Discovery Phase Report

KEMCO WATER SYSTEMS

TECH P3P
SOLUTIONS

Table of Contents

Objective

1. Develop cloud connectivity architecture to support data collection from KEMCO water heating systems.
2. Recommend infrastructure setup and components required to securely transmit field data from programmable logic controllers which in turn can be utilized for visualization, automated reporting, and event/ alarm notification.
3. Provide detail on software tools, hardware, and configuration of cloud services to meet above requirements.

Existing System Details

The current heater systems utilize a programmable logic controller (PLC) to aggregate field instrumentation data, provide process control, compute certain system parameters, and set alarm conditions. A common PLC used is the Allen Bradley Micro Logix PLC which utilizes ETHERNET/IP to communicate with external systems. As an alternate to the Allen Bradley PLC, KEMCO could also utilize the Schneider M251 PLC and it communicates over MODBUS TCP.

The heater systems can also include a human machine interface (HMI) which communicates with the PLC using the ETHERNET/IP or Modbus protocol. The CMT series HMI's from Maplesoft is one of the HMI's currently being used with the system.

MAPLE SYSTEMS HMI Specification

A heater system that utilizes a Maple systems HMI for local visualization and monitoring of the heater system parameters also provides capability for the PLC data to be retrieved and sent to the cloud if an internet connection is available. Below highlighted are some of features of the maple systems HMI that were reviewed during the discovery phase that make it a good option for communication over MQTT protocol.

HMI CONNECTION SPECIFICATIONS

- DUAL ETHERNET, WIFI Connectivity Supported
- Since there is NO GPRS/GSM Supported additional edge Gateway, router or connection to the internet is required.

HMI MQTT SPECIFICATIONS

a. HMI supports MQTT v3.1 & v3.1.1
b. Topics can be user-defined
c. Data can be published at a fixed interval or event based
d. Data compression & Retention available
e. All three levels of QOS available
f. JSON & Raw Data format supported
g. Username & Password Authentication supported
h. X.509 Certificate Management supported
i. Data encrypted supported using mTLS/SSL

Architecture

The primary focus areas around which the architecture is designed are

- Ease of use
- Standard Off the shelf components
- Security
- Scalability

CONNECTIVITY PROTOCOL

Basis the use cases and the existing hardware the MQTT protocol is recommend for communication of the field data to the cloud. MQTT is the most used messaging protocol for the Internet of Things (IoT). MQTT stands for MQ Telemetry Transport. The protocol is a set of rules that defines how IoT devices can publish and subscribe to data over the Internet. MQTT is used for messaging and data exchange between IoT and industrial IoT (IIoT) devices, such as embedded devices, sensors, industrial PLCs, etc. The protocol is event driven and connects devices using the publish /subscribe (Pub/Sub) pattern. The sender (Publisher) and the receiver (Subscriber) communicate via Topics and are decoupled from each other. The connection between them is handled by an MQTT broker. An **MQTT broker** is a server that receives all messages from the MQTT publisher / clients and then routes the messages to the appropriate subscribers. The broker is at the heart of any publish/subscribe protocol. The MQTT broker filters all incoming messages and distributes them correctly to the Subscribers.

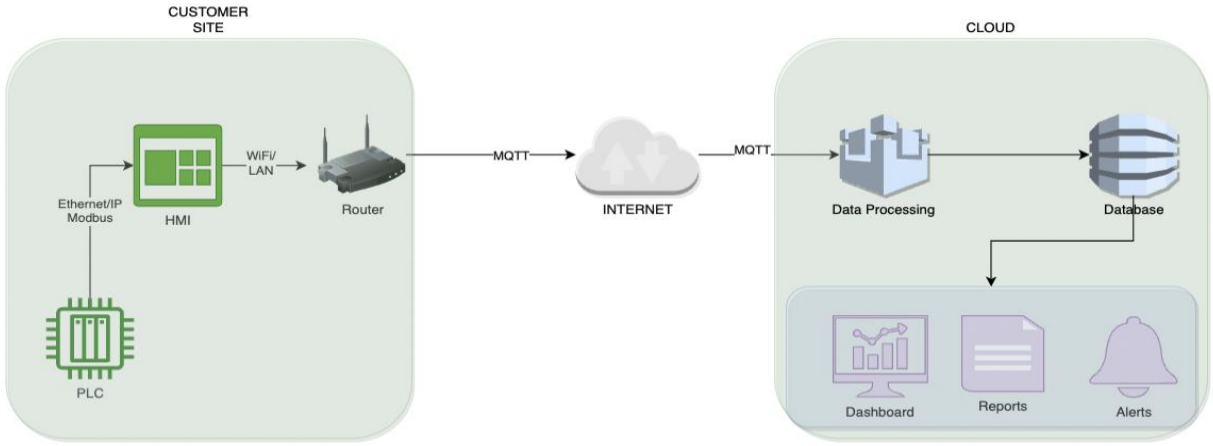


Figure 1: Field to Cloud Connectivity using MQTT

CLOUD SERVICE PROVIDER

A cloud service provider is a third-party company offering a cloud-based platform, infrastructure, application and storage services. Much like a homeowner would pay for a utility such as electricity or gas, companies must pay only for cloud services they use. Microsoft, Google, Amazon and IBM are the top 4 cloud service providers.

Based on the requirements, cost and available feature set Amazon web services (AWS) is the recommended cloud provider that should be used. Amazon Web Services (AWS) is currently the top cloud service provider, in terms of popularity as well as usage, accounting for 32% of the cloud share market.

AWS IOT and Services

AWS provides a vast spectrum of services that help manage the infrastructure, connectivity and applications required to meet the business needs of this project. AWS IoT provides the cloud services that connect your IoT devices to other devices and AWS cloud services. AWS IoT provides device software that can help you integrate your IoT devices into AWS IoT-based solutions. If your devices can connect to AWS IoT, AWS IoT can connect them to the cloud services that AWS provides. The three key areas of services that we need to focus on are Communication, Computing and Storage.

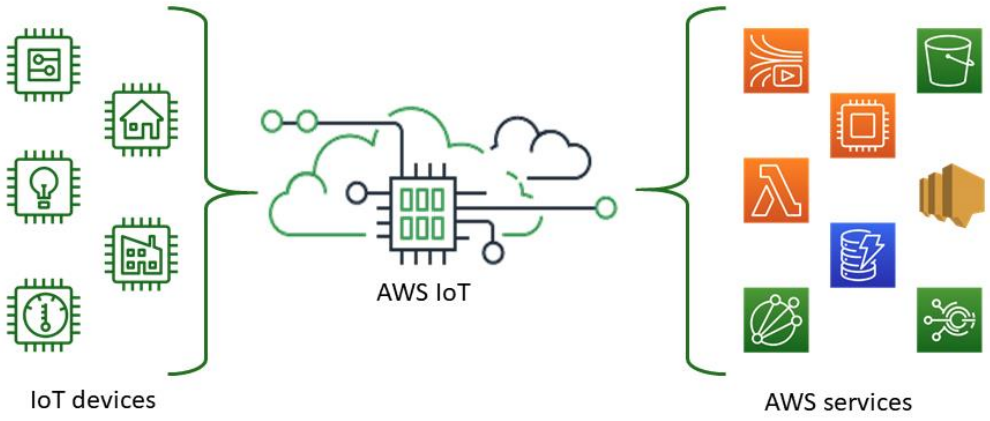


Figure 2 : AWS Services

AWS IoT CORE

AWS IoT Core is a managed Cloud service which securely connects with Industrial assets using communication protocols like MQTT, HTTPS, MQTT over WSS and LoraWAN. It has capability to connect billions of devices & capable of routing trillions of messages to other IoT services. AWS IoT Core policies allows access to control the device connectivity, send & receive MQTT Messages. Below are the specifications within which we have designed the architecture for this project.

- AWS IoT supports MQTT v3.1.1 specification
- No restriction on MQTT topics definition
- Quality of Service (QOS) 0 & 1 only supported
- JSON Payload Format Supported
- Device & Server authentication based on X.509 Certificate Mechanism
- Communication data are encrypted using TLS1.2 version

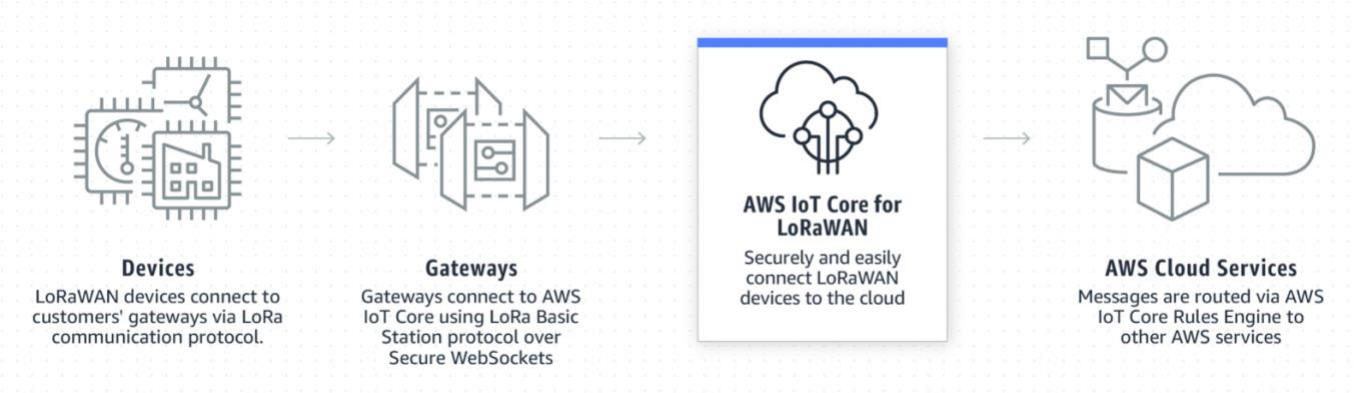


Figure 3 : Sample connection flow between devices and AWS Cloud

AWS IoT Rules

AWS IoT Rules is part of AWS IoT core which creates the ability for the devices to interact with AWS Services. It is the main component of the AWS IoT ecosystem which analyzes actions based on the MQTT topic stream.

AWS IoT rules help filter data received from a device and can be routed to multiple service as listed below

- Write data to Amazon DB like Timestream, DynamoDB
- Send Notification to Users using Amazon SNS
- Send Messages for IoT Analytics etc.

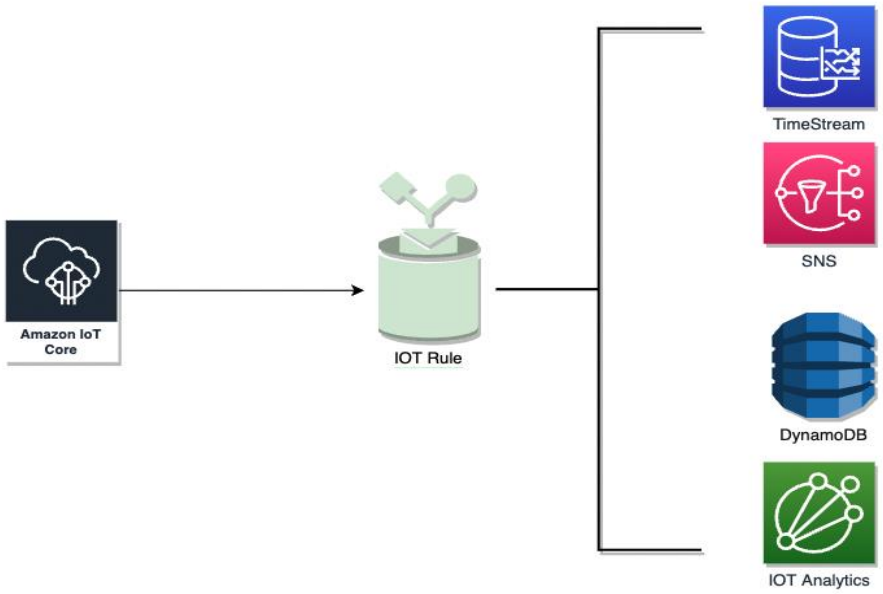


Figure 4: IoT Rules Data flow diagram

Recommendation on IoT Rules

- The topic structure that originates from the PLC or HMI or gateway need to be well defined to comply with AWS IoT rules structure.
- The topic (Decimal) rule function needs to be utilized for extracting the sub topic segments as shown in the example below.

Example:

```
dt/<application-prefix>/<context>/<thing-name>/<dt-type>
```

AWS TimeStream DB

AWS Timestream is a serverless time series database which has fast retrieval speed & scalable storage capacity. It has the capability to move old historical data to a cost optimized storage tier based upon user defined policies.

Time Series data is a sequence of data points recorded over a time. Most of the real time changing telemetry data are referred to as Time Series data. The data consists of a timestamp and one or more attributes.

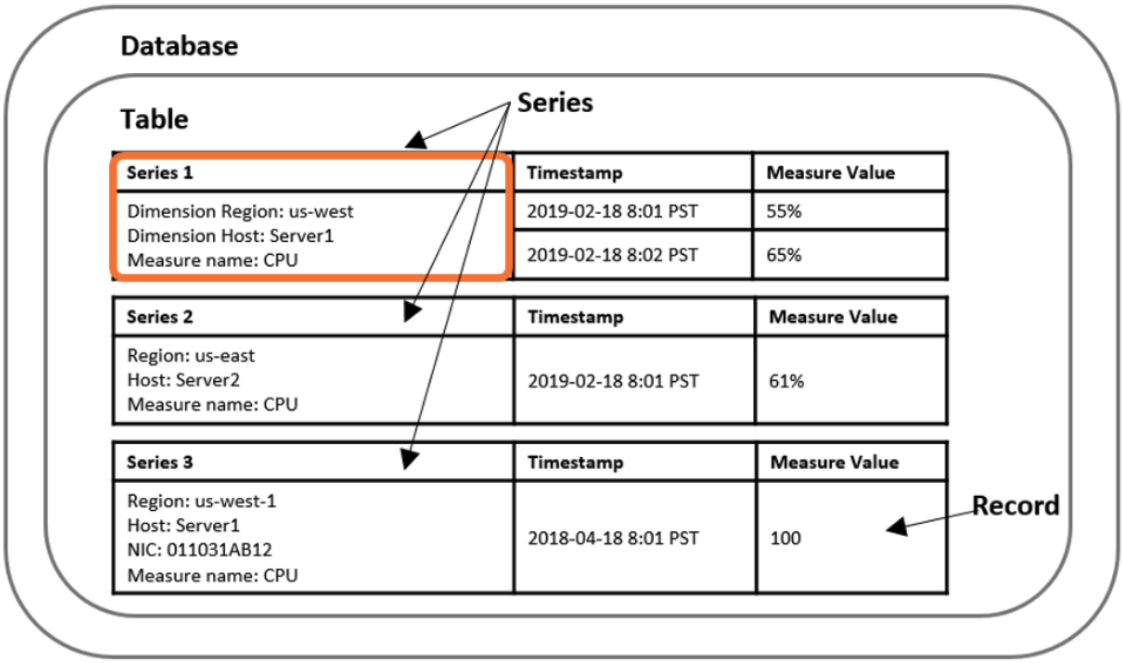


Figure 5 : Timeseries sample configuration

- Database consists of one or more Tables
- Table consists of one or more Series
- Each Series consists of sequence of records
- Each record consists of measured value with Timestamp

Key Benefits

1. Timestream is cheaper as compared to all other aws Database
2. Timestream is well designed for most IoT data and IoT Core
3. Built In Athena like SQL query functionality

4. Auto-scaling capabilities – Automatically scales to handle highspeed real-time data ingestion
5. Time Series data always gets encrypted before storage
6. Data gets auto replicated on 3 different availability zones within a single region for high availability

Supported Integration & Drivers

1. Amazon QuickSight
2. AWS IoT Core
3. Grafana (Open-Source Edition)
4. Database Tools via JDBC (SQL Workbench/J, DataGrip,DBVIsualizer)

FIELD TO CLOUD INTERNET SERVICE

Getting the data to the cloud from the field devices would require internet connectivity. This could be achieved by adding an industrial cellular router. This would require an active cellular connection from a provider such as AT&T or Verizon. Alternate solutions would include tapping into the customers network or connecting to a wi fi access point on site.

Teltonika RUT200 supports 4G LTE, Wi-Fi and Ethernet ports which provides internet connectivity in the field. It is compact in size and manufactured for industrial environments. It is recommended to use this or an alternate router with a similar feature set.

Figure 6 : Recommended Industrial Cellular Router

Key features

- RUT200 supports 4G/3G/2G networks with capabilities to blacklist operators
- It supports Wireless Access Points & Station Mode upto 50 users
- It has inbuilt firewall which helps to add port forwarding, traffic rules etc
- VPN tunneling supported with multiple protocols like OpenVPN, Ipsec, Stunnel, WireGuard etc
- It has built in capability to communicate with devices over Modbus TCP
- It supports MQTT over SSL to communicate with Cloud Platform

Visualization

There are several visualization tools available that can connect to the AWS infrastructure and display a user defined dashboard. Basis the use cases for KEMCO it is recommended to use Grafana for visualization.

Grafana

Grafana is open-source analytics & interactive visualization web based solution for time series database. It allows an application to query time series data on a time interval. It supports a wide range of plugins with enriched dashboard features such as charts, graphs, and alerts for the web when connected to supported data sources.

Grafana Enterprise Version (recently upgraded to Amazon managed Grafana) comes with some additional features mentioned below :

- Role-based access control to control access with role-based permissions. (Generally Support three role type: Viewer, Editor & Admin)
- Data source permissions to restrict query access to specific teams and users.
- Export dashboard as PDF
- Reporting to generate a PDF report from any dashboard and set up a schedule to have it emailed to whoever you choose.
- Vault integration to manage your configuration or provisioning secrets with Vault.
- Custom branding to customize Grafana from the brand and logo to the footer links.

Plugins

Grafana is a multi-platform open-source analytics and interactive visualization web application. Grafana supports scalable vector graphics (SVG) which can be leveraged for industrial Component representation on the dashboard but needs additional configuration / programing. This is achieved utilizing plugins. As an example the SCADAvis Synoptic panel plugin provides an incredibly powerful SVG Editor that can be used to create free-form graphics animated with Grafana data. This panel plugin allows unleashing the power of SCADA-like graphics in Grafana.

Basic Setup Requirements

Grafana needs an IAM role with Timestream db full access to connect with the database.

Grafana needs to run on an EC2 instance within a public ip for accessibility. Subdomain certificate required

Figure 7 : Sample Dashboard displaying a few trends and heat map

PRICING ESTIMATES

The architecture is designed to be scalable and pay as you go. Below are cost estimates for hardware, communication, data storage and use of cloud services. Assumptions on the amount of data being collected and the frequency of collection have been made and could be adjusted as required.

Total number of Parameters	30 Parameters
Frequency of	Every 10 mins
Total Storage required	8 GB per System
Database Detail	AWS Timeseries
Visualization	1CPU , 1GB RAM

Figure 8 : Estimated data management details per system per month

Industrial Router	$ 250 per system
Monthly Cellular Costs	$20 per month per router
AWS Compute	$ 200 per month per 100 systems
AWS Time Stream DB	$ 200 per month per 100 systems
AWS IoT Core	$200 Per month per 100 systems
Custom Reporting	$200 Per month per 500 reports (if required beyond what Grafana offers)

Figure 9 : Estimated Cost Structure

IMPLEMENTATION

In this section the implementation and configuration detail for the cloud services and visualization tool is described. Regardless of the hardware utilized the cloud services and configuration will remain the same however it is important to make sure that the MQTT topis and payload are configured correctly.

Implementing a system that utilizes a Maplesystems HMI was reviewed in detail and the configuration steps to setup the cloud services and a basic dashboard are provided in this report.

Other hardware configurations such as direct connection of the PLC to the cloud without an HMI or using a different HMI would require proper conversion of the data or PLC tags to MQTT. This can be achieved using the Teltonika RUT200 industrial router for systems that are capable of communicating Modbus RTU/ TCP or Ethernet IP

ARCHITECTURE FOR SYSTEM WITH MAPLESYSTEMS HMI

Systems with a Maple systems HMI are capable of creating Visualization dashboards or screens that are accessible through a web browser on the local network. By connecting a Teltonika RUT200 router to the Maplesoft HMI all the PLC tags written to the HMI can be transported to the cloud for remote data access and enhanced data extraction.

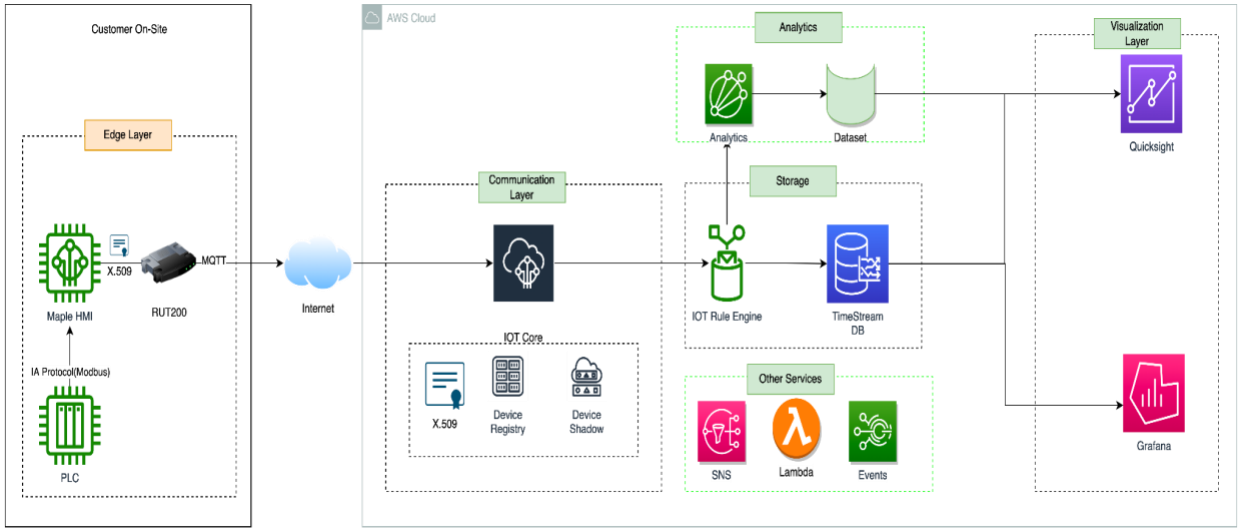


Figure 10 : Architecture for systems with a Maplesystems HMI

CONFIGURING A MAPLE SYSTEM HMI

Maple system HMI supports MQTT which can be leveraged to connect with Cloud IOT platform for data retrieval. It processes data from PLCs and can publish messages to the external MQTT broker, which will handle message delivery to the subscribers. The Advanced and cMT series HMIs support MQTT features & AWS IOT core. Configuration of MQTT on the HMI is done using the Easybuilder Pro 2.0 tool and includes Connection setup, Topic Creation and Payload definition

MQTT TOPICS

In MQTT, the word topic refers to an UTF-8 string that the broker uses to filter messages for each connected client. The topic consists of one or more topic levels. Each topic level is separated by a forward slash (topic level separator).

In comparison to a message queue, MQTT topics are very lightweight. The client does not need to create the desired topic before they publish or subscribe to it. The broker accepts each valid topic without any prior initialization.

The MQTT topic should be 5 parts as shown below. Each represents the meta information about the site from where the telemetry is getting published.

<dt>/<site_id>/<application_id>/telemetry/<device_id>

- dt -> represents data
- Site_id -> represents the site name or unique site id
- application_id -> Source Application like HMI or Gateway
- device_id-> PLC or HMI ID

Example Topic name

dt/site1/HMI/telemetry/PLC001/

MQTT PAYLOAD

The Payload generated from Maple systems HMI has a specific format which cannot be customized using the HMI configuration software. This requires complex IoT rules to be written to filter the required data for proper extraction and database storage.

Sample Payload that originates from the HMI


```
{ "d": {
      "heater_message_1": 1
      "heater_message_2": 2
      "heater_temp_rise": 78.5
      "heater_stack_temperature": 310.8,
      "heater_inlet_temperature": 200.8,
      "heater_outlet_temperature": 320.0
      "heater_stack_approach": 315.
      "hot_water_tank_1_temperature": 75.6
      "hot_water_tank_1_level": 1000.0
      "hot_water_tank_1_header": 800.4
      "hot_water_tank_1_pump_1_speed": 52.0
      "hot_water_tank_1_pump_2_speed": 60.0
      "hot_water_tank_1_header_gpm": 200.0
      “Device_id”: “PLC001”
   },
   "ts": "2017-04-18T17:36:52.501856""}
```

AWS IOT CORE - X.509

Each thing has its unique certificate attached with the policy. The Policy should be set only to Publish since the project POC use case is only to publish data to the cloud. The X509 certificate contains below certs in which the private key needs to be handled & stored securely

- Root CA certificate
- Client Certificate
- Private Key

AWS IOT RULES

IOT rules should be defined in such a way that the message payloads are processed and mapped against each measure available on the payload. Please note it is critical the message payload should have a dimension which is the metadata about the devices. It could be the device_id, site_id or any meta data related to the site. IoT rule should route all filtered & processed message to the AWS Timestream DB

AWS TIMESTREAM

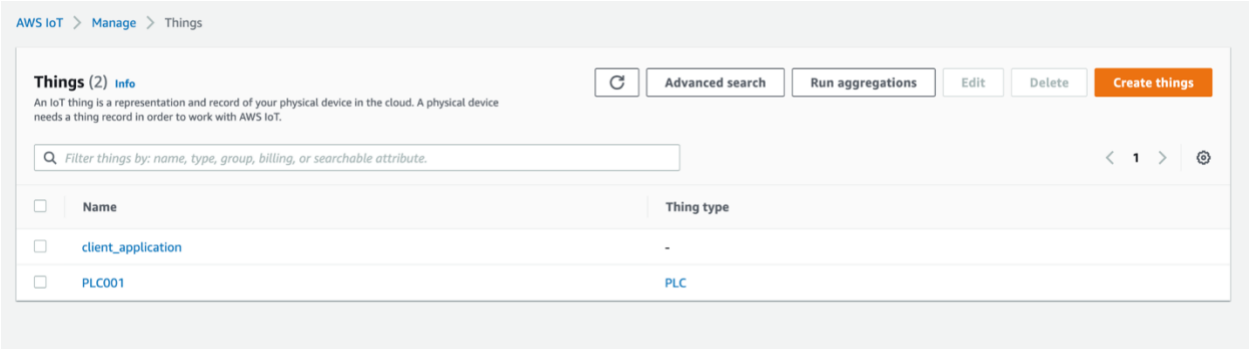
Timestream Databases need to be defined with the retention period for individual tables. There are two types of retention period

- Memory storage retention
- Magnetic storage retention

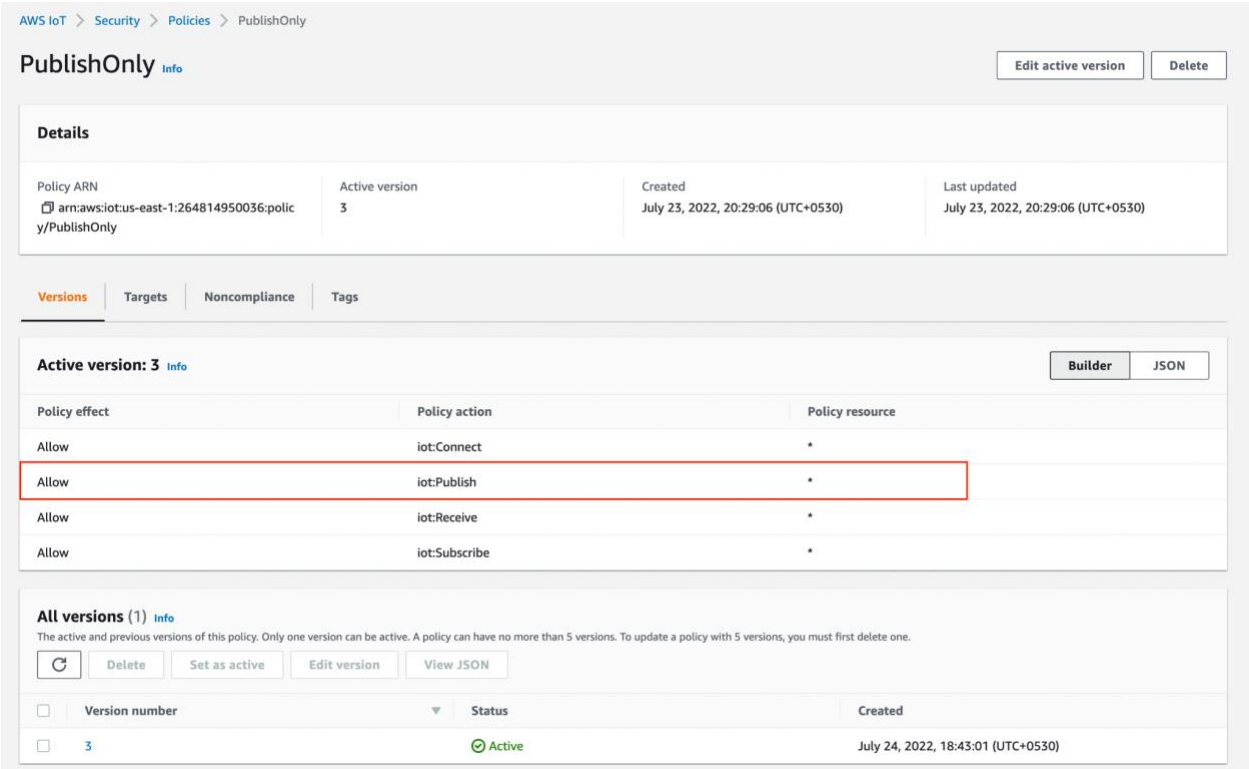
Dimensions and Measure are the key components in storing data as time series on the database. These need to be well defined on the payload as well as on the IOT rules

AWS CONFIGURATION

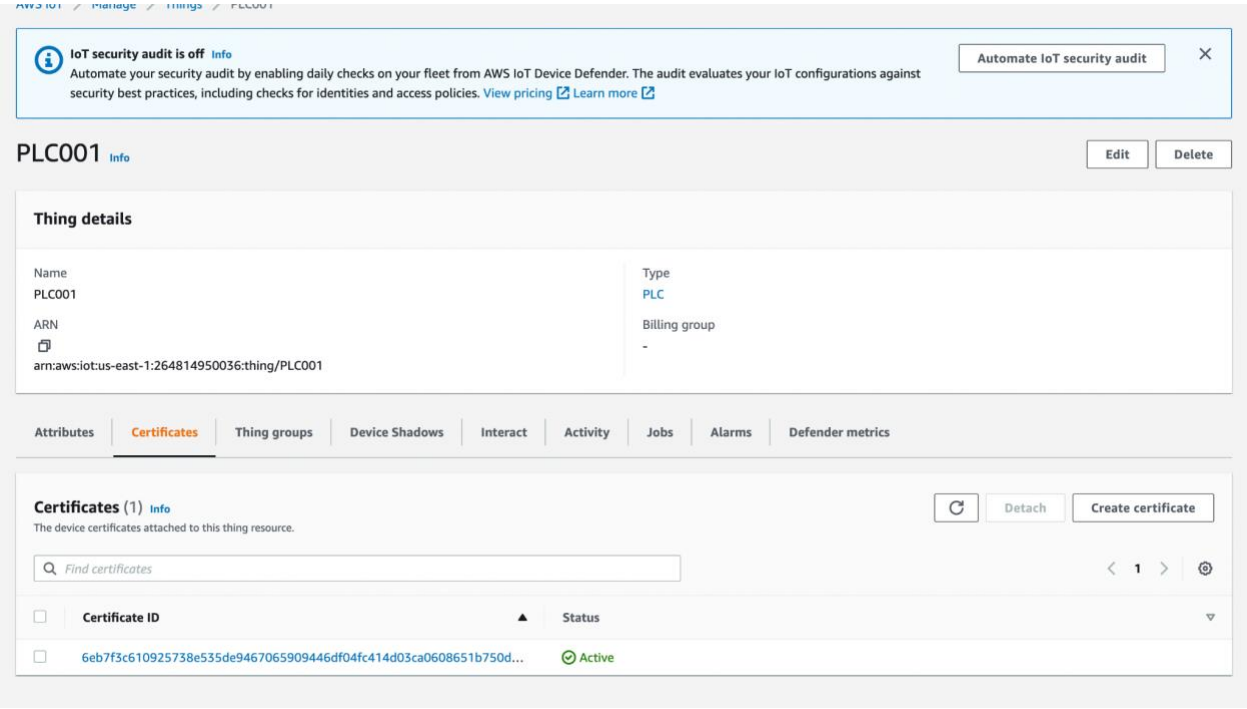
1. Create the MQTT Client Devices as a Thing on the IOT Core



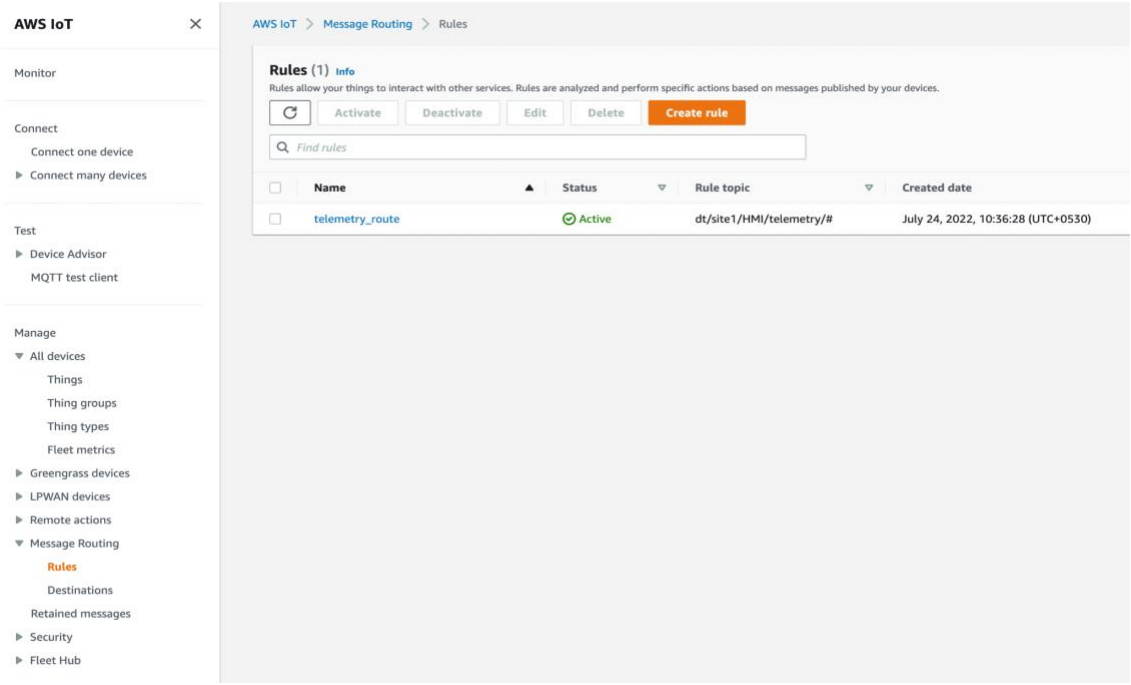
2. Create Policy with action as IOT:Publish and attach to the Things



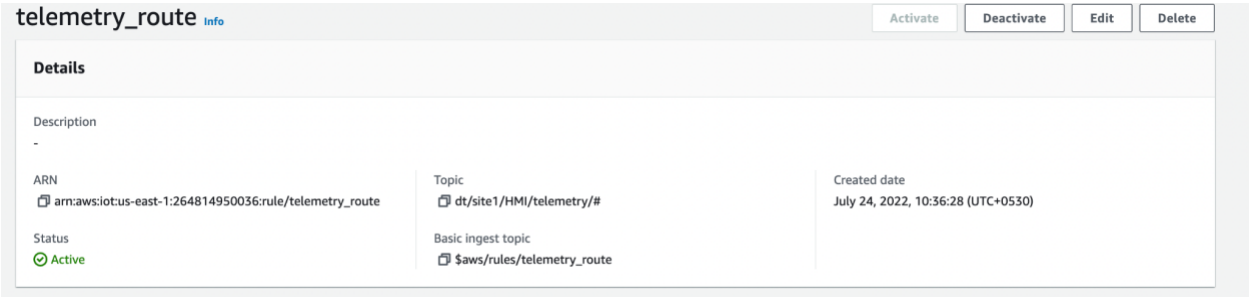
3. Download the certificate and store in secured place



4. Navigate to the Message Routing option and click 'Rule'

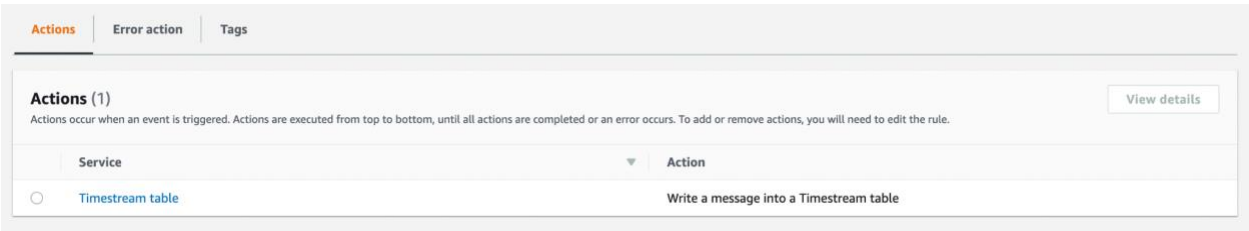


5. Create a rule with source topic published by the devices



6. Create Rule engine with SQL filter which parses the incoming topic and payload

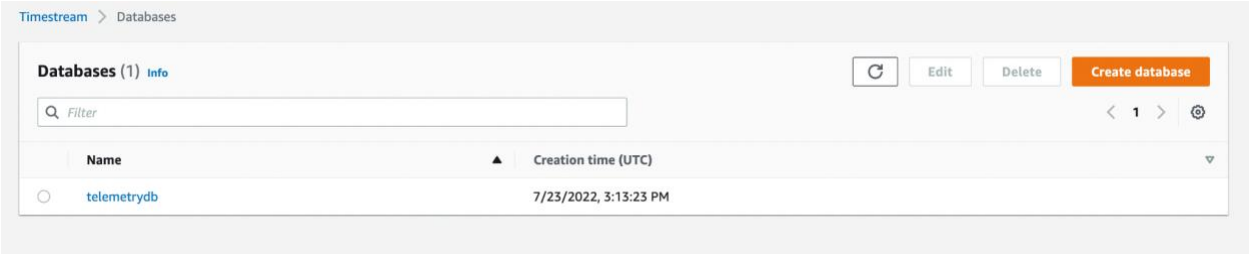
7. Now create the action to which the messages to be stored. Please note AWS Timestream to be created in prior to this step



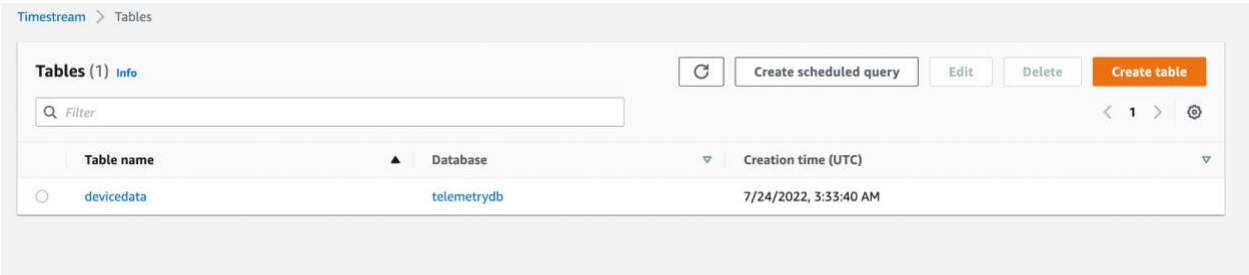
8. Please note the dimensions

Database name	Table name	Dimensions
telemetrydb	devicedata	• 1 ◦ name: device_id ◦ value: ${d.device_id}

9. Create AWS Time stream in the below region US East (N. Virginia), US East (Ohio), US West (Oregon), and EU (Ireland)



10. Create a table with empty records



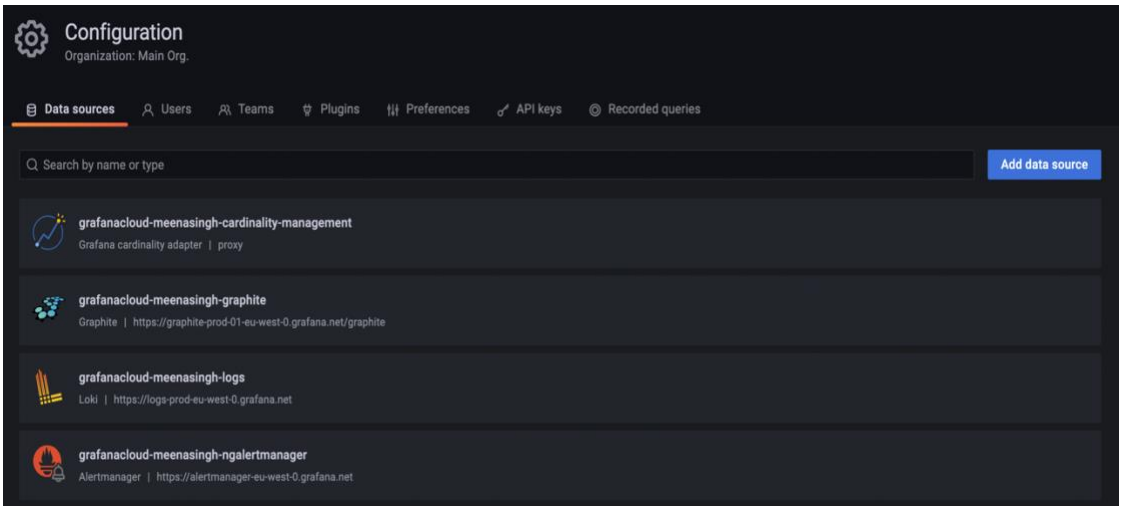
11. Next Create IAM role for Grafana to connect with the AWS Timestream db

Grafana Configuration

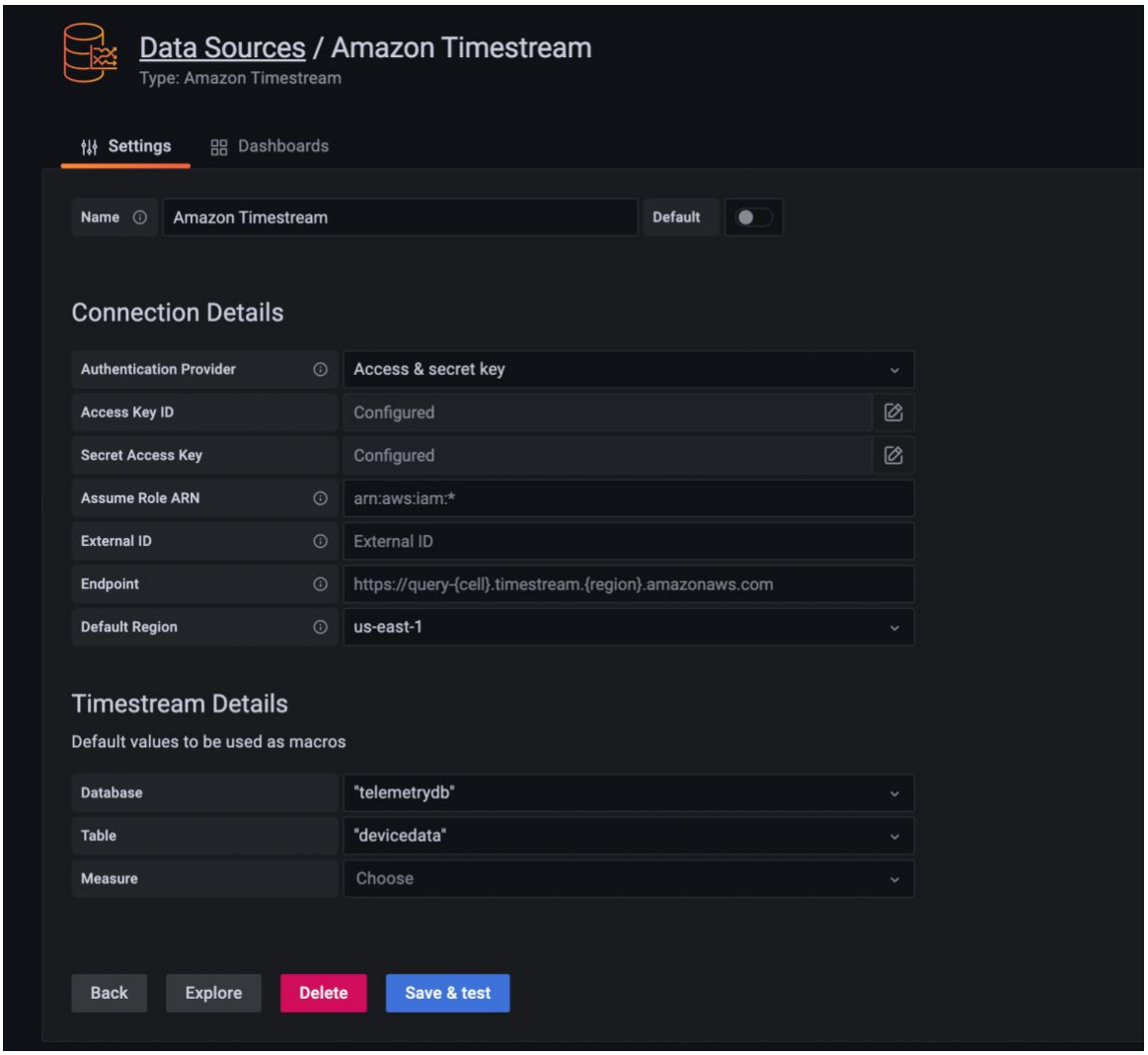
1. Install Grafana on EC2 Ubuntu instance

2. Add Data source before creating dashboard. (Users with the Organization Admin Permission can add data source). Below are the steps to add data source
    a. In the side menu under the **Configuration** link, click on **Data Sources**.
    b. Click the **Add data source** button.
    c. Select Amazon **Timestream** in the **Time series databases** section.

    We are using Amazon Timestream. If you are unable to find Amazon Timestream data source form the given list, then click on Plugin and install Amazon Timestream data source. Repeat from step b.

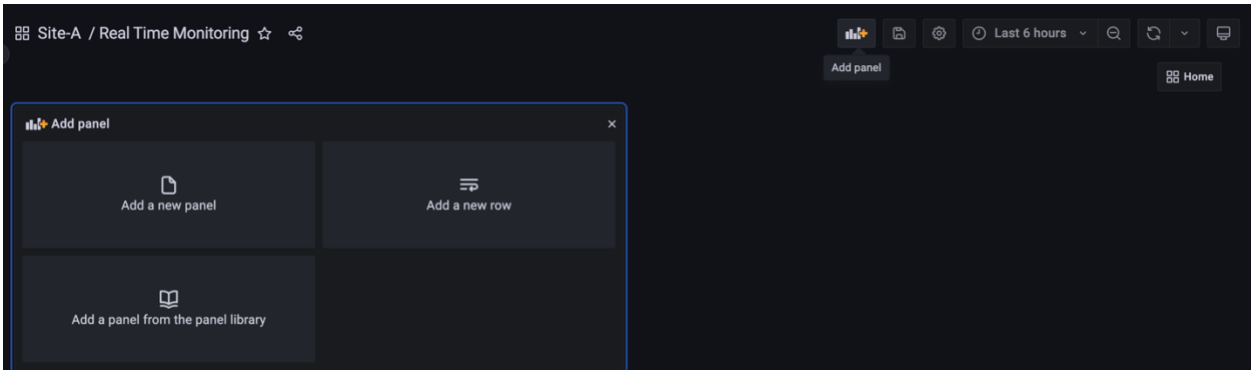


3. Select Amazon Time stream and provide the access key & secret key created during the user creation. Access and secret key corresponds to the StaticProvider and uses the given access key ID and secret key to authenticate. This method doesn't have any fallbacks and will fail if the provided key pair doesn't work.

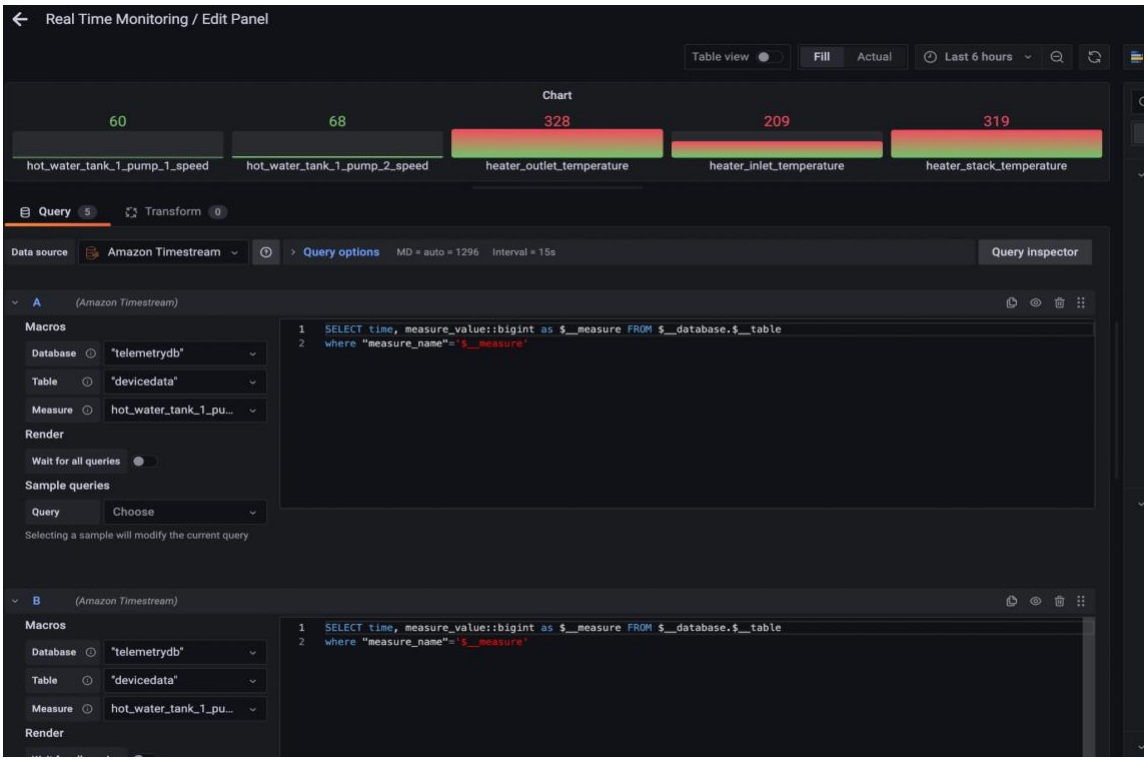


Creating a Dashboard

1. Click on Add Panel icon on the top tight corner to add widget. Select Add a new Panel and customize the widget as per need.



2. Write the timestream query to populate the selected widget with the real time data. Along with Data configuration, create Alert rule to link this panel.



Generating Reports

Grafana's Reporting function allows you to automatically generate PDFs from any of your dashboards and have Grafana email them to interested parties on a schedule. This is available in Grafana Cloud Pro and Advanced and in Grafana Enterprise. There are third party tools such as Skedler (www.Skedler.com) that allow detailed customization beyond what Grafana offers if required.

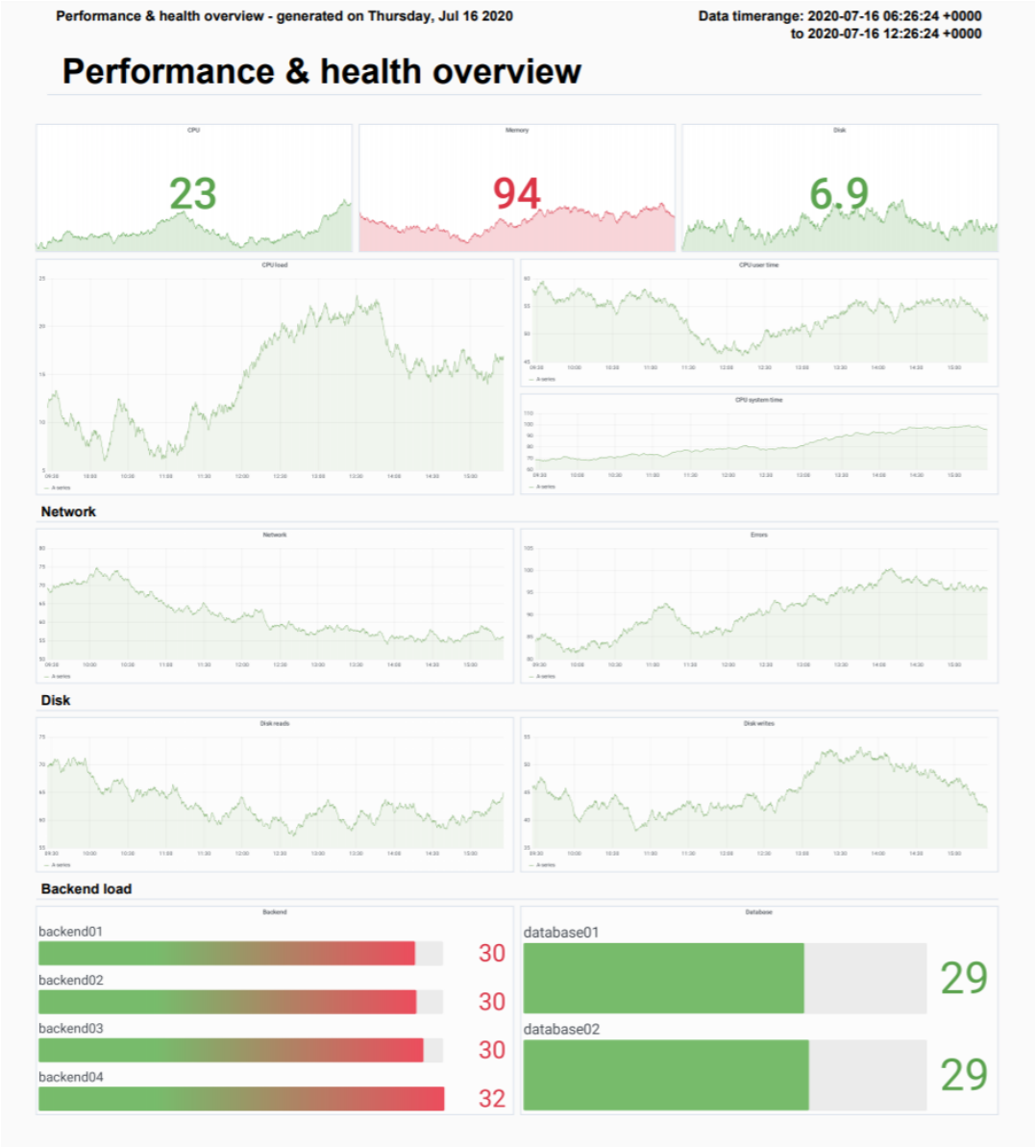


Figure 11: Sample Automated Report

Follow the steps in the link below to setup reporting functionality in Grafana.
https://grafana.com/docs/grafana/latest/enterprise/reporting/

Alerts & Notifications

Grafana's Alerting function allows you to learn about problems in your systems moments after they occur. You can create, manage, and act on your alerts in a single, consolidated view, and improve your team's ability to identify and resolve issues quickly.

Grafana Alerting is available for Grafana OSS, Grafana Enterprise, or Grafana Cloud. With Mimir and Loki alert rules you can run alert expressions closer to your data and at massive scale, all managed by the Grafana UI you are already familiar with.

The diagram below gives you an overview of how Grafana Alerting works and describes the key concepts that work together and form the core of our flexible and powerful alerting engine.

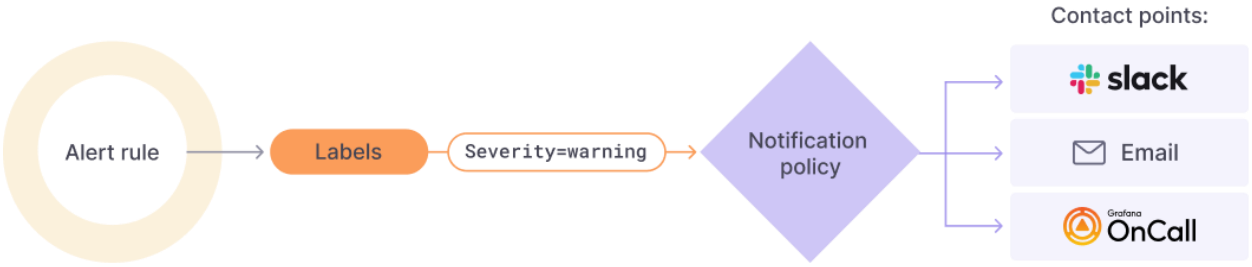


Figure 12 : Sample Alert & Notification Workflow

Alert rules

Alert rules set evaluation criteria that determines whether an alert instance will fire. An alert rule consists of one or more queries and expressions, a condition, the frequency of evaluation, and optionally, the duration over which the condition is met.

Grafana managed alerts support multi-dimensional alerting, which means that each alert rule can create multiple alert instances. This is exceptionally powerful if you are observing multiple series in a single expression.

Once an alert rule has been created, they go through various states and transitions. The state and health of alert rules help you understand several key status indicators about your alerts.

Labels

Match an alert rule and its instances to notification policies and silences. They can also be used to group your alerts by severity.

Notification policies

These policies set where, when, and how the alerts get routed. Each notification policy specifies a set of label matchers to indicate which alerts they are responsible for. A notification policy has a contact point assigned to it that consists of one or more notifiers.

Contact points

Contact points define how your contacts are notified when an alert fires. Grafana supports a multitude of ChatOps tools to ensure the alerts come to your team.