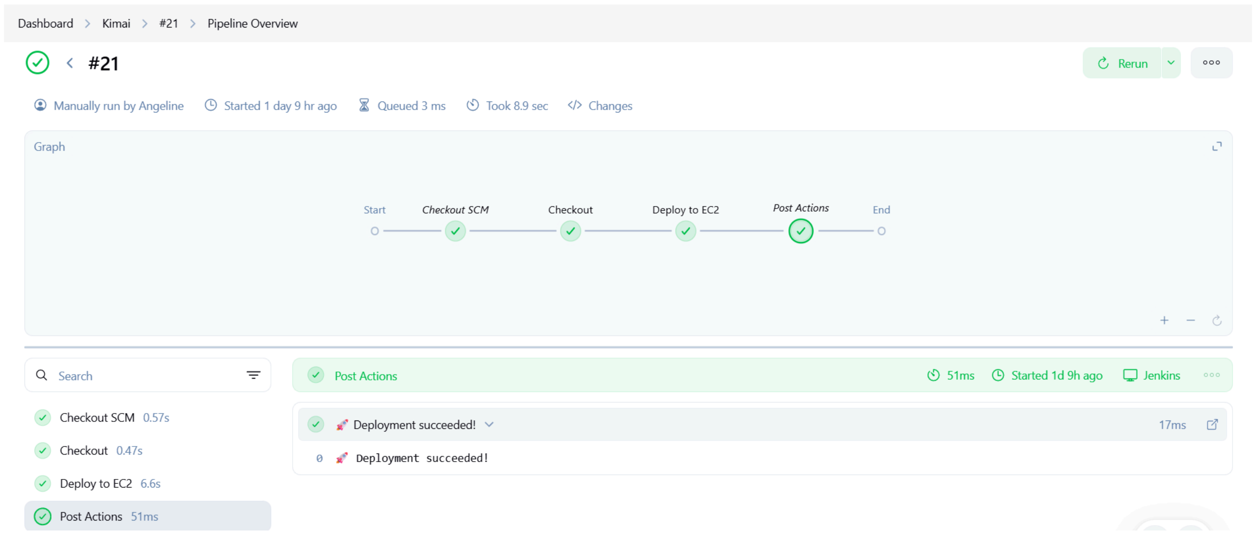
Pipeline is stored as `Jenkinsfile` in the GitHub repository.

Trigger: Push to `main` branch

Stages:

- Checkout
- Build Docker Image
- Run Tests
- Push to DockerHub
- Deploy via SSH to EC2

Pipeline is fully automated using a `Jenkinsfile`.


Dashboard > Kimai > #21 > Pipeline Overview

#21

Manually run by Angeline · Started 1 day 9 hr ago · Queued 3 ms · Took 8.9 sec · Changes

Checkout SCM 0.57s
Checkout 0.47s
Deploy to EC2 6.6s
Post Actions 51ms

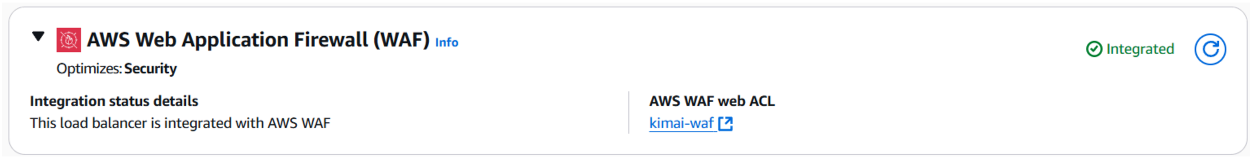
Post Actions · 51ms · Started 1d 9h ago · Jenkins

Deployment succeeded! 17ms

```
0  🚀 Deployment succeeded!
```

Security

- WAF filters traffic on Load Balancer
- Bastion Host enables safe key-based access to private instances
- IAM Role attached to Kimai EC2 for CloudWatch agent and S3 access

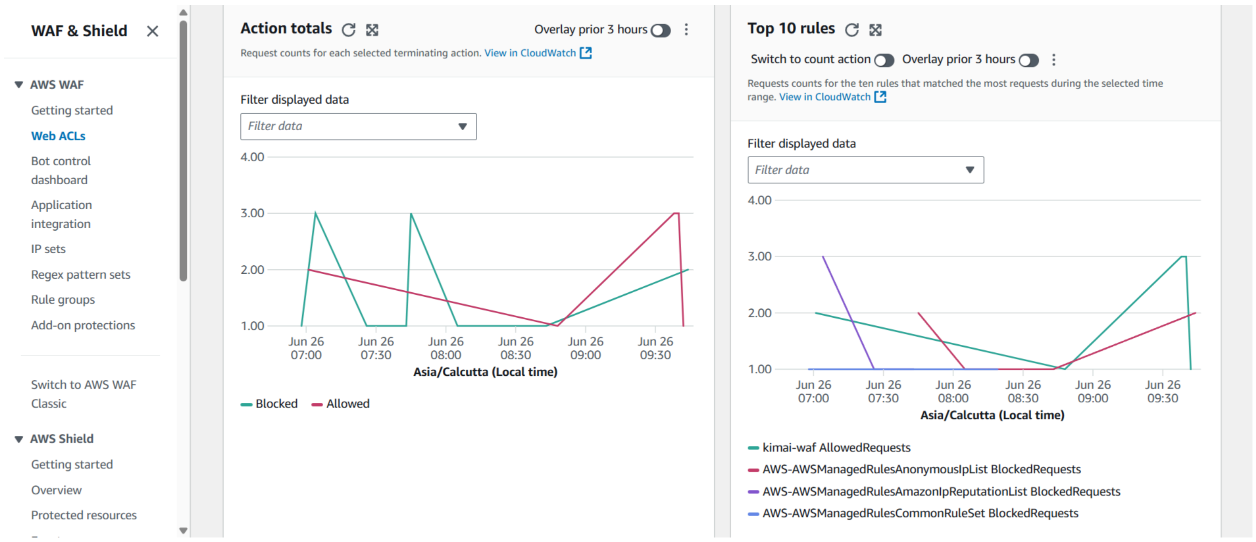

AWS Web Application Firewall (WAF) Info

Optimizes: **Security**

Integrated

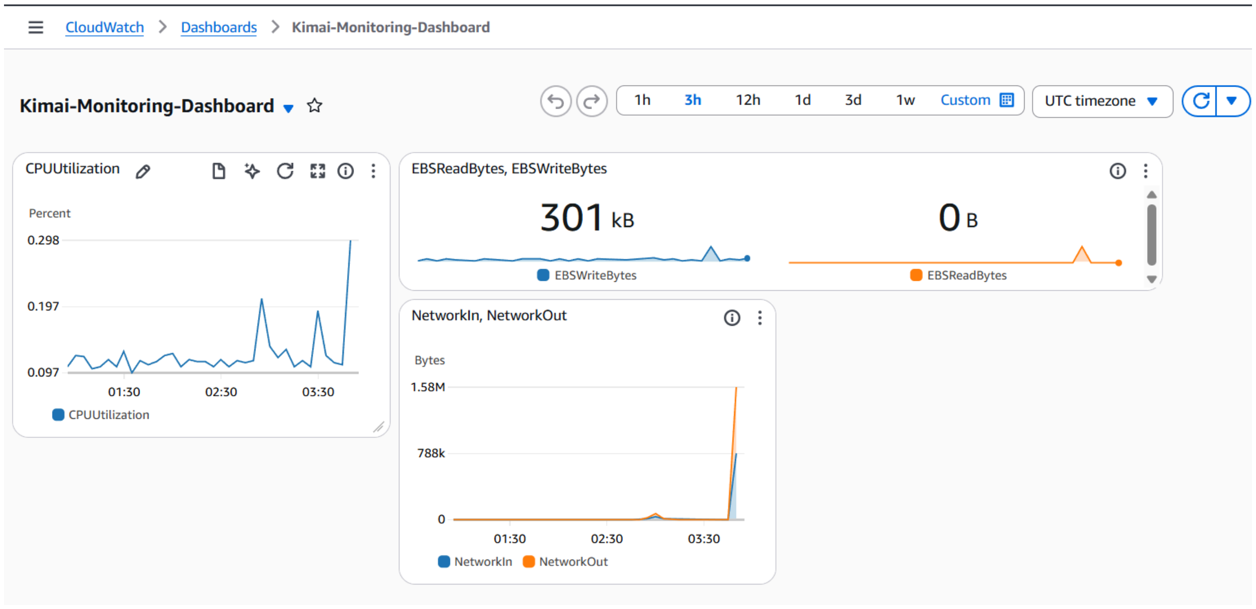
Integration status details
This load balancer is integrated with AWS WAF

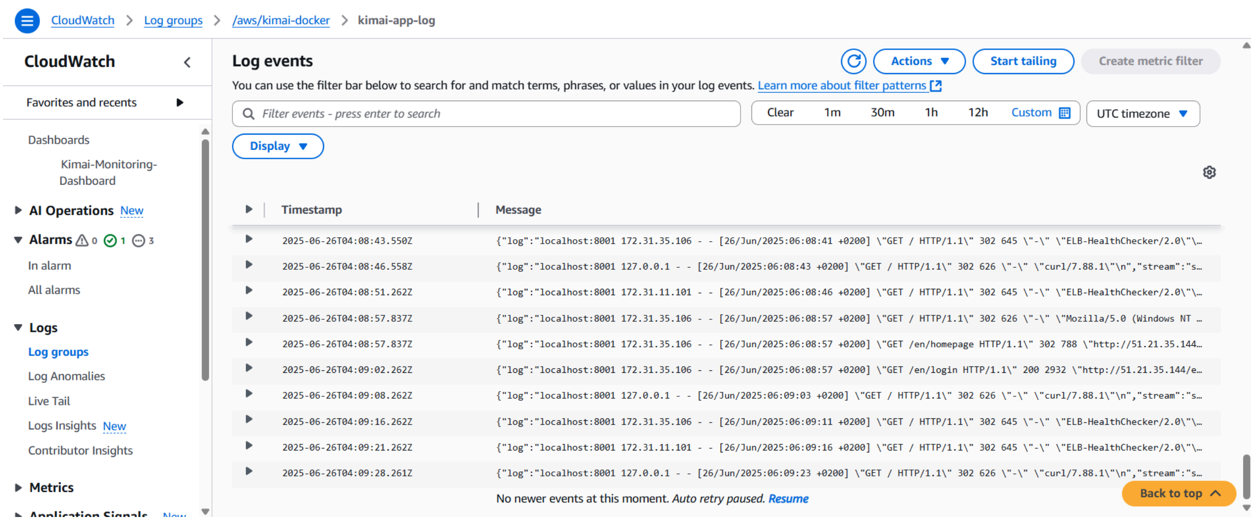
AWS WAF web ACL
kimai-waf



Monitoring & Logging

- CloudWatch Agent installed and configured on EC2
- Monitors:
  - CPU usage
  - Memory and Disk space
  - Application logs
- Alerts configured for:
  - High CPU (> 80%)
  - Health Check failures

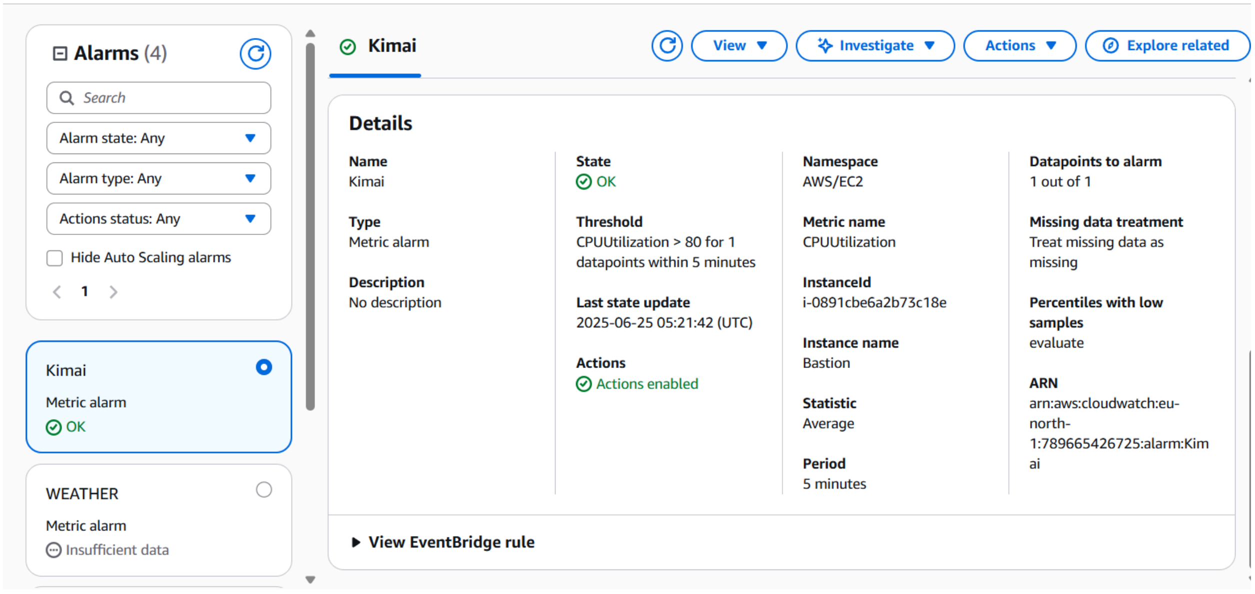




You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                  Message
2025-06-26T04:08:43.550Z   {"log":"localhost:8001 172.31.35.106 - - [26/Jun/2025:06:08:41 +0200] \"GET / HTTP/1.1\" 302 645 \"-\" \"ELB-HealthChecker/2.0\"\…
2025-06-26T04:08:46.558Z   {"log":"localhost:8001 127.0.0.1 - - [26/Jun/2025:06:08:43 +0200] \"GET / HTTP/1.1\" 302 626 \"-\" \"curl/7.88.1\"\n","stream":"s…
2025-06-26T04:08:51.262Z   {"log":"localhost:8001 172.31.11.101 - - [26/Jun/2025:06:08:46 +0200] \"GET / HTTP/1.1\" 302 645 \"-\" \"ELB-HealthChecker/2.0\"\…
2025-06-26T04:08:57.837Z   {"log":"localhost:8001 172.31.35.106 - - [26/Jun/2025:06:08:57 +0200] \"GET / HTTP/1.1\" 302 626 \"-\" \"Mozilla/5.0 (Windows NT …
2025-06-26T04:08:57.837Z   {"log":"localhost:8001 172.31.35.106 - - [26/Jun/2025:06:08:57 +0200] \"GET /en/homepage HTTP/1.1\" 302 788 \"http://51.21.35.144…
2025-06-26T04:09:02.262Z   {"log":"localhost:8001 172.31.35.106 - - [26/Jun/2025:06:08:57 +0200] \"GET /en/login HTTP/1.1\" 200 2932 \"http://51.21.35.144/e…
2025-06-26T04:09:08.262Z   {"log":"localhost:8001 127.0.0.1 - - [26/Jun/2025:06:09:03 +0200] \"GET / HTTP/1.1\" 302 626 \"-\" \"curl/7.88.1\"\n","stream":"s…
2025-06-26T04:09:16.262Z   {"log":"localhost:8001 172.31.35.106 - - [26/Jun/2025:06:09:11 +0200] \"GET / HTTP/1.1\" 302 645 \"-\" \"ELB-HealthChecker/2.0\"\…
2025-06-26T04:09:21.262Z   {"log":"localhost:8001 172.31.11.101 - - [26/Jun/2025:06:09:16 +0200] \"GET / HTTP/1.1\" 302 645 \"-\" \"ELB-HealthChecker/2.0\"\…
2025-06-26T04:09:28.261Z   {"log":"localhost:8001 127.0.0.1 - - [26/Jun/2025:06:09:23 +0200] \"GET / HTTP/1.1\" 302 626 \"-\" \"curl/7.88.1\"\n","stream":"s…
```

No newer events at this moment. *Auto retry paused.* Resume



Alarm	Type	State
Kimai	Metric alarm	OK
WEATHER	Metric alarm	Insufficient data

Kimai — Details

Name	Kimai
Type	Metric alarm
Description	No description
State	OK
Threshold	CPUUtilization > 80 for 1 datapoints within 5 minutes
Last state update	2025-06-25 05:21:42 (UTC)
Actions	Actions enabled
Namespace	AWS/EC2
Metric name	CPUUtilization
InstanceId	i-0891cbe6a2b73c18e
Instance name	Bastion
Statistic	Average
Period	5 minutes
Datapoints to alarm	1 out of 1
Missing data treatment	Treat missing data as missing
Percentiles with low samples	evaluate
ARN	arn:aws:cloudwatch:eu-north-1:789665426725:alarm:Kimai

View EventBridge rule

Grafana Setup

- Grafana installed on EC2
- Connected to CloudWatch via IAM role
- Dashboards created for:
  - CPU Utilization
  - Memory Usage
  - Disk Usage
  - Network Traffic

- Alerts enabled with thresholds

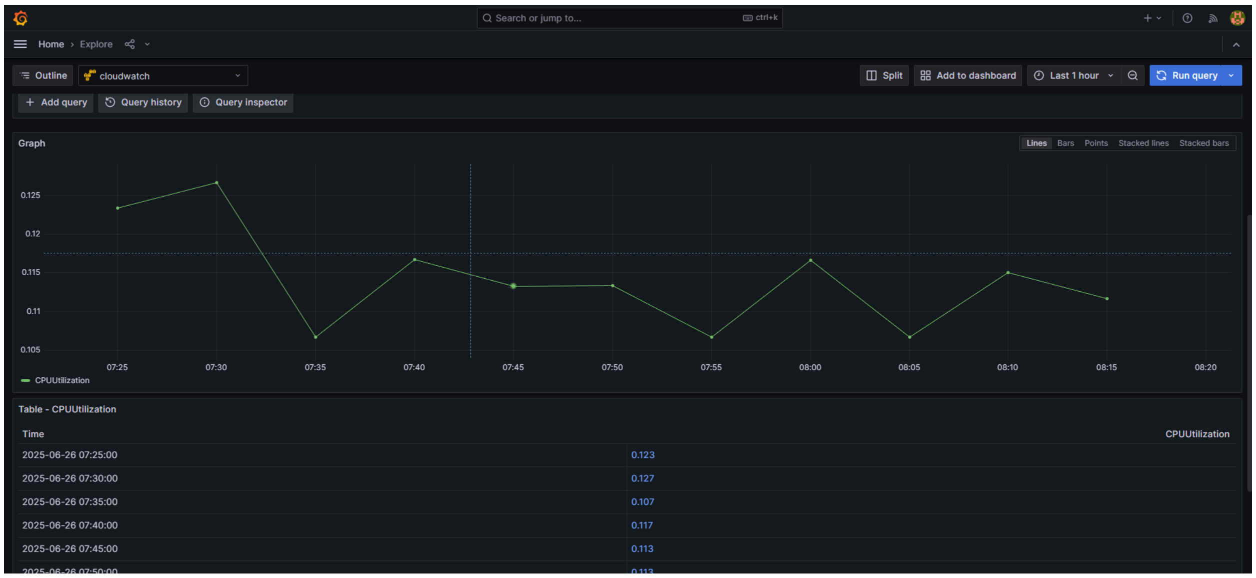


Table - CPUUtilization

Time	CPUUtilization
2025-06-26 07:25:00	0.123
2025-06-26 07:30:00	0.127
2025-06-26 07:35:00	0.107
2025-06-26 07:40:00	0.117
2025-06-26 07:45:00	0.113
2025-06-26 07:50:00	0.113

Cost Estimation (Monthly)

Resource	Cost
EC2 (t3.large)	~$33.28
Bastion Host (t3.micro)	~$7.62
EBS (28GB total)	~$2.80
S3 (backups + state)	~$0.12
CloudWatch	~$2.00
Total	**~$45–48/month**

Notes for Teams

- Use Git to always pull latest changes
- All setup scripts are idempotent
- Logs and backups go to AWS
- Easily replicable using Terraform anywhere
