RAIN from MeghaAI

Rapid Asset Insights Navigator

Deployment Guide

Feb 2022

Megha AI

Actionable Machine Insights

Table of Contents

Megha AI
Actionable Machine Insights

Introduction

This deployment guide provides step-by-step instructions for deploying MeghaAI on AWS cloud. This guide is for users who want to deploy MeghaAI RAIN (Rapid Actionable Insights Navigator) to enable a digital twin of the plant floor, visualize machine insights, and deploy machine learning models to monitor asset health.

This guide is targeted towards IT and cloud professionals who want to understand the infrastructure requirements of this solution and enable production insights and modeling for end users in their organizations.

Core Capabilities of the Software

- Asset Navigator
- Insights Generator: configurable dashboard with real time data
- AI/ML based anomaly detection, diagnosis, and conditional monitoring
- Manufacturing Data Lake built using Unified Namespace

MeghaAI on AWS

MeghaAI software deployed on AWS provides users a powerful platform to visualize and analyze their machines/assets. This guide demonstrates the workflow for installing the necessary services, accessing the platform, and generating insights.

Cost and Licenses

Cost of the AWS services used while running this deployment is the responsibility of the user. The retail license cost is $100,000 USD per manufacturing plant (100 assets or lower) per annum. For manufacturing site for more 100 assets, contact MeghaAI sales for custom quote.

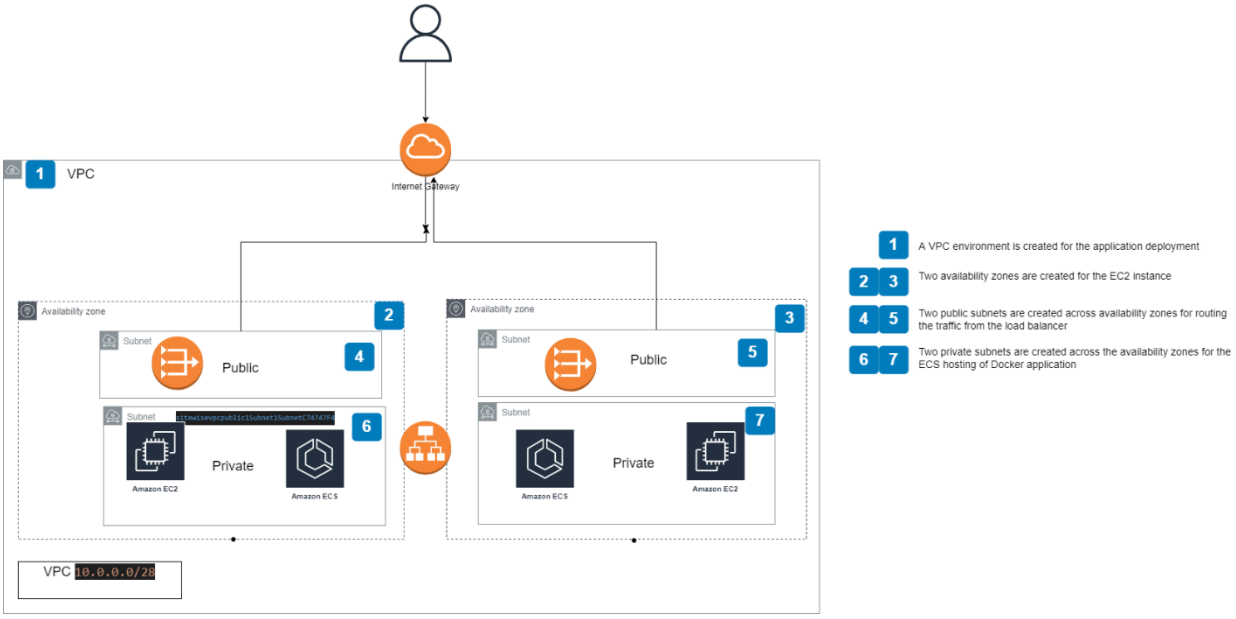
This includes year over year updates of new features, and any customer support for resolving core product issues. Note that the customer may receive discounts depending on the nature and scale of their needs. Software license agreement between MeghaAI and the customer will be established as a contractual agreement with annual invoicing. Once MeghaAI is established in AWS Marketplace, invoicing will be through AWS Marketplace.

The AWS CloudFormation template for this guide includes configuration parameters that you can customize. Some of these settings, such as instance type, will affect the cost of deployment. For cost estimates, see the pricing pages for each AWS service you will be using.

Tip: After you deploy MeghaAI, we recommend that you enable the AWS Cost and Usage Report to track costs associated with deployment. This report delivers billing metrics to an S3 bucket in your account. It provides cost estimates based on usage throughout each month and finalizes the data at the end of the month. For more information about the report, see the AWS documentation.

Deployment and Solution Architecture

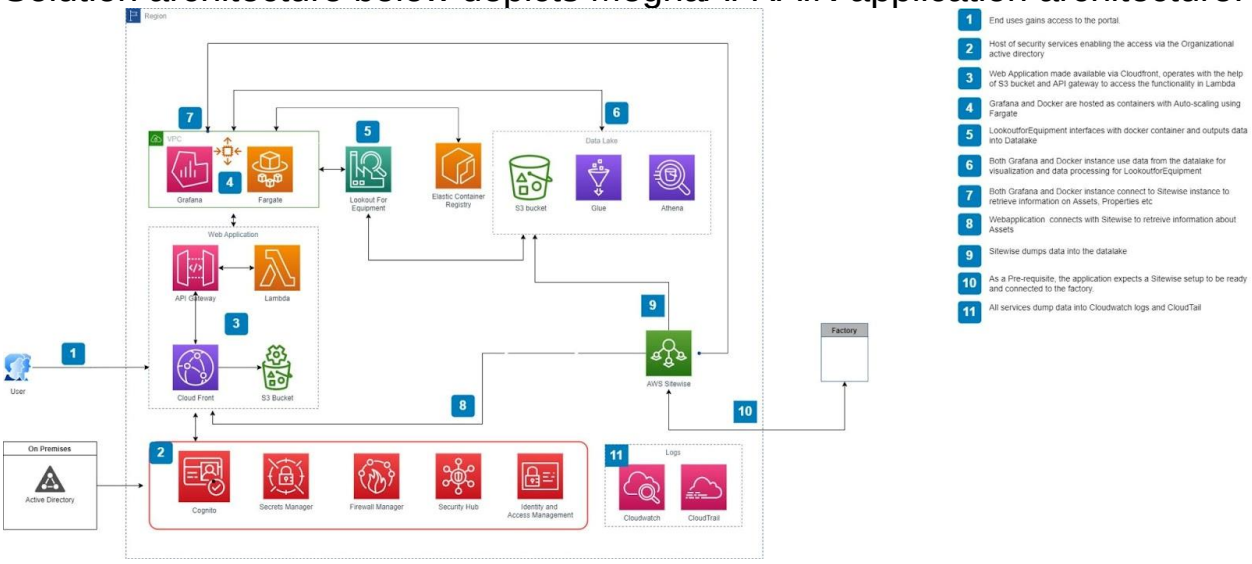
Deploying this Quick Start for a new virtual private cloud (VPC) with default parameters builds the following environment in the AWS Cloud.



Deployment components of the architecture:

- A Virtual Private Cloud (VPC) configured with public subnets, to provide you with your own virtual network on AWS.
- Internet gateway
- In the public subnets:
    - Managed NAT gateways to allow outbound internet access for resources in the private subnets.
- In the private subnets:
    - Grafana Server deployed on an Amazon EC2 instance within your VPC with appropriate security permissions.

Solution architecture below depicts MeghaAI RAIN application architecture:



Deployment Planning

This Quick Start assumes familiarity with a basic understanding of data architecture and visualization. This deployment guide also requires a moderate level of familiarity with AWS services. If you're new to AWS, visit the Getting Started Resource Center and the AWS Training and Certification website for materials and programs that can help you develop the skills to design, deploy, and operate your infrastructure and applications on the AWS Cloud.

AWS Account

If you don't already have an AWS account, create one at https://aws.amazon.com by following the on-screen instructions. Your AWS account is automatically signed up for all AWS services. You are charged only for the services you use.

Megha AI
Actionable Machine Insights

Technical requirements

This deployment is completed via AWS CloudFormation template. The deployment takes about 20 mins on a customer AWS account. Before you launch the MeghaAI RAIN CloudFormation template, your account must be configured as specified in the following table. Otherwise, deployment might fail.

IAM permissions	To start deployment, you must log in to the AWS Management Console with AWS Identity and Access Management (IAM) permissions for the resources and actions the templates will deploy. The Administrator Access managed policy within IAM provides sufficient permissions, although your organization may choose to use a custom policy with more restrictions.
IoT SiteWise	Edge devices and equipment must be configured to send data to Amazon IoT SiteWise.
Region	Ensure your region is set to the same region in which AWS IoT SiteWise is configured
OS Supported	Linux
Sizing	2 EC2 Machines T2 Micro

Security Best Practices

Please review AWS security best practices for additional guidance: https://aws.amazon.com/architecture/security-identity-compliance.

MeghaAI strongly recommends the following best practices.

1. Avoid the use of the "root" account and Ensure MFA is enabled for the "root" account and ensure no root account access key exists.
2. Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password

Megha AI
Actionable Machine Insights

3. Ensure access keys are rotated every 90 days or less
4. Ensure IAM policies that allow full "*:*" administrative privileges are not created
5. Ensure IAM password policies are very secure.
6. AWS Cloud Trail: Ensure CloudTrail is enabled in all regions and Ensure CloudTrail log file validation is enabled
7. Recommend enabling AWS Security Hub.

Deployment Resources

Below table lists public resources, IAM roles, Keys and VPC resources created that will be created by the deployment.

Public Resources

Public Resource	**Purpose**	**Encryption**
S3 Bucket	Hosts Static Website	None
Grafana, EC2 Instance/EBS volume	Data Visualizer	None

IAM Roles

Resource	**Role Name**	**Purpose**
AWS Lambda	AWSLambdaExecutionRole	Used by these lambda function: Save state lambda,mqtt update lambda, generateInferenceData lambda, copy master data lambda,
Grafana	AllowSecretManagerRole	Used by Grafana EC2 instance
Amazon ECS	ECSTaskExecutionRole	Used ECS execution
Amazon ECS	ECSTaskRole	Used by ECS task definition for Flask application
AWS Lambda, AWS IoT SiteWise	IoTSiteWiseExportToS3CoreAccessToFirehoseRole	Used for IoT Topic rule to get data from IoT SiteWise when IoT SiteWise property updates
AWS Lambda, AWS IoTSiteWise	IoTSiteWiseExportToS3MetadataFunctionRole	Used by lambda function to export metadata from IoT SiteWise to Amazon S3

AWS Lambda, AWS IoT SiteWise	IoTSiteWiseExportToS3TransformFunctionRole	This Role allows lambda to store runtime logs for the transform function
Amazon Kinesis Data Firehose	IoTSiteWiseKinesisDeliveryRole	The role allows Kinesis Data Firehose to perform operations on the S3 Bucket, AWS Glue Table and Lambda function
Amazon lookoutequipment	ExecuteL4ERole	This Role is required to execute Lookout For Equipment
Cognito	confirmcognitorole	Used by cognito to confirm AWS Lambda
CloudWatch Logs	sitewiseapiCloudWatchRole	Used for API Amazon CloudWatch logs
SiteWise	sitewiselambdarole	Used by IoT Sitewise get asset Lambda
Lambda	sitewisereportServiceRole	Used by report Lambda
Cognito	sitewisesignin	Used by cognito login Lambda

Secret Keys

Key	**Location**	**Purpose**	**Rotation Policy**
meghaairesource		Authorizes API keys for Grafana	None, can be configured by customer
emcresource			None, can be configured by customer

VPC Resources

SitewiseVPC	VPC	VPC created for the whole app	
sitewisevpc	VPCGatewayAttachment	used for VPC and Internet Gateway attachment	
ECSSecurityGroup	SecurityGroup	used by ECS Service	
ECSSecurityGroupIngressFromPublicALB	SecurityGroupIngress	used by ECS Load balancer Security Group	
ECSSecurityGroupIngressFromSelf	SecurityGroupIngress	used by ECS Security Group	
PublicLoadBalancerSecurityGroup	SecurityGroup	It is a security Group used by Application Load Balancer	

Load Balancers

Resource	Name	Purpose
GrafanaElasticLoadBalancer	ElasticLoadBalancingV2	This load balancer is used for Grafana EC2 instance
GrafanaALBTargetGroup	TargetGroup	This target is used by GrafanaElasticLoadBalancer
grafanaALBListener	Load balancer Listener	This Listener is used by GrafanaElasticLoadBalancer
GrafanaAutoScalingGroup	AutoScalingGroup	This Autoscaling group is used for Grafana EC2 instance
FlaskElasticLoadBalancer	ElasticLoadBalancingV2	This load balancer is used for Flask App
FlaskALBTargetGroupPublic	TargetGroup	This target is used by FlaskElasticLoadBalancer
FlaskALBTargetGroup	TargetGroup	This target is used by FlaskElasticLoadBalancer
FlaskALBListener	Load balancer Listener	This Listener is used by FlaskElasticLoadBalancer
LoadBalancerRule	ListenerRule	This Listener Rule is used by FlaskElasticLoadBalancer

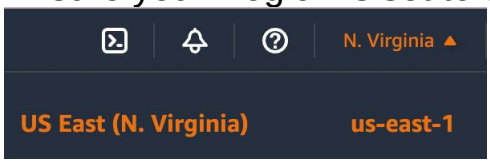
CloudFront Distribution

Resource	Name	Purpose
SitewiseWebCloudFrontDistribution	CloudFront	website hosting to https
GrafanaCloudFrontDistribution	CloudFront	ec2 machine http to https
FlaskCloudFrontDistribution	CloudFront	flap app http to https

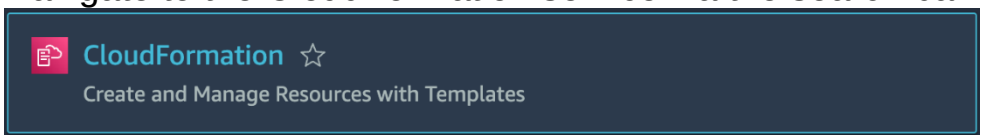
Deployment Steps

1. Sign in to your AWS account on the AWS console with an IAM user role that has the necessary permissions. For details, see Planning Deployment earlier in this guide.

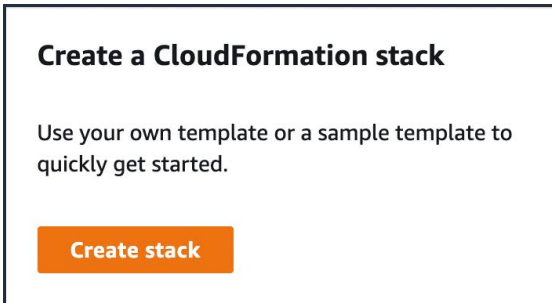
2. Ensure your Region is set to us-east-1



3. Navigate to the CloudFormation service via the search bar



4. Select “Create Stack”

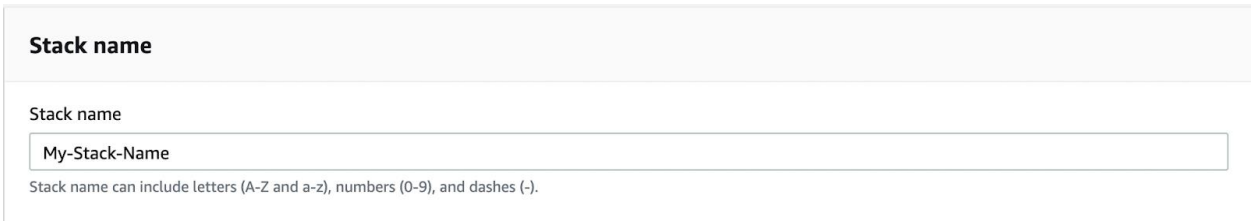


5.Under “**Prerequisite - Prepare template**” select “Template is ready”

6. Under “**Specify template**” select “Upload a template file” and upload the yaml file provided by Megha AI

7. Enter a **Stack Name** (example: MeghaAI-Customer-Stack)

Megha AI
Actionable Machine Insights



8. Provide a AdminUserId and AdminUserPassword, this will be the login information to your insight's platform

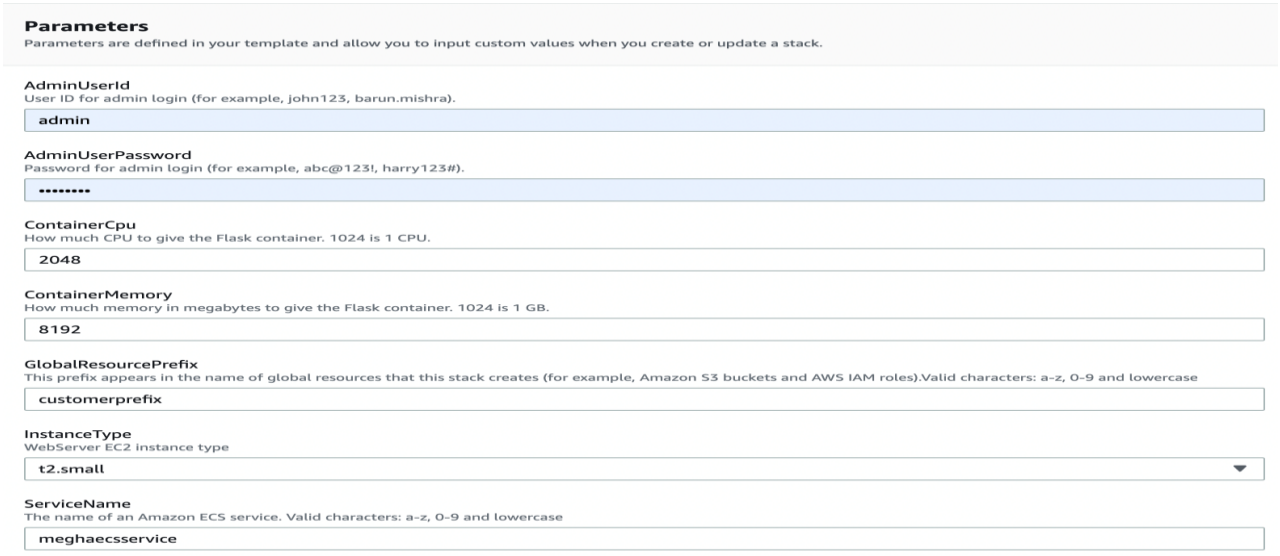
9. Provide a **GlobalResourcePrefix**, this prefix appears in the name of global resources that this stack creates

10. Hit next to navigate to **Configure stack options**

11. Select Next to navigate to **Review**

12. Scroll to the bottom of the page and under **Capabilities** select "*I acknowledge that AWS CloudFormation might create IAM resources with custom names.*"

13. Select "**Create Stack**", at this stage the AWS services associated with Megha AI will begin to spin up- this process takes around 5-10 minutes:



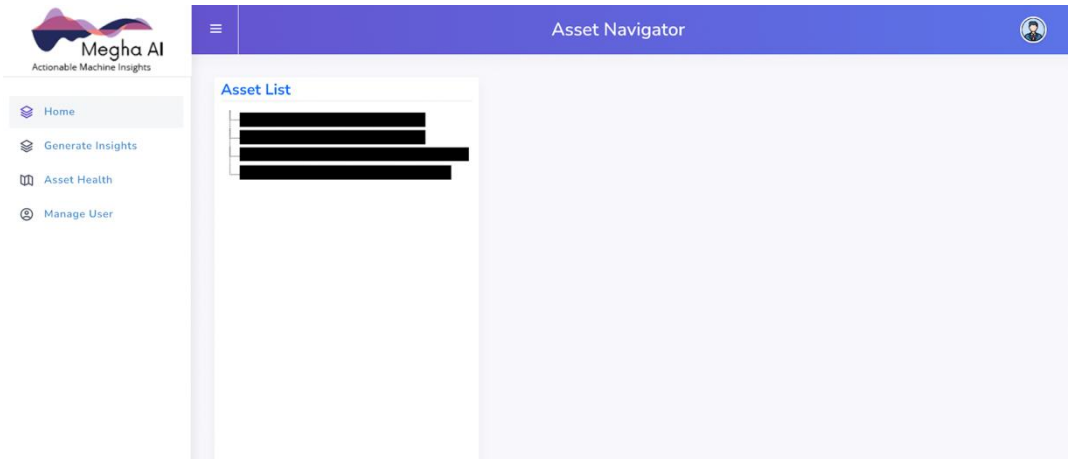
Megha AI
Actionable Machine Insights

Deployment Validation

1. Once your stack status is “CREATE_COMPLETE”, you are ready to view machine insights

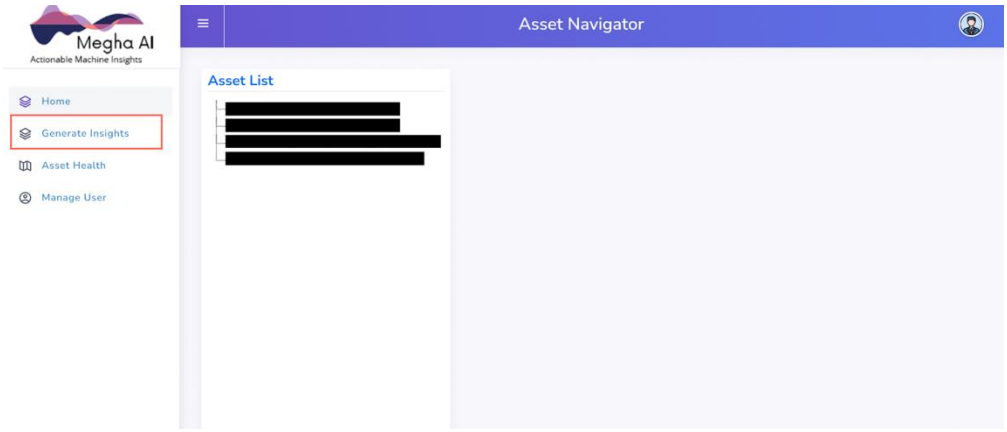
2. To arrive at your platform website, select “Outputs” under the created Stack and select the link associated with the key “websiteURL”

3. You should now be able to view the asset navigator, and all assets connected in AWS IoT Sitewise will appear in real time.



4. To create insights, select “Generate Insights” in the left side menu

Megha AI
Actionable Machine Insights



5. Select your assets, properties, and graphs and start building your dashboard.

6. To access the Grafana dashboard directly, select “Outputs” under the created Stack and select the link associated with the key “grafanaUrl”

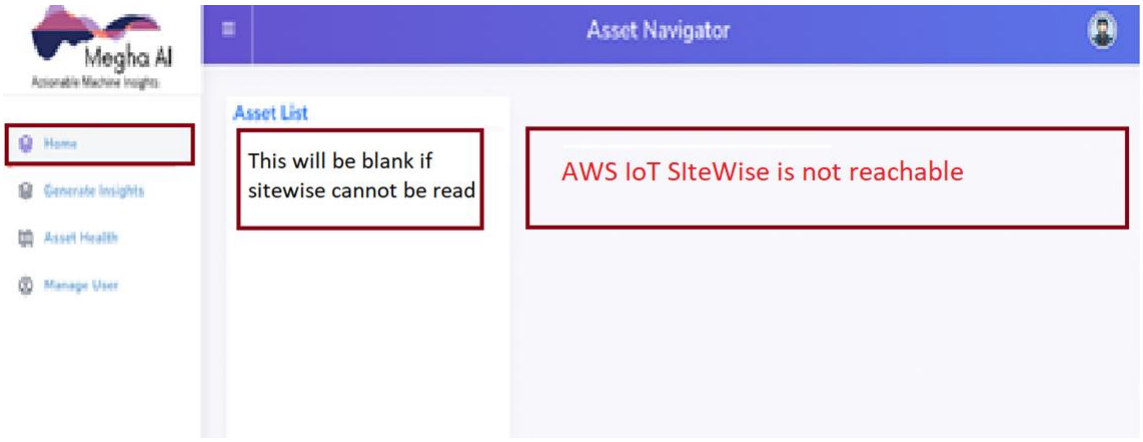
Troubleshooting

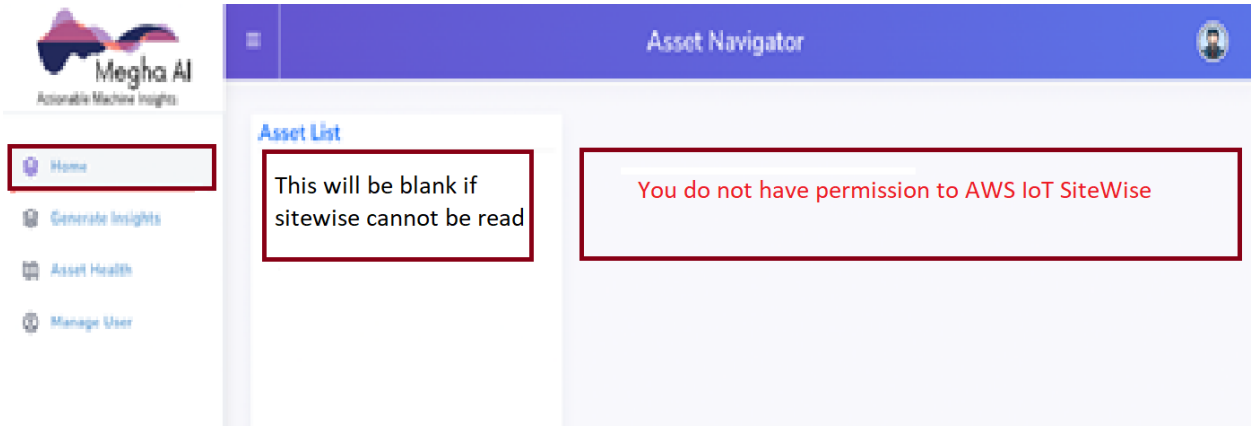
AWS CloudFormation stack CREATE_FAILED error – Recommend to delete and re-launch of the MeghaAI CloudFormation template. You can refer to AWS CloudFormation troubleshooting documentation for more information: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/troubleshooting.html

If you encounter problems related to dependencies, the following section will help you to troubleshoot.

Megha AI
Actionable Machine Insights

- AWS IoT SiteWise dependency
  - If the AWS IoT SiteWise is not reachable either because it is down or user does not have the permission.
    - Home Tab/Generate Insights / Assets Health pages show following appropriate messages. Further steps needed to resolve this. Example of Home tab is shown below
      - AWS IoT SiteWise is not reachable.
      - You do not have permission to access the AWS IoT SiteWise .





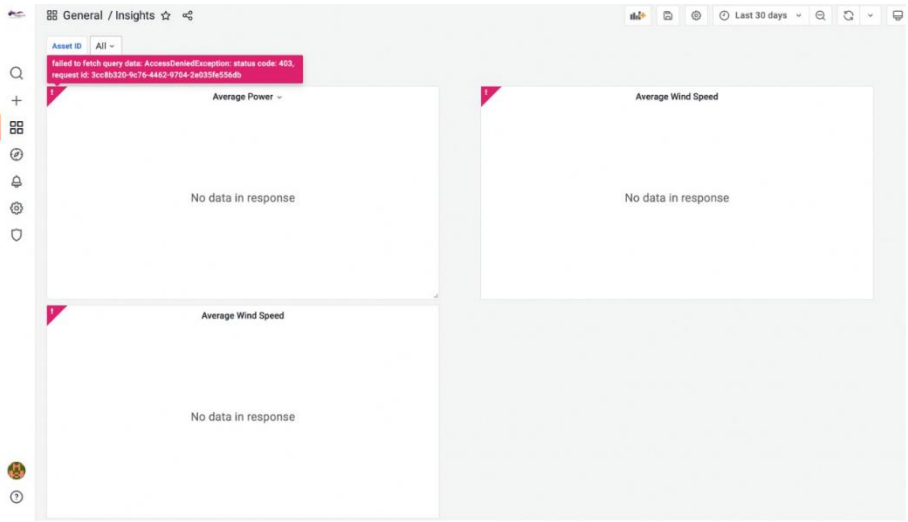
- Amazon Lookout for Equipment dependency

Megha AI
Actionable Machine Insights

- If the Amazon Lookout for Equipment service is not reachable because it is down or user does not have the permission. Further steps needed to resolve this.
  - Assets Health pages show following appropriate messages.
    - Amazon Lookout for Equipment is not reachable.
    - You do not have permission to access the Amazon Lookout for Equipment.

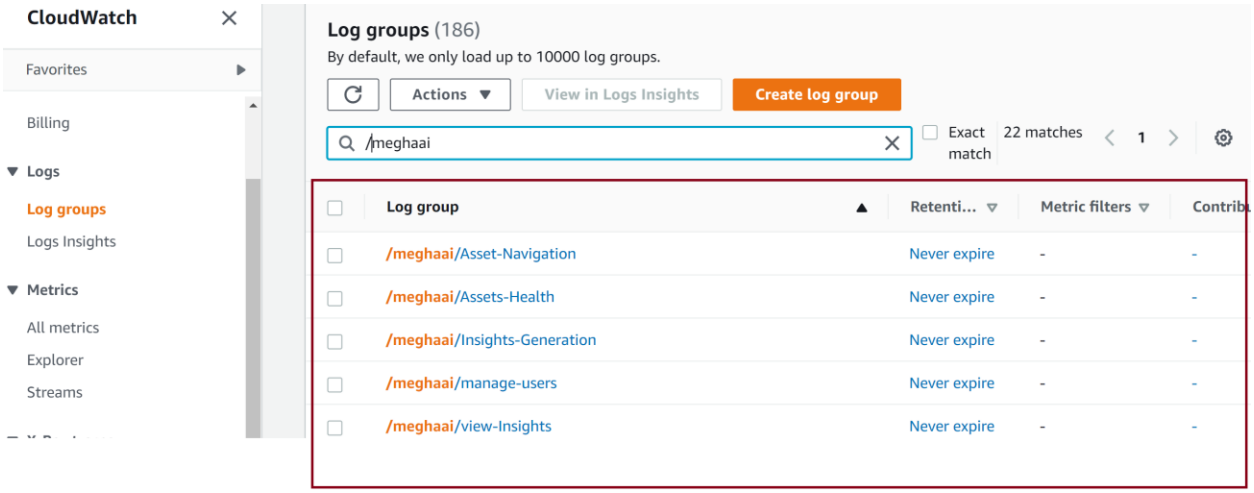
- Grafana Server dependency
  - When Grafana server is down, the site cannot be reached the following message will appear on the grafana URL.

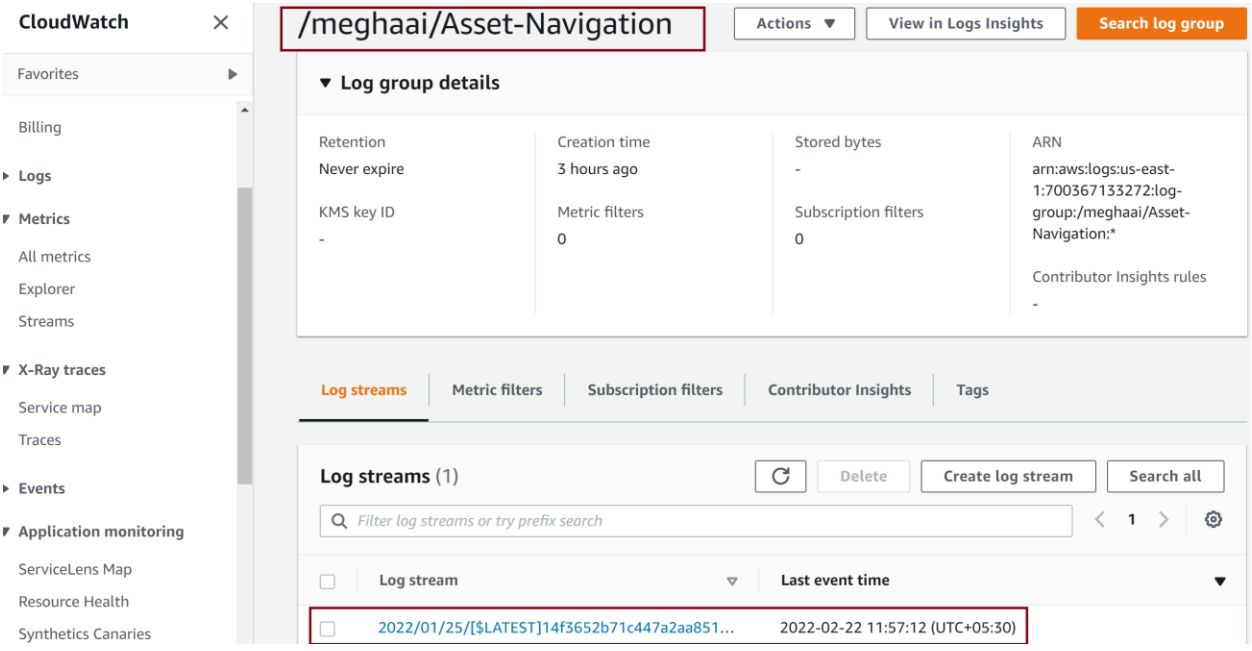
  - AWS IoT SiteWise dependency at Grafana UI
    - If AWS IoT SiteWise is not reachable either because it is down or user does not have the permission. Further steps needed to resolve this. Appropriate message appears, example of “Access denied from permission is shown below”.



- Additional troubleshooting using Analyze Cloud Watch Logs for Failures

  The application creates and logs following log groups in the aws cloud watch service for the purpose of additional monitoring, generating insights and dashboards.





Megha AI
Actionable Machine Insights

Health Check

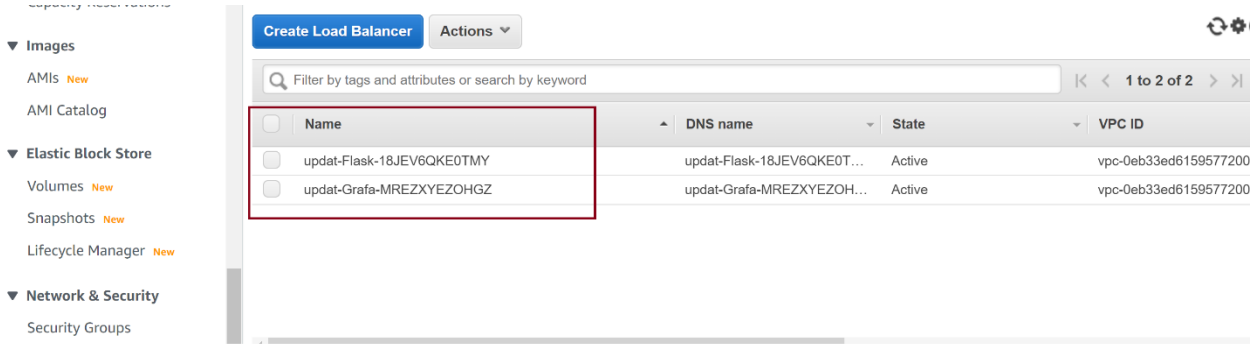
The health of the application is determined by the health of the different AWS services making up the application. MeghaAI product depends on the availability and performance of different AWS services as depicted in the architecture diagrams.

The application creates two load balancers, one for the Grafana server and one for the flask server. These two load balances have naming conventions as shown below , GlobalPrefixName is name of the parameter in cloud formation template and is given by the user.

‘GlobalPrefixName-grafana-*’

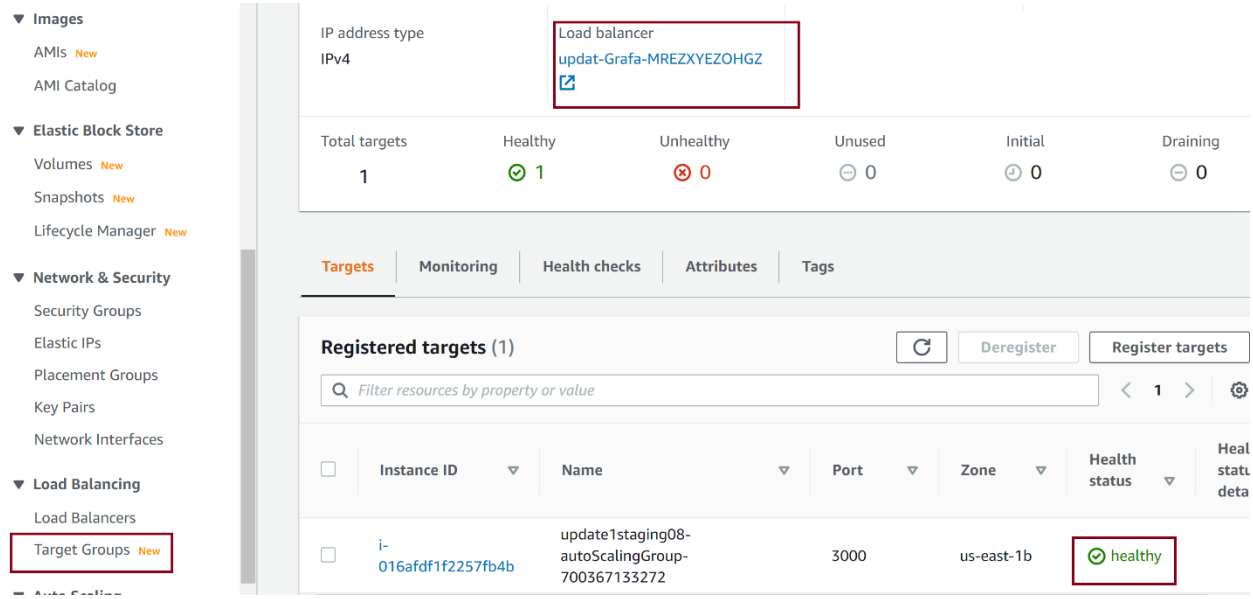
‘GlobalPrefixName-flask-*’

Example is shown below with GlobalPrefixName = update

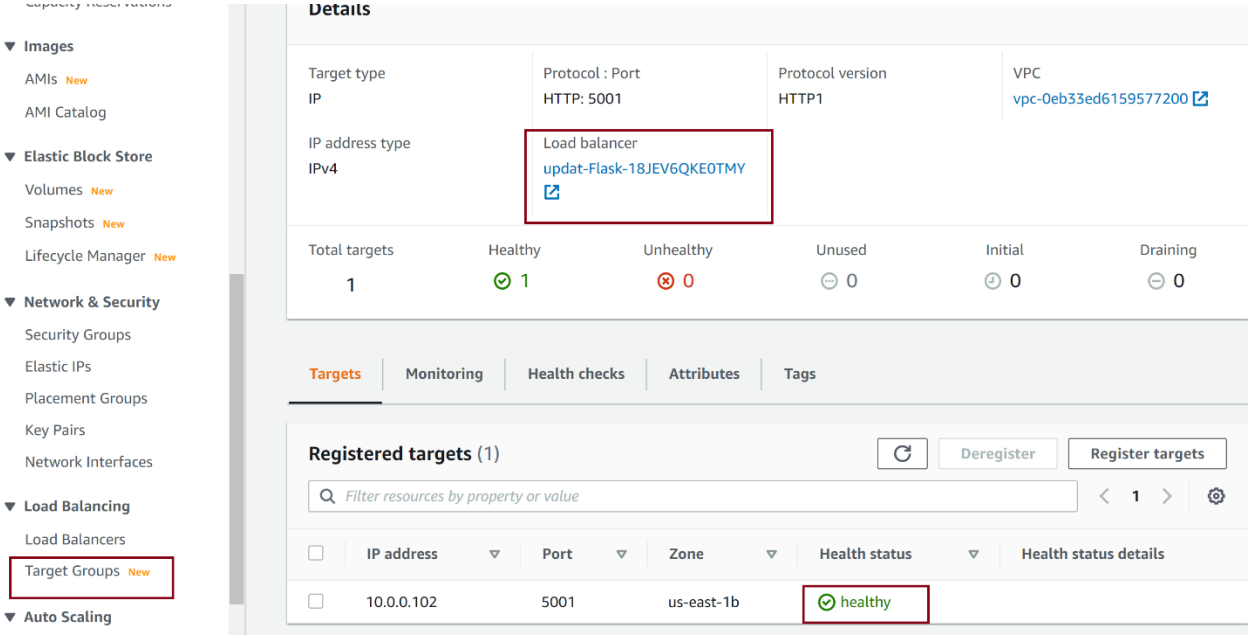


Megha AI
Actionable Machine Insights

The corresponding target groups for the Grafana load balancer is shown below

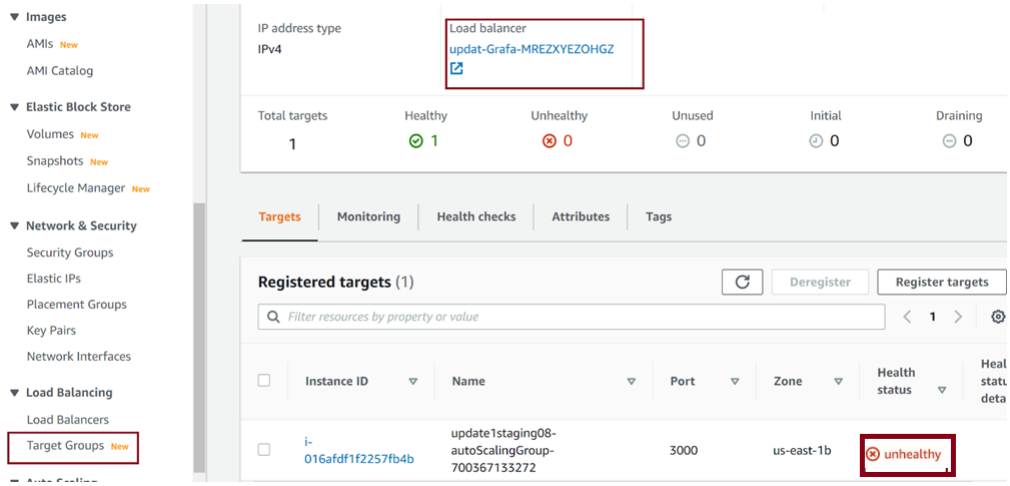


The corresponding target groups for the flask load balancer is shown below

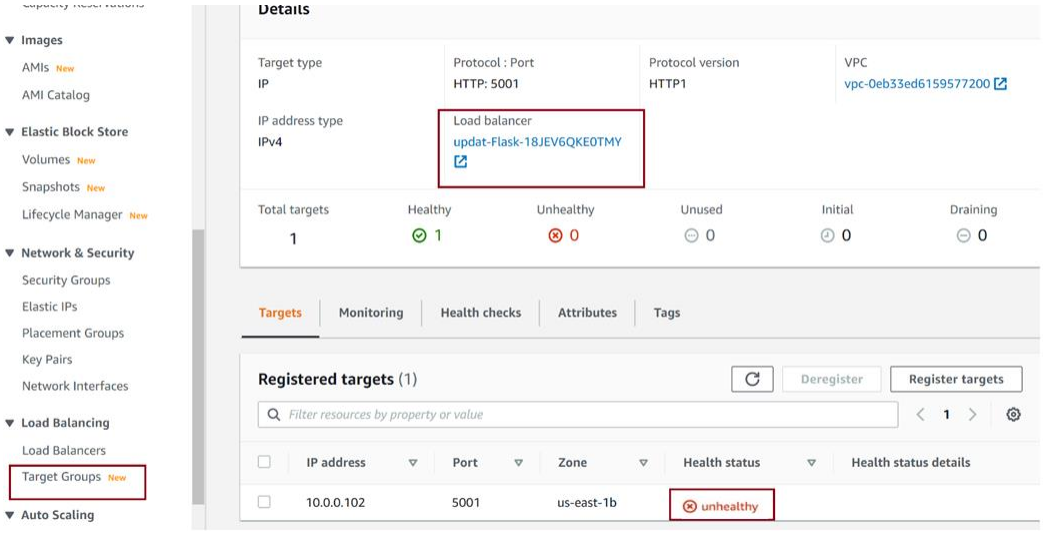


In normal working conditions, these will appear as ‘healthy’ , implying that both Grafana server and the Amazon Lookout for Equipment are working normally.

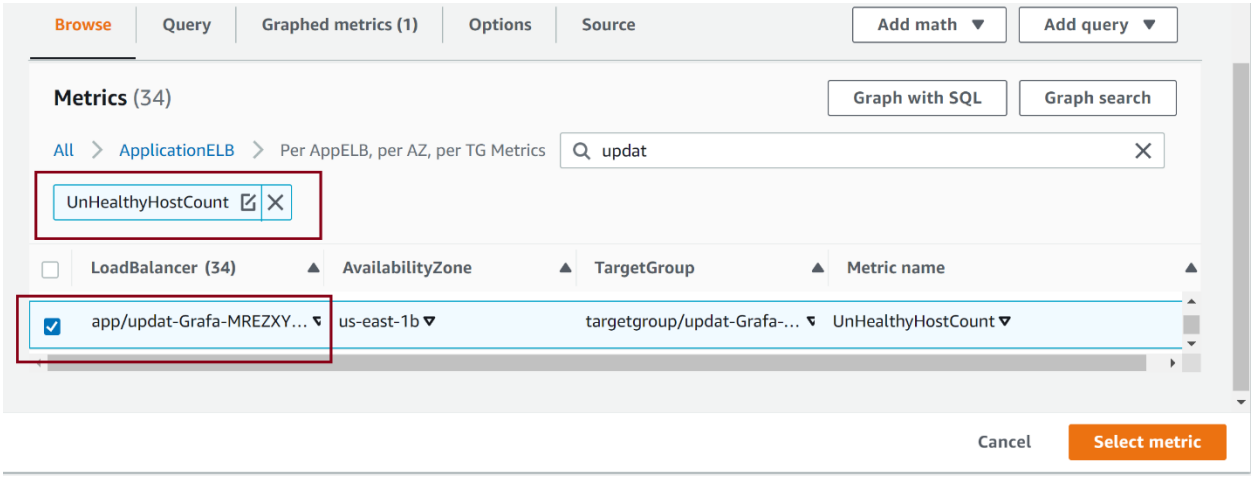
- If Grafana service is down, the target group will show server as ‘unhealthy’ as shown below in the picture. Since the auto scaling feature is ON, the failure will automatically spins up stand by Grafana server.



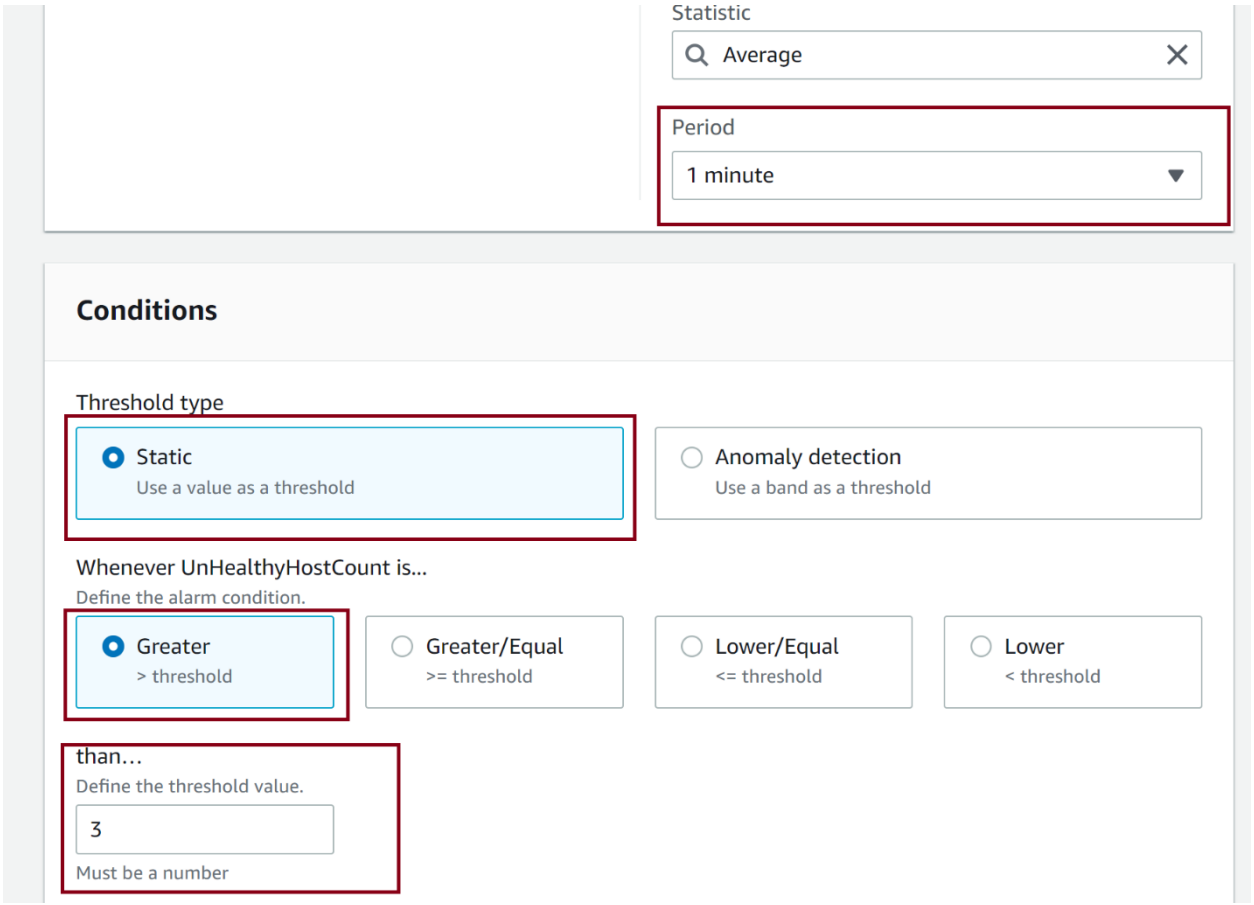
- If flask service is down, the target group will show server as ‘unhealthy’ as shown below in the picture. Since the auto scaling feature is ON, the failure will automatically spins up stand by flask service.



- If the service is found unhealthy, the alarm on UnhealthyHostCount metric provided by ELB trigger the SNS email notification to the users. The details for Grafana server configuration as an illustration are provided below in the screen shots.



Megha AI
Actionable Machine Insights



Alarm state trigger

Define the alarm state that will trigger this action.

Remove

- (●) In alarm — The metric or expression is outside of the defined threshold.
- ( ) OK — The metric or expression is within the defined threshold.
- ( ) Insufficient data — The alarm has just started or not enough data is available.

Select an SNS topic

Define the SNS (Simple Notification Service) topic that will receive the notification.

- ( ) Select an existing SNS topic
- (●) Create new topic
- ( ) Use topic ARN

Create a new topic…

The topic name must be unique.

Meghaai-Unhealthy-Grafana

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (_).

Email endpoints that will receive the notification…

Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

Krishan@meghaai.com

user1@example.com, user2@example.com

Add name and description

Name and description

Alarm name

Grafana-Server-Meghaai

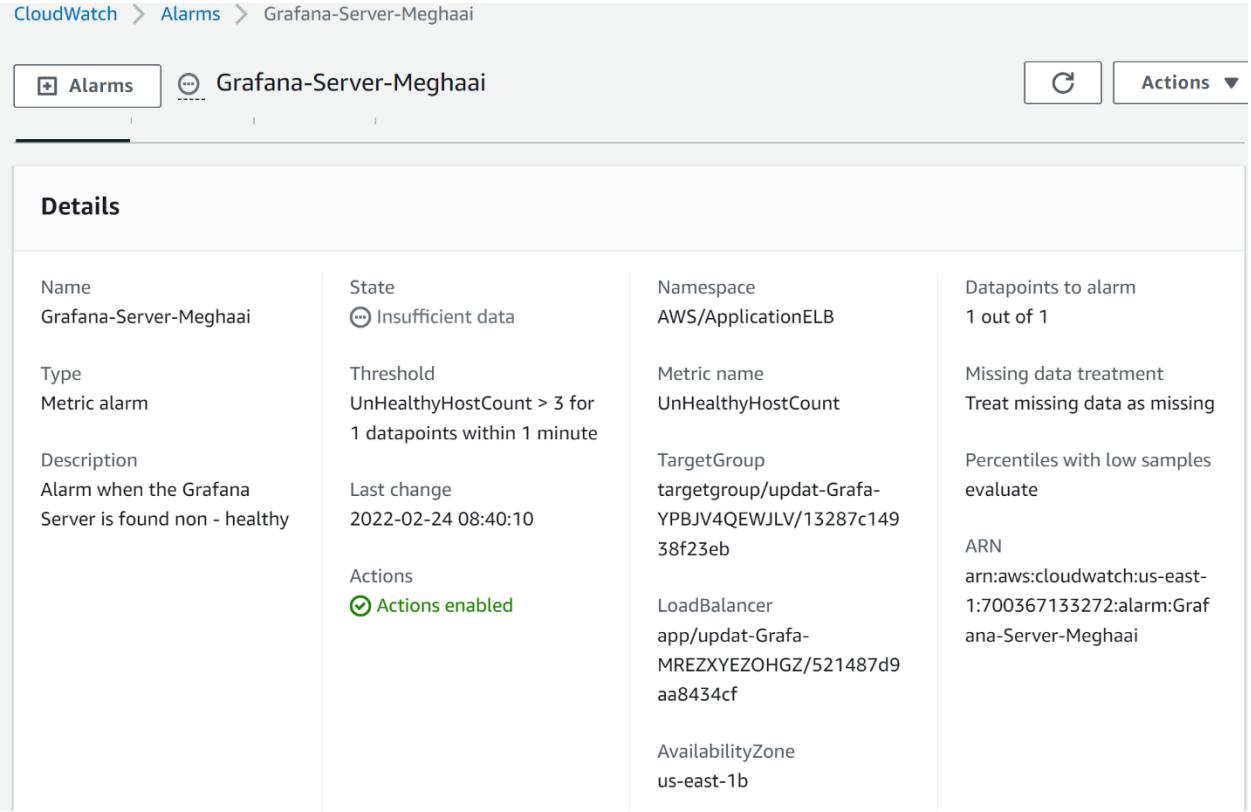
Alarm description - *optional*

Alarm when the Grafana Server is found non - healthy

Up to 1024 characters (52/1024)

Megha AI

Actionable Machine Insights



Backup and Recovery

MeghaAI recommends enabling AWS Backup, which provides an ideal solution for implementing standard backup plans for your AWS resources in an AWS account. Because AWS Backup supports multiple AWS resource types, it makes it easier to maintain and implement a backup strategy.

Maintenance

On faults the default solution is a reinstallation of the software, which includes deletion of the CloudFormation template and reinstallation (see

deployment guide for details). However, the customer should contact the MeghaAI support line before attempting any action.

On accidental deletion or removal of any component and/or AWS services, the software may lose some or all capabilities. Customers can contact the MeghaAI support for a copy of the CloudFormation template for reinstallation. MeghaAI software updates are run by using CloudFormation template which in turns updates both the server-side software and the web application.

Support

If you need additional support from MeghaAI, please contact support@meghaai.com.

Note: At this point of time we have only one level of support. We plan to upgrade to multiple levels in future.