Kimai Cloud Migration Project

GITHUB REPOSITORY

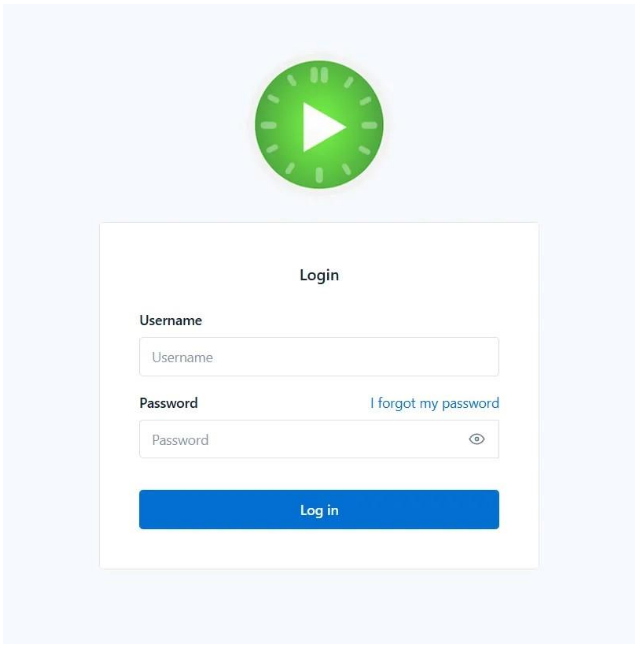
This project is maintained on GitHub:

Repo Name: Cloud-Migration-Project

Link: https://github.com/dhanalakshmim-eng/Cloud-Migration-Project2.git

It contains:

- **terraform/** – Infrastructure-as-Code (IaC) configuration using modular architecture with S3 as the remote backend for state management.
- **kimai/** – Containerization setup for the Kimai application, including a Dockerfile and Docker Compose configuration.
- **Jenkinsfile** – Declarative Jenkins Pipeline script defining the CI/CD workflow stages.
- **docs/** – Comprehensive documentation including High-Level Design (HLD) and Low-Level Design (LLD) artifacts.
- **README.md** – Contains setup instructions, system architecture overview, and cloud cost estimation details.



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

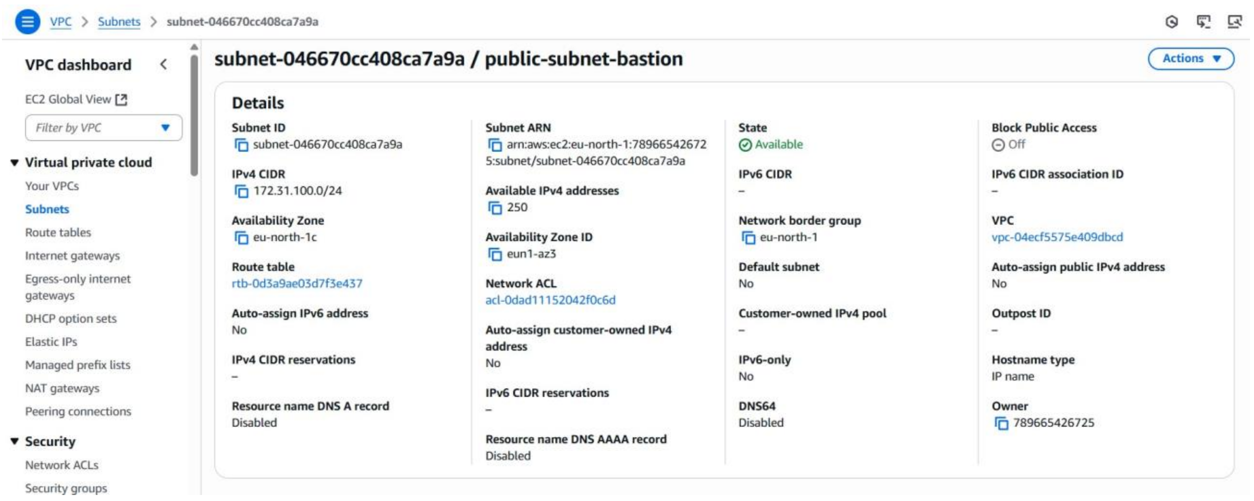
Deployment Architecture

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

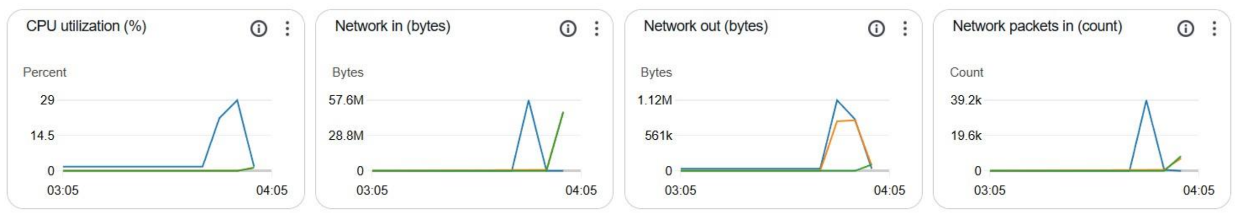


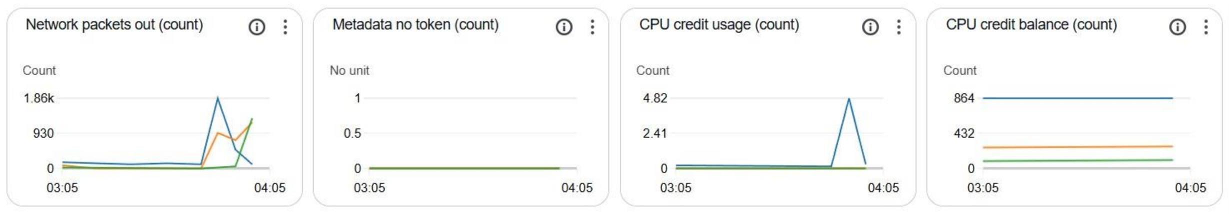
subnet-046670cc408ca7a9a / public-subnet-bastion

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

IAM with least privilege





Repository Structure

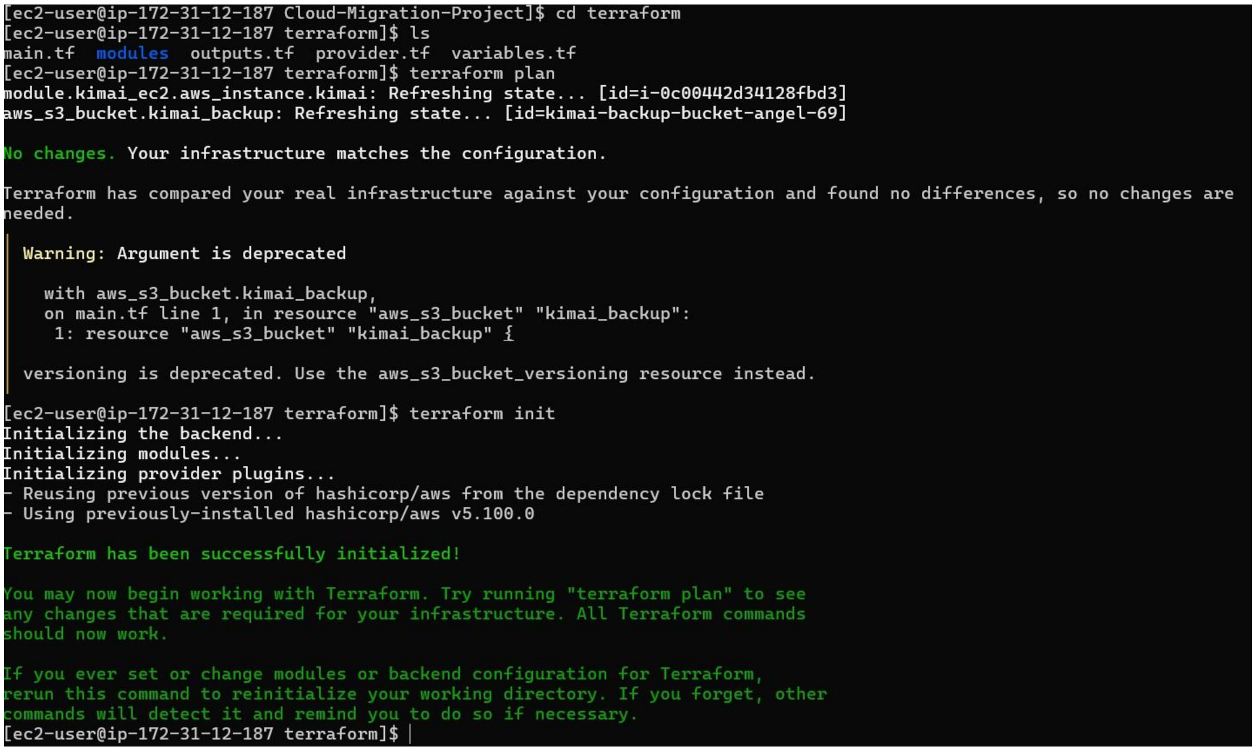
```
Cloud-Migration-Project/
├── terraform/
│   ├── main.tf
│   ├── variables.tf
│   ├── outputs.tf
│   └── modules/
│     └── ec2/
│         ├── main.tf
│         ├── variables.tf
│         └── outputs.tf
├── Jenkinsfile
├── kimai/
│     ├── Dockerfile
│     └── docker-compose.yml
├── docs/
│   ├──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

Terraform Setup

All Terraform modules and configuration files are organized within the terraform/ directory of the GitHub repository.
Modular Design: Enables reusability and maintainability of infrastructure components.
Remote State Management: Terraform state is securely managed using Amazon S3.
EC2 Instances:
t3.large instance is provisioned for the Kimai application.
t3.micro instance is used for the Bastion Host.
Output Variables: Includes public and private IP addresses, as well as instance IDs for downstream use.

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]$ |
```

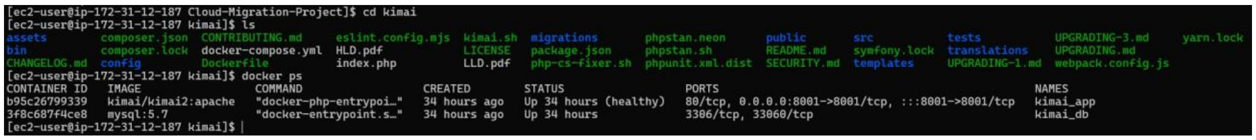
Docker Setup

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
```



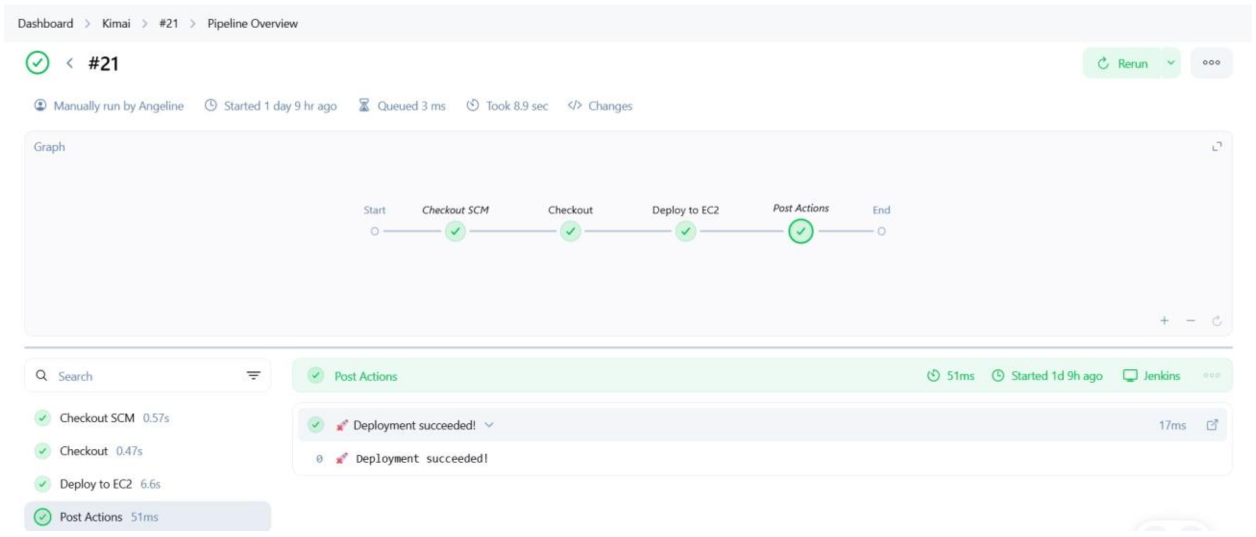
```
[ec2-user@ip-172-31-12-187 Cloud-Migration-Project]$ cd kimai
[ec2-user@ip-172-31-12-187 kimai]$ ls
assets        composer.json  CONTRIBUTING.md     eslint.config.mjs  kimai.sh  migrations        phpstan.neon       public         src            tests              UPGRADING-3.md     yarn.lock
bin           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

The CI/CD pipeline is defined in a Jenkinsfile located in the GitHub repository.

- **Trigger**: Automatically initiated on every push to the main branch.
- **Pipeline Stages**:
    1. **Checkout** – Retrieves the latest code from the repository.
    2. **Build** – Constructs the Docker image for the Kimai application.
    3. **Test** – Executes defined tests to ensure application integrity.
    4. **Push** – Uploads the built Docker image to DockerHub.
    5. **Deploy** – Connects via SSH to the EC2 instance and deploys the updated container.

Pipeline is fully automated using a `Jenkinsfile`.



```
0 🚀 Deployment succeeded!
```

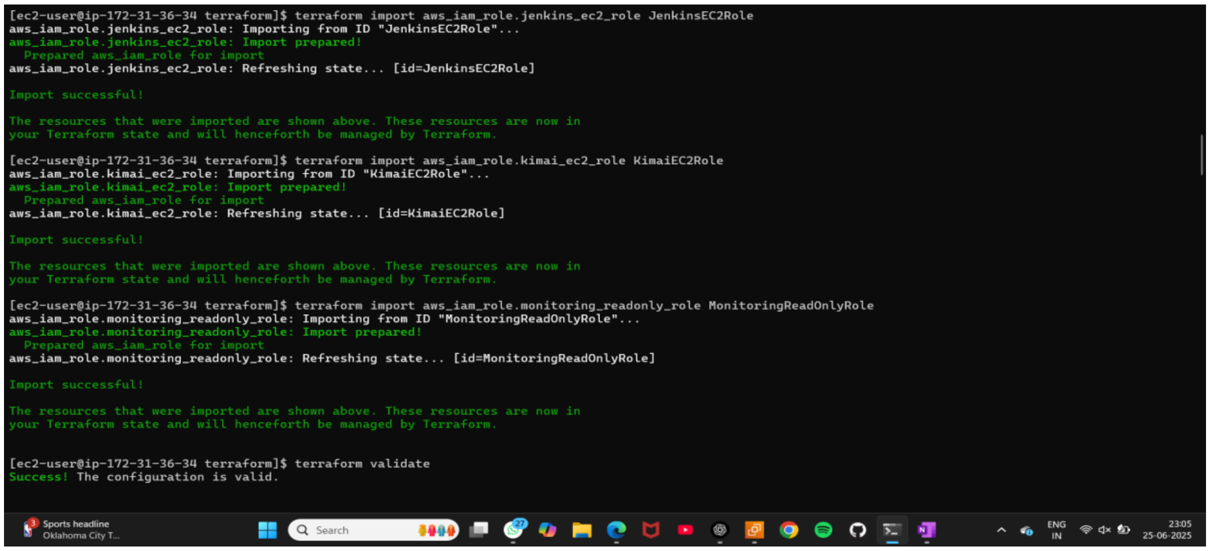
Security

IAM Champ – Least Privilege IAM Roles

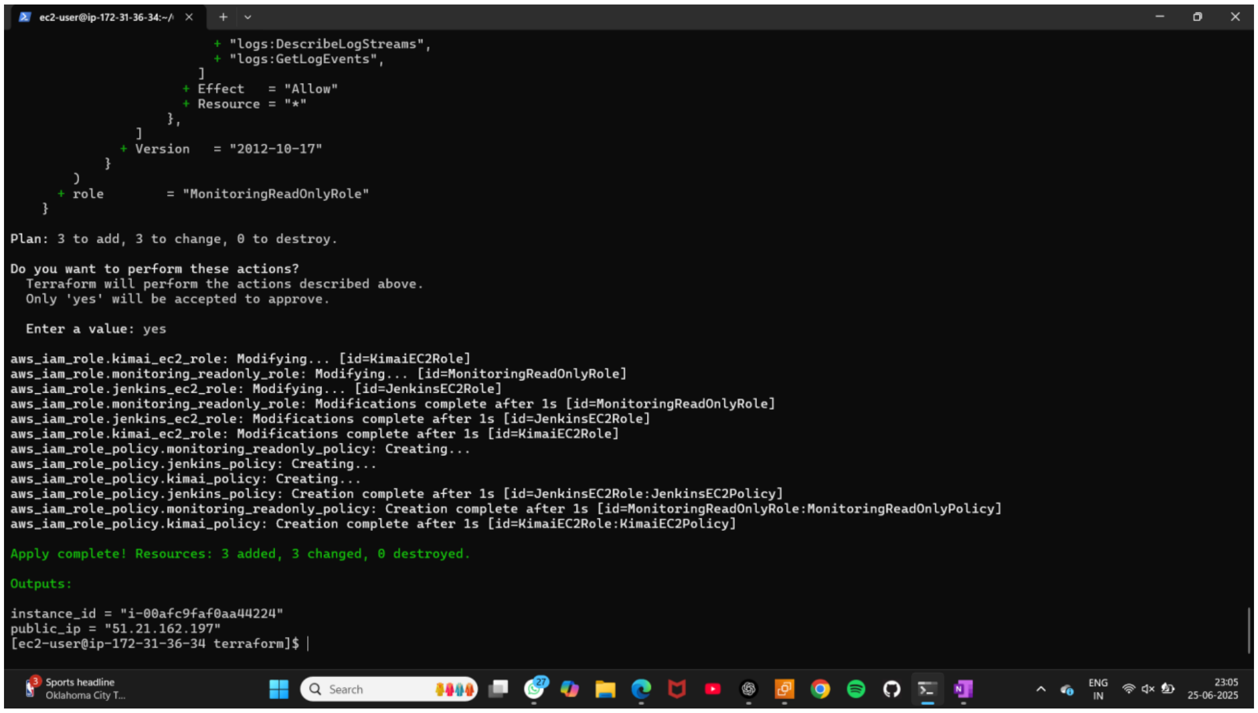
- Created three distinct IAM roles:
    - **JenkinsEC2Role**: Allows Jenkins server limited access to EC2 and S3 services.
    - **KimaiEC2Role**: Grants Kimai container access to CloudWatch and S3.
    - **MonitoringReadOnlyRole**: Provides read-only access for metrics via CloudWatch.
- IAM policies were defined in Terraform using aws_iam_role and aws_iam_role_policy resources.
- Attached roles to EC2 instances via instance profiles.

Network Hardener – Bastion, SGs, and WAF

- Launched a **Bastion Host** in a public subnet to securely SSH into private EC2s.
- Configured **Security Groups**:
    - Only port 22 open to **my IP** for SSH.
    - HTTP (80), HTTPS (443), Prometheus (9090), Grafana (3000), and Node Exporter (9100) only as needed.
- **WAF** configured on ALB for filtering traffic with AWS Managed Rules.
- Verified SSH access to internal Kimai EC2 via Bastion using private IP and .pem key.

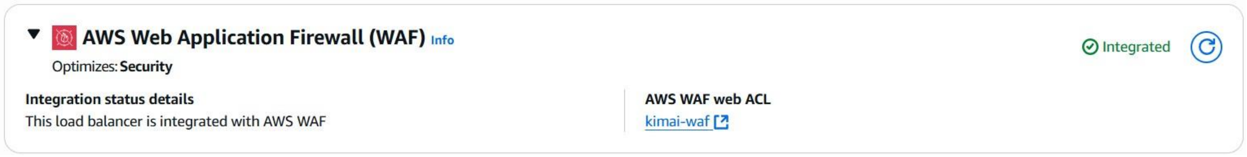


```
[ec2-user@ip-172-31-36-34 terraform]$ terraform import aws_iam_role.jenkins_ec2_role JenkinsEC2Role
aws_iam_role.jenkins_ec2_role: Importing from ID "JenkinsEC2Role"...
aws_iam_role.jenkins_ec2_role: Import prepared!
  Prepared aws_iam_role for import
aws_iam_role.jenkins_ec2_role: Refreshing state... [id=JenkinsEC2Role]

Import successful!

The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.

[ec2-user@ip-172-31-36-34 terraform]$ terraform import aws_iam_role.kimai_ec2_role KimaiEC2Role
aws_iam_role.kimai_ec2_role: Importing from ID "KimaiEC2Role"...
aws_iam_role.kimai_ec2_role: Import prepared!
  Prepared aws_iam_role for import
aws_iam_role.kimai_ec2_role: Refreshing state... [id=KimaiEC2Role]

Import successful!

The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.

[ec2-user@ip-172-31-36-34 terraform]$ terraform import aws_iam_role.monitoring_readonly_role MonitoringReadOnlyRole
aws_iam_role.monitoring_readonly_role: Importing from ID "MonitoringReadOnlyRole"...
aws_iam_role.monitoring_readonly_role: Import prepared!
  Prepared aws_iam_role for import
aws_iam_role.monitoring_readonly_role: Refreshing state... [id=MonitoringReadOnlyRole]

Import successful!

The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.


[ec2-user@ip-172-31-36-34 terraform]$ terraform validate
Success! The configuration is valid.
```

- 

```
                          + "logs:DescribeLogStreams",
                          + "logs:GetLogEvents",
                        ]
                      + Effect   = "Allow"
                      + Resource = "*"
                    },
                ]
              + Version   = "2012-10-17"
            }
        )
      + role         = "MonitoringReadOnlyRole"
    }

Plan: 3 to add, 3 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_iam_role.kimai_ec2_role: Modifying... [id=KimaiEC2Role]
aws_iam_role.monitoring_readonly_role: Modifying... [id=MonitoringReadOnlyRole]
aws_iam_role.jenkins_ec2_role: Modifying... [id=JenkinsEC2Role]
aws_iam_role.monitoring_readonly_role: Modifications complete after 1s [id=MonitoringReadOnlyRole]
aws_iam_role.jenkins_ec2_role: Modifications complete after 1s [id=JenkinsEC2Role]
aws_iam_role.kimai_ec2_role: Modifications complete after 1s [id=KimaiEC2Role]
aws_iam_role_policy.monitoring_readonly_policy: Creating...
aws_iam_role_policy.jenkins_policy: Creating...
aws_iam_role_policy.kimai_policy: Creating...
aws_iam_role_policy.jenkins_policy: Creation complete after 1s [id=JenkinsEC2Role:JenkinsEC2Policy]
aws_iam_role_policy.monitoring_readonly_policy: Creation complete after 1s [id=MonitoringReadOnlyRole:MonitoringReadOnlyPolicy]
aws_iam_role_policy.kimai_policy: Creation complete after 1s [id=KimaiEC2Role:KimaiEC2Policy]

Apply complete! Resources: 3 added, 3 changed, 0 destroyed.

Outputs:

instance_id = "i-00afc9faf0aa44224"
public_ip = "51.21.162.197"
[ec2-user@ip-172-31-36-34 terraform]$ |
```

- 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



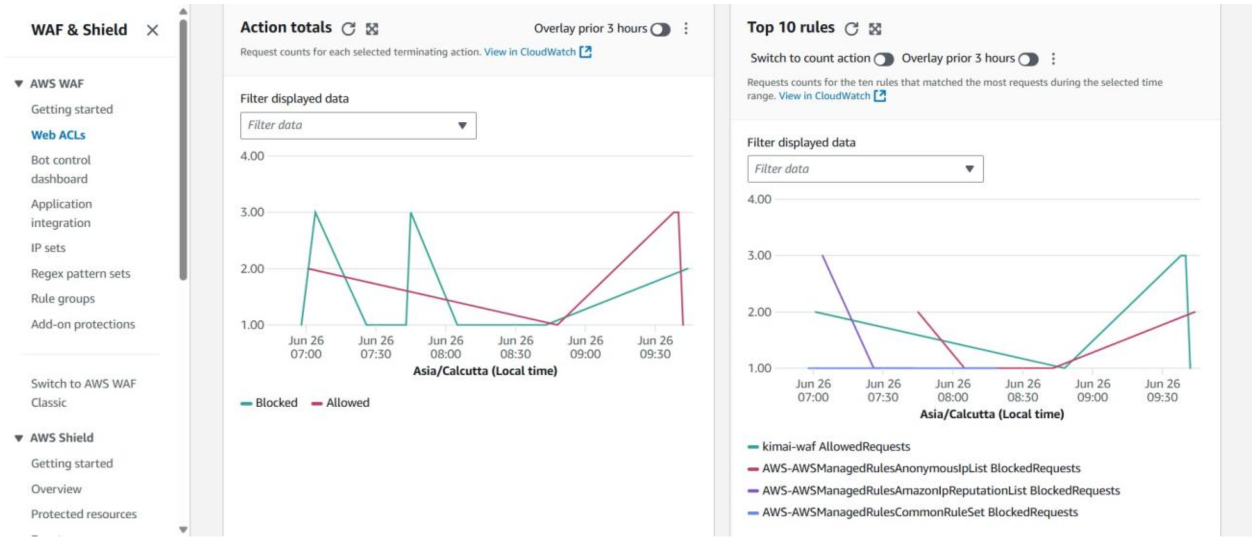
AWS Web Application Firewall (WAF) Info

Optimizes: **Security**

Integrated

Integration status details
This load balancer is integrated with AWS WAF

AWS WAF web ACL
kimai-waf

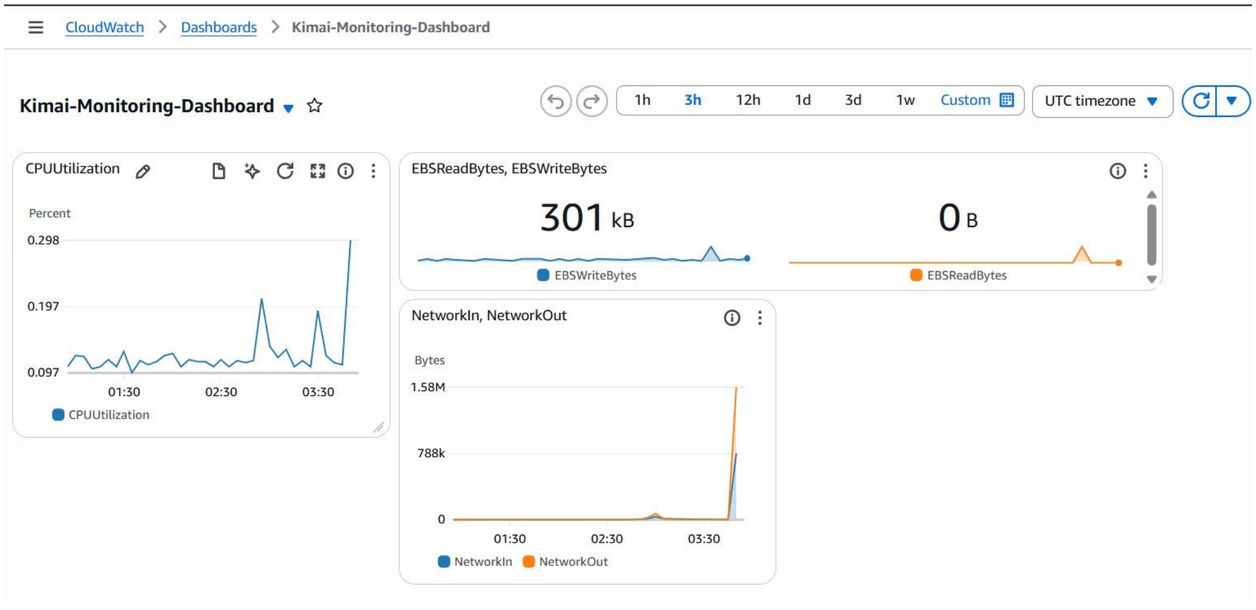


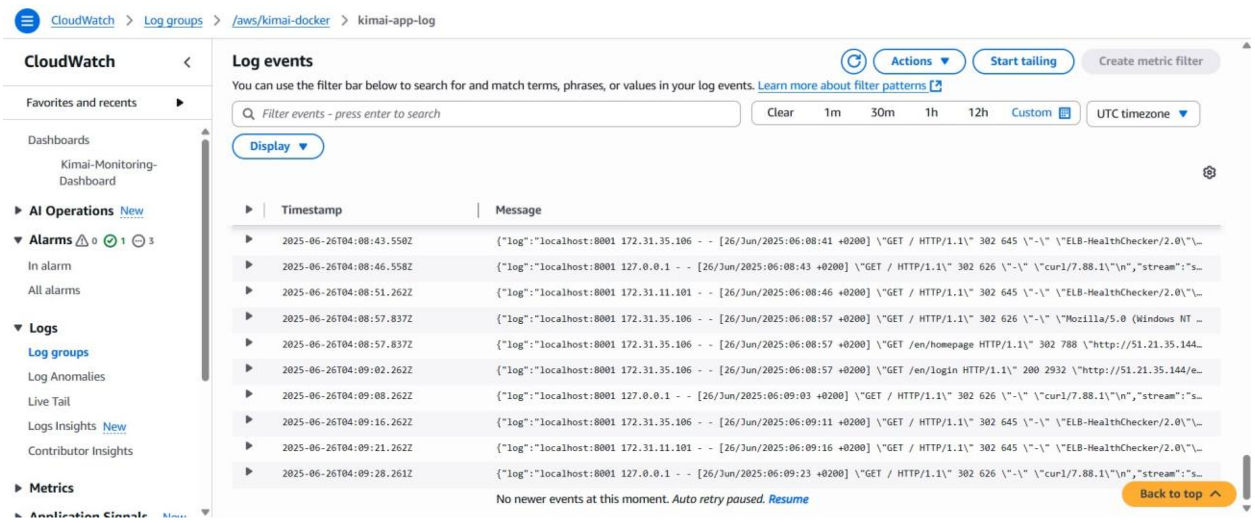
Request counts for each selected terminating action. View in CloudWatch

Requests counts for the ten rules that matched the most requests during the selected time range. View in CloudWatch

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



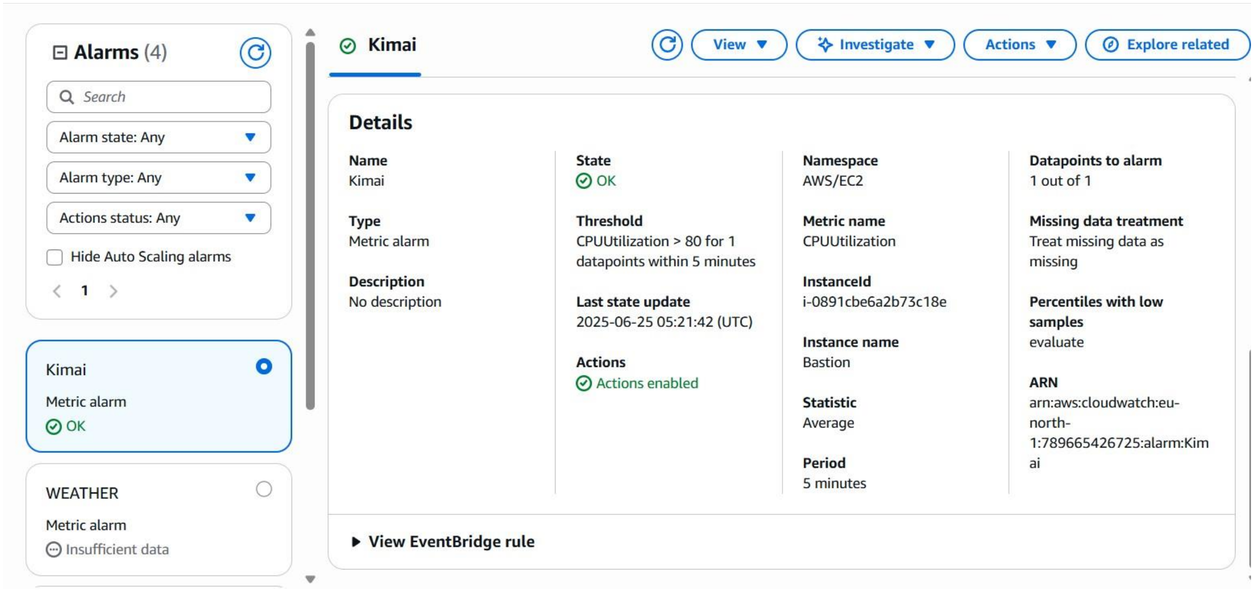


Log events

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



Kimai — Metric alarm — OK

WEATHER — Metric alarm — Insufficient data

Kimai

Details

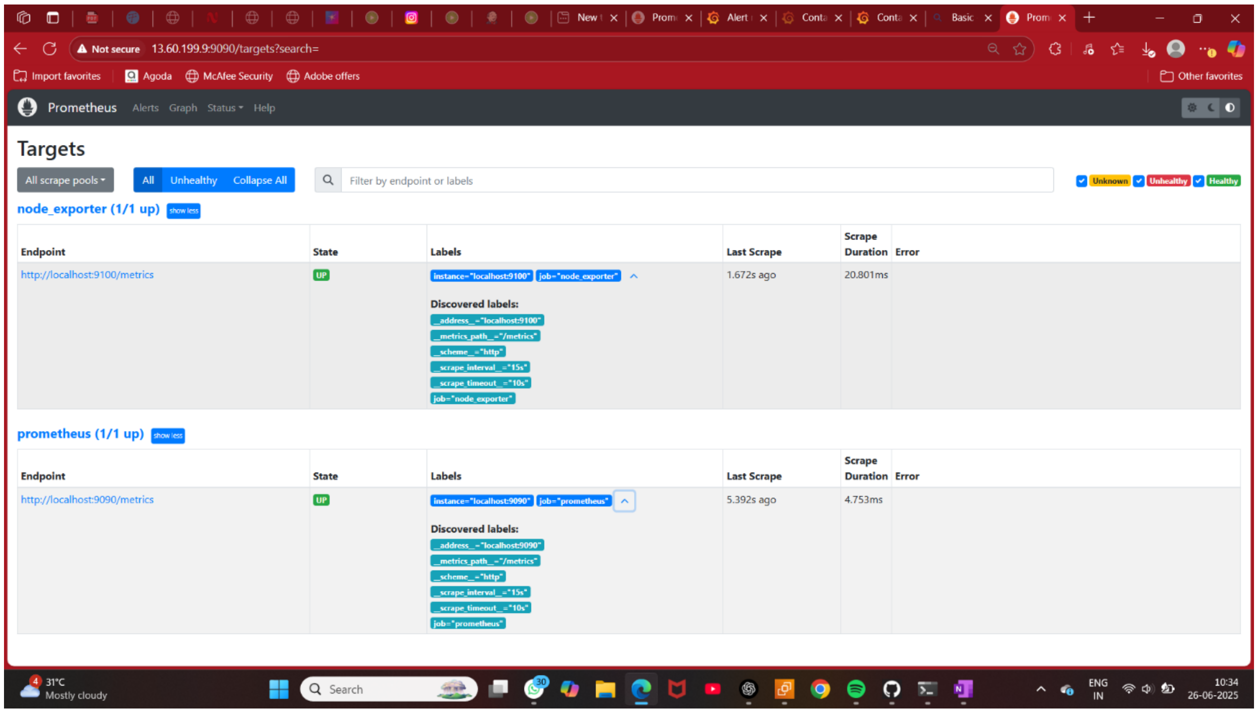
Name Kimai	**State** OK	**Namespace** AWS/EC2	**Datapoints to alarm** 1 out of 1
Type Metric alarm	**Threshold** CPUUtilization > 80 for 1 datapoints within 5 minutes	**Metric name** CPUUtilization	**Missing data treatment** Treat missing data as missing
Description No description	**Last state update**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

View EventBridge rule

Observability

Observability Officer – Prometheus & Grafana

- Deployed **Prometheus** on the same EC2 as Kimai:
    - Configured /etc/prometheus/prometheus.yml with targets:


13.60.199.9:9090/targets?search=

Targets

node_exporter (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node_exporter" Discovered labels: __address__="localhost:9100" __metrics_path__="/metrics" __scheme__="http" __scrape_interval__="15s" __scrape_timeout__="10s" job="node_exporter"	1.672s ago	20.801ms	

prometheus (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus" Discovered labels: __address__="localhost:9090" __metrics_path__="/metrics" __scheme__="http" __scrape_interval__="15s" __scrape_timeout__="10s" job="prometheus"	5.392s ago	4.753ms	

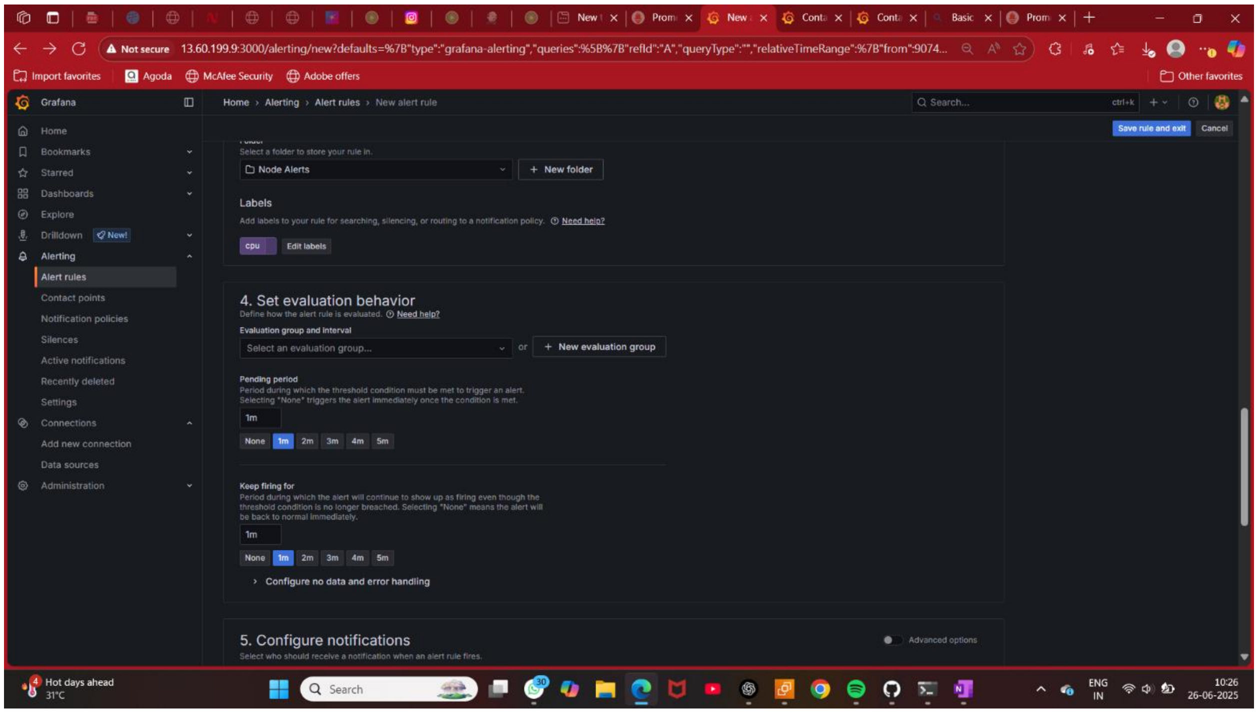
```
scrape_configs:
  - job_name: 'node_exporter'
    static_configs:
      - targets: ['localhost:9100']
```

  - Service enabled and tested via: curl http://localhost:9100/metrics

- Installed **Node Exporter** as a service to collect EC2 metrics (CPU, memory, disk).
- Installed **Grafana** on the same instance:
  - Added Prometheus as a data source.
  - Imported **Node Exporter Full** dashboard from Grafana.com.
  - Created visualizations for:
    - CPU usage
    - Memory usage
    - Disk space
    - Network traffic

Alerts Setup in Grafana

- Created alert for **High CPU usage**:
  - **Query**: 100 - (avg by(instance)(irate(node_cpu_seconds_total{mode="idle"}[5m])) * 100)
  - **Condition**: Above 70% for 1 minute.
  - **Notification**: Configured email alerts.



13.60.199.9:3000/alerting/new?defaults=%7B"type":"grafana-alerting","queries":%5B%7B"refId":"A","queryType":"","relativeTimeRange":%7B"from":9074...

Home › Alerting › Alert rules › New alert rule

Select a folder to store your rule in.
Node Alerts

Labels
Add labels to your rule for searching, silencing, or routing to a notification policy.
cpu

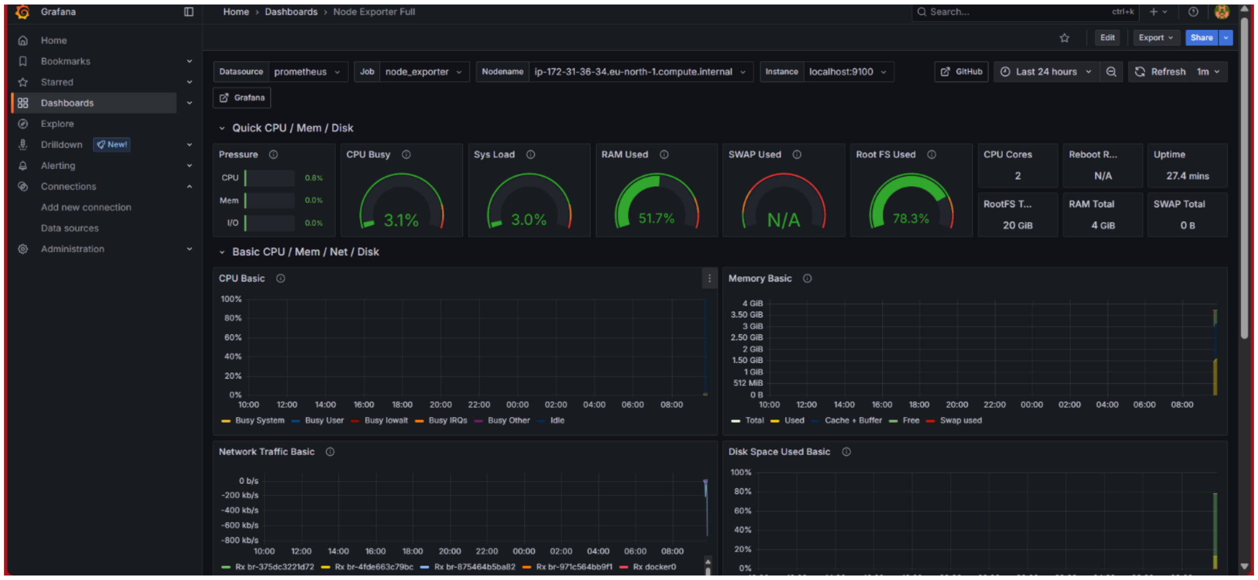
4. Set evaluation behavior
Define how the alert rule is evaluated.
Evaluation group and interval
Select an evaluation group... or + New evaluation group

Pending period
Period during which the threshold condition must be met to trigger an alert.
Selecting "None" triggers the alert immediately once the condition is met.
1m

Keep firing for
Period during which the alert will continue to show up as firing even though the threshold condition is no longer breached. Selecting "None" means the alert will be back to normal immediately.
1m

› Configure no data and error handling

5. Configure notifications
Select who should receive a notification when an alert rule fires.



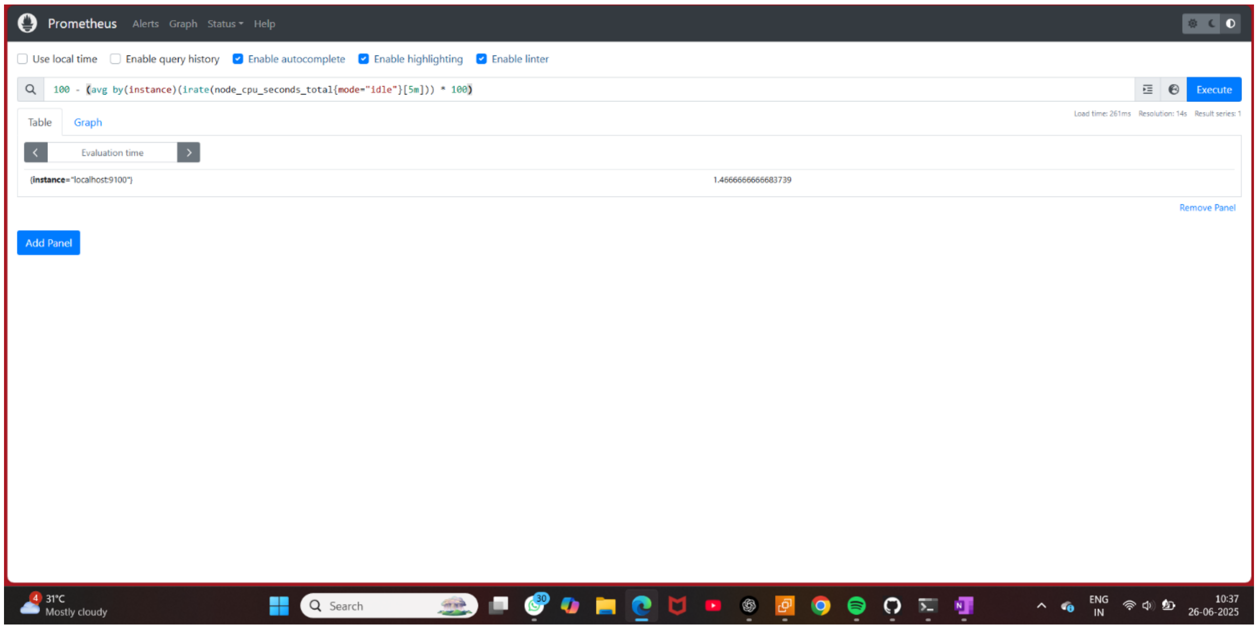
Home › Dashboards › Node Exporter Full

Datasource prometheus | Job node_exporter | Nodename ip-172-31-36-34.eu-north-1.compute.internal | Instance localhost:9100 | Last 24 hours | Refresh 1m

Quick CPU / Mem / Disk

Pressure	CPU Busy	Sys Load	RAM Used	SWAP Used	Root FS Used	CPU Cores	Reboot R...	Uptime
CPU 0.8%, Mem 0.0%, I/O 0.0%	3.1%	3.0%	51.7%	N/A	78.3%	2	N/A	27.4 mins
						RootFS T... 20 GiB	RAM Total 4 GiB	SWAP Total 0 B

Basic CPU / Mem / Net / Disk



Use local time ☐ Enable query history ☐ Enable autocomplete ✓ Enable highlighting ✓ Enable linter ✓

```
100 - (avg by(instance)(irate(node_cpu_seconds_total{mode="idle"}[5m])) * 100)
```

Load time: 261ms Resolution: 14s Result series: 1

```
{instance="localhost:9100"}    1.4666666666683739
```

- Alerts enabled with thresholds

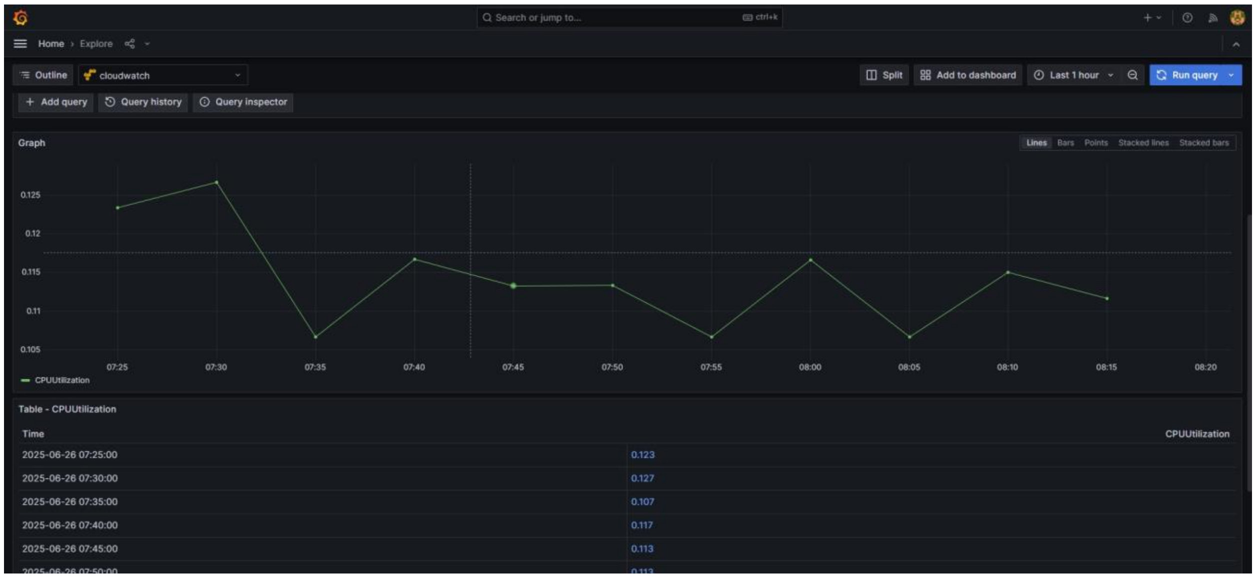


Table - CPUUtilization

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

Author:
This project was fully designed, implemented, and documented by **DHANA LAKSHMI M** as part of the **TechForce Cloud Migration** initiative.

Repository:
🗀 GitHub Repo – https://github.com/dhanalakshmim-eng/Cloud-Migration-Project2.git

Scope & Ownership:

- All infrastructure setup, including VPC, IAM, EC2 provisioning, and Dockerization, was independently managed.
- CI/CD Pipelines were built from scratch using **Jenkins + Docker**.

- **Security** hardening (IAM roles, SGs, Bastion Host) and **monitoring** (Prometheus, Node Exporter, Grafana) were configured using AWS best practices.
- All phases of the TechForce blueprint were executed on a live AWS environment with real-time validations.

Status:
Successfully deployed and monitored the Kimai Timesheet Application using containerized infrastructure with observability and security layers in place.