**NPM (Network Performance Monitoring) On New Relic**

**What is NPM?**

**NPM** **–**

* Network performance monitoring in New Relic involves tracking and analyzing the performance of network infrastructure and communication channels within an organization's IT environment.
* It helps in identifying issues such as latency, packet loss, and bandwidth constraints that can impact the overall performance and availability of applications and services.

**Objectives –**

* Poll SNMP data from network devices and send it to New Relic.
* Sample network flow data and send it to New Relic.
* Create a New Relic dashboard for monitoring your network devices.
* Create a New Relic workload to logically group your devices and set anomaly detection.
* Use your new data to understand behaviors within your network.

**Requirements -**

* A New Relic Account.
* A full platform user or a core user with the Nerdpack modify user privilege.

**Steps -**

* Access the GCP jump server to establish connectivity to our server from here.
* Log in to the server using its name.



* Once you are logged into the server, you should have root access. Then switch from a normal user to the root user.
* Then, to check the running containers on the server, use the command below.

**Command**: docker ps

Here, you can see all the running Docker containers

* There is 3 docker containers-
* SNMP v3 polling
* Syslog collector
* NetFlow collector

A black screen with white text

Description automatically generated

* We will mostly focus on SNMP v3 polling. Because rest 2 collectors are using the same configuration.
* In order to create this container, you just need to pull that image from Newrelic public repository and run this image then this particular container will get hosted here, that should have all the API keys and all.
* Suppose we do have 10 network devices.

**Problem Statement:**

* We do have a server here which is basically a collector and here we do have New Relic which is basically a SAAS based platform. Sending data from our network device to New Relic is a challenge for us.

**Solution**:

* So, what we did was put a server in middle and host a docker container on that. What configuration we have did in this container, this configuration will do SNMP polling. SNMP polling will be done from this particular collector (SNMP v3 polling) to 10 network devices through port 161.
* It will get pulled from network devices through port 161 and the data which will come from there will come on port 162 of VM, then this data will be sent to container (ktranslate-snmp-v3-polling) via port 1620.

161>>162>>1620

New Relic

VM

Network Devices

SNMP Polling

API

Containers

161 162:1620

**Fig: Architecture of SNMP V3 Polling**

* This particular collector will start sending data to New Relic cloud which is basically SAAS through Rest API.
* So, this is an overview of the architecture that we do have.

**For Syslog Collector**

* It will get pulled from network devices through port 161 and the data which will come from there will come on port 514 of VM, then this data will be sent to container (ktranslate-syslog-collector) via port 5143.

161>>514>>5143

* But for SNMP and syslog port is default i.e. 162 and 514
* All the SNMP (V1, V2, V3) are using either port 161 and 162

**For NetFlow**

* For NetFlow, here we are using port 9995. This port can be customized.
* We have used different ports for network devices here.

**For checking configuration file**:

* If you just switch to root directory.

**Command: cd/**

Above command redirects to on **[root@cg1-nsnp00apo1 /]**

* For getting list of directories and files:

**Command**: ls -lthr

Here you will see the **SNMP-Base.yaml** file.

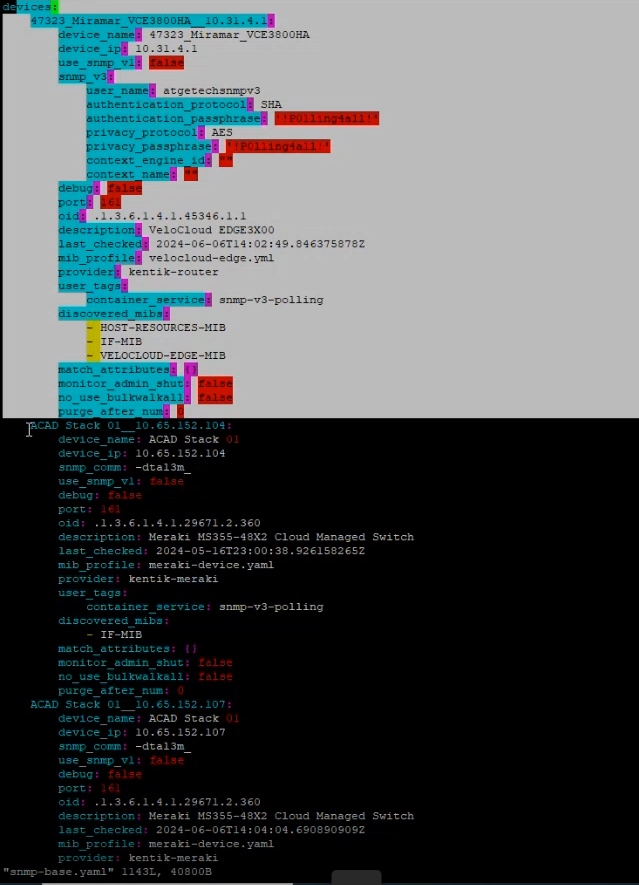
* To open this file in editor mode, Enter the below command.

**Command**: vi SNMP-base. yaml

* If you are familiar with docker you might be aware of it that indentation in YAML file is quite important.
* The highlighted part is the configuration for one device.
* It will not check v1 because it is mentioned as false. It will check v3. But if v2 is configured on this device. In that case v3 will be checked, condition will not be fulfilled or not getting v3 credentials then it will come out from the entire loop and redirect towards keyword **default\_communiti**es:

And it will check this default communities through snmp v2

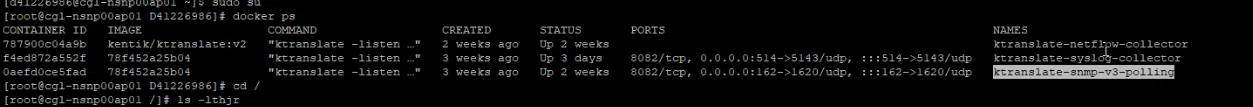
* Note- When you are using SNMP v1, there is no authentication password/protocol for v1.
* If you are using v2 for that you just need to community string.
* The only thing is that they need to use the same credentials. If they used the same credentials, then you don’t need to change the password here.



A screenshot of a computer

Description automatically generated

* If you want to add new device for pooling then you need to insert the IP address and suffix in cidrs block and when you restart the container it will automatically do SNMP walk.
* You need not do it manually. The only thing is that you need to mention the credential here. You need to make sure either they are using SNMP v2 or v3 with these credentials.

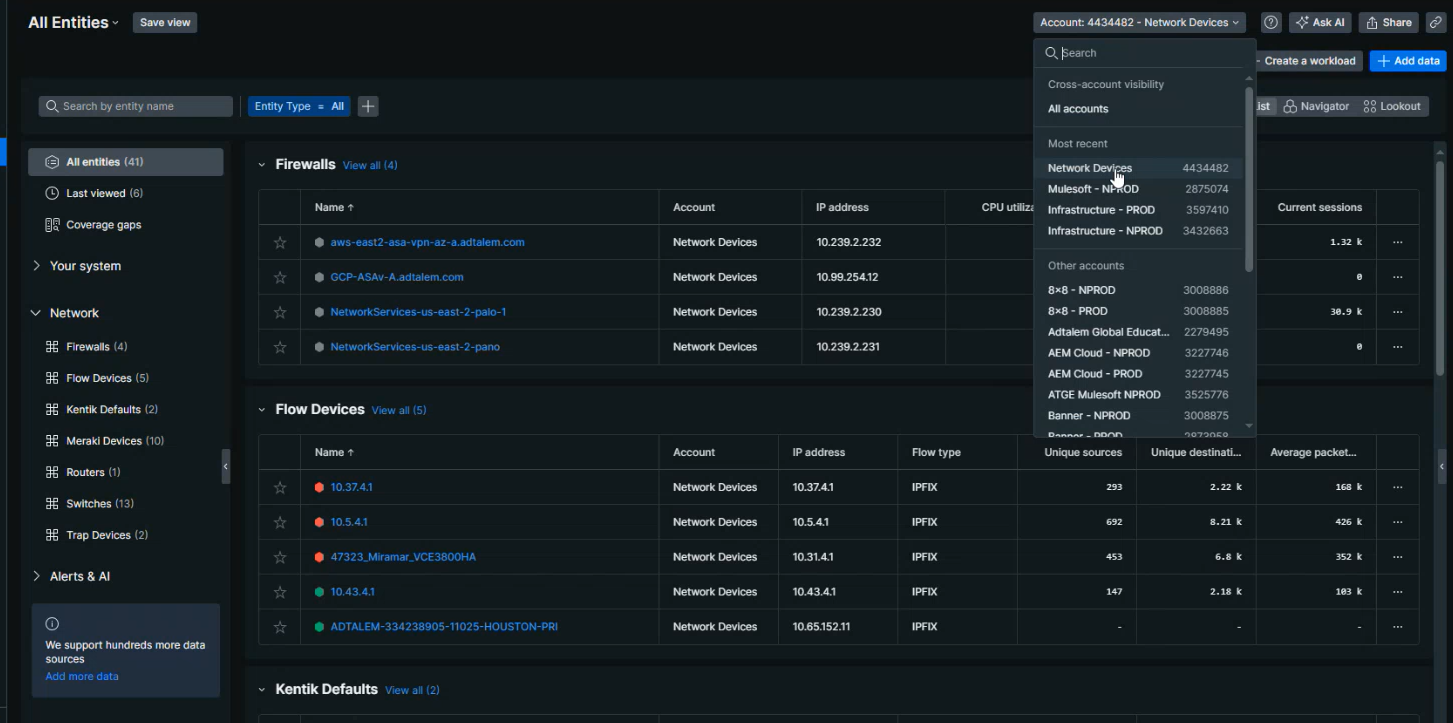


**Object Identifier (OID):**

* An Object Identifier (OID) is a globally unique identifier used in SNMP (Simple Network Management Protocol) to uniquely identify managed objects in a MIB (Management Information Base).
* OIDs are structured hierarchically and are used to access and manage network devices such as routers, switches, and servers. Each OID corresponds to a specific parameter or statistic, like device status, configuration settings, or performance metrics.

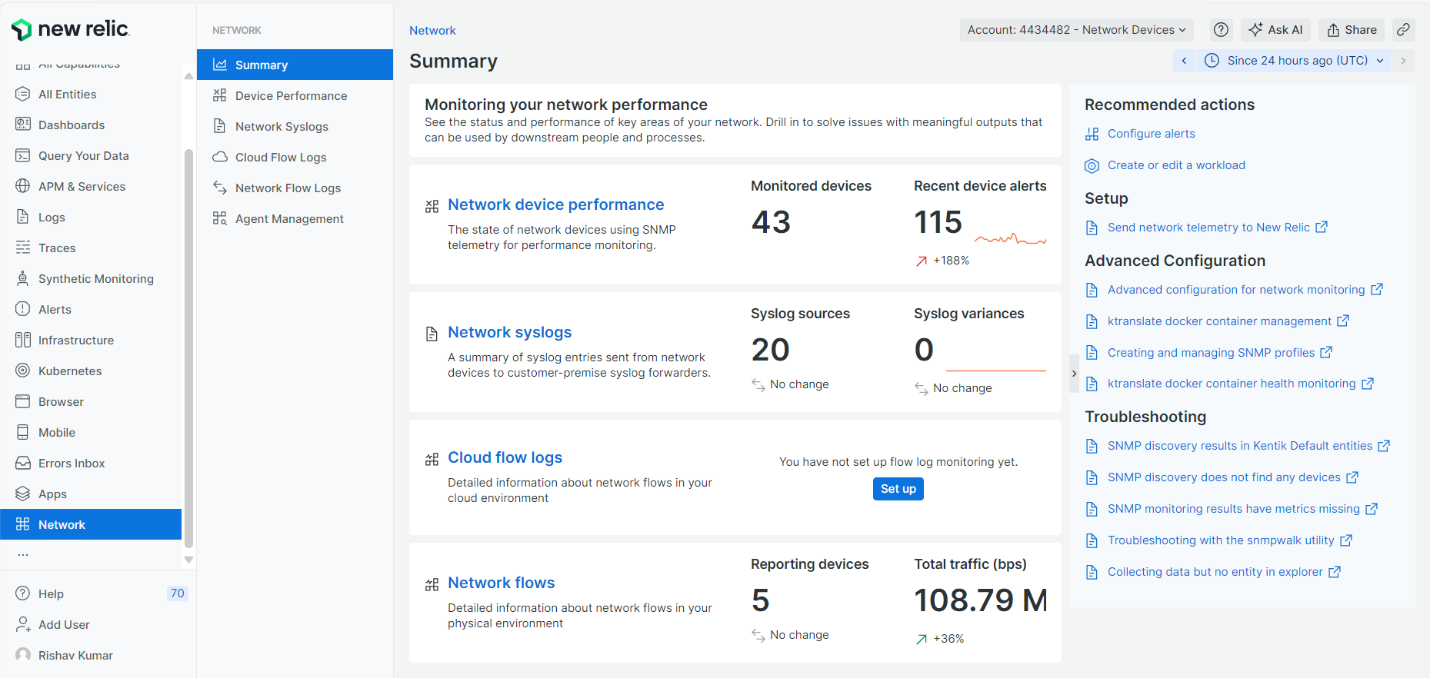
**Management Information Base (MIB)**:

* A Management Information Base (MIB) is a collection of information organized hierarchically.
* These collections are accessed using a network management protocol such as SNMP. MIBs are used to manage entities in a network, such as routers, switches, servers, and other network devices.
* There we have separate accounts in New Relic for the Network devices.



* Log in to your New Relic account. Then go to Network -> click on summary.

**Network Performance Monitoring Summary:**



* Network device performance = SNMP v3 polling + NetFlow

**Network Syslog and Network Flows Configuration:**

* NetFlow and syslog configuration is like that device sending the data to the collector and collector is sending the data to New Relic cloud.

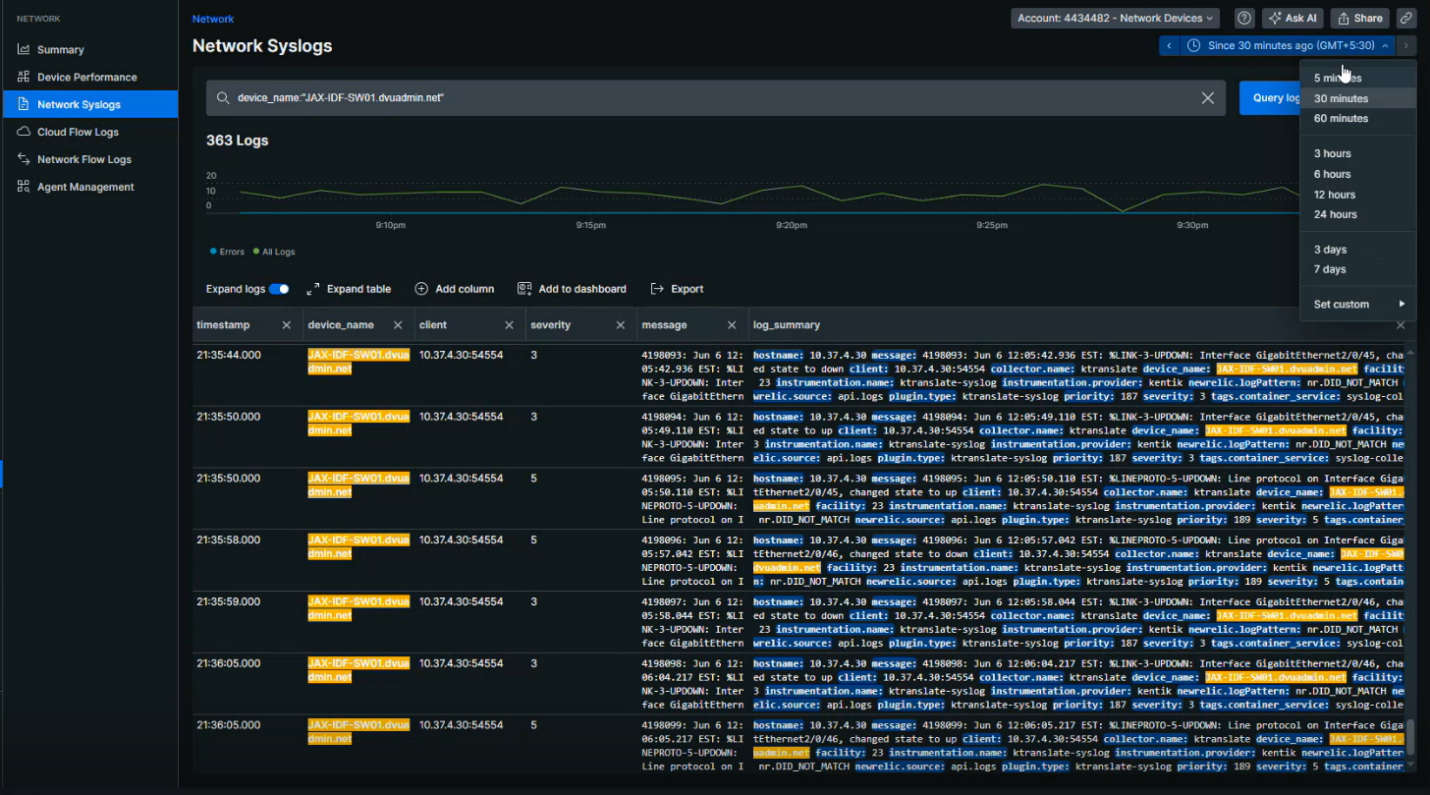
**SNMP Configuration:**

* Configuration for the SNMP is our collector is polling data to the network device through SNMP v3 polling then devices sending data to the collector. Then the collector sends the data back to the New Relic.

**Syslog –**

If you want to find out the log of any device then you can use filter here according to timestamp, device name etc.

Example: device\_name: “test1”



* The count of network syslog and network flows varying because, If any device does not generating a logs from more than 30 mins then that device will be removed from the network syslog and network flows as well.
* You can export your logs in JSON or CSV format. Or you can add your logs to dashboards as well.
* We have set up an alert for error and warning for logs under alerts and AI.

**Network Flow Logs.**

* You can see the flow overview by clicking on Flow overview section.
* You can change the count of displaying the top as per your requirement to see network flow graph.

A screenshot of a computer

Description automatically generated

* Here you can filter the result based on your need from dropdown.
* In the below graphs you can see the top 10 received bits per conversation.

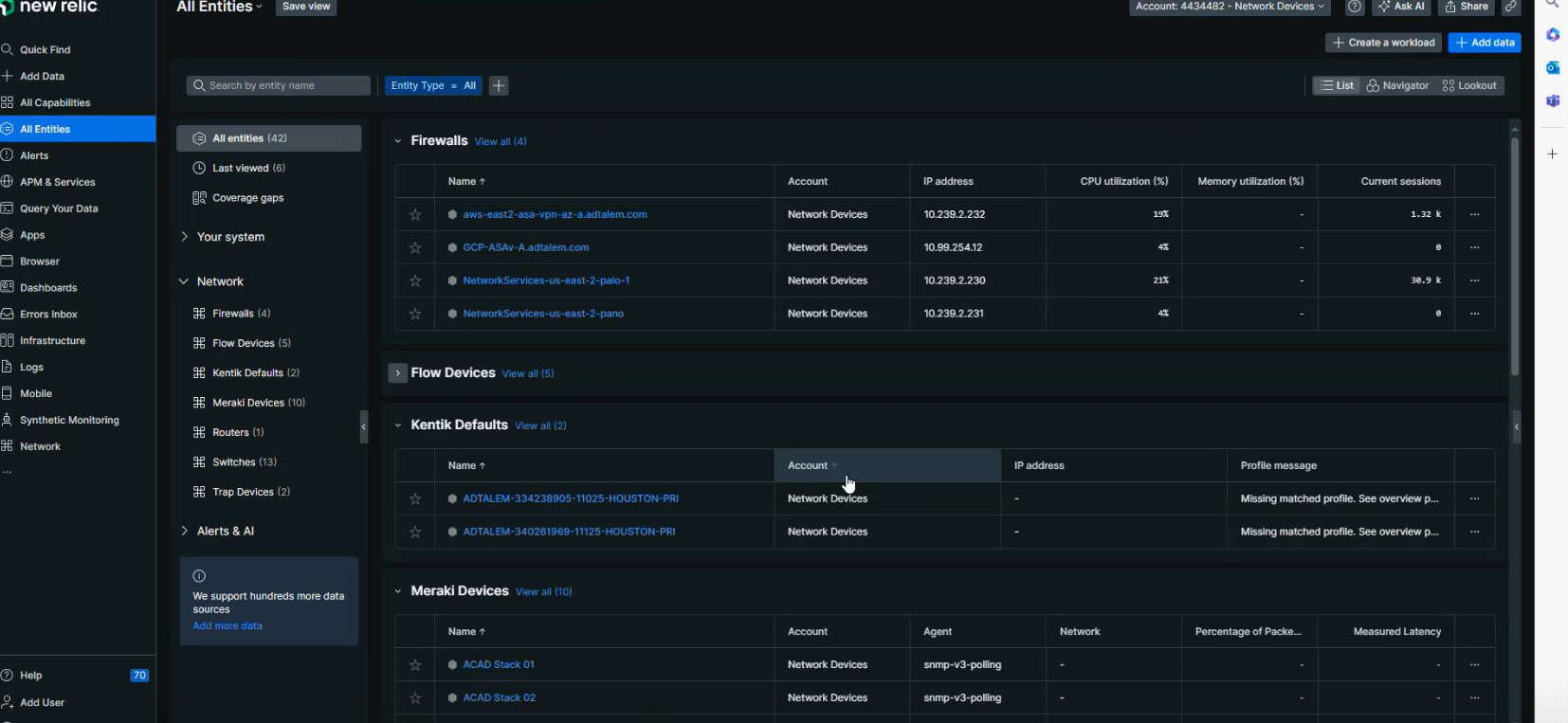
**Top talkers:**

* **Top Talkers** refers to the entities (such as IP addresses, users, or devices) that generate the most network traffic within a specific time frame.
* This metric is critical in network performance monitoring as it helps identify which sources or destinations are contributing the most to network load, allowing for more effective traffic management and troubleshooting.

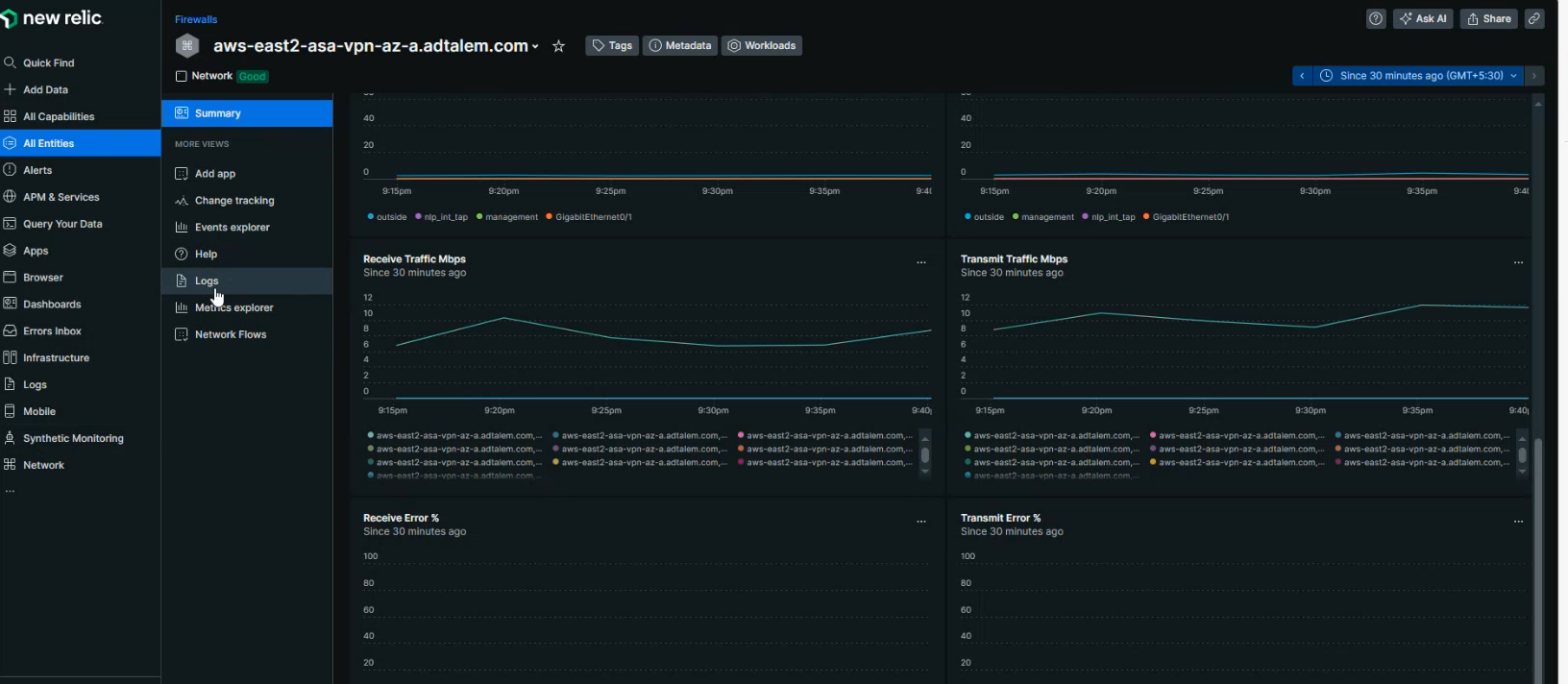
A screenshot of a computer

Description automatically generated

* You can see Firewalls, Flow Devices, Kentik Defaults, Meraki Devices, Routers and Switches by clicking on all entities section in the left pan of this environment.



* If we are enabled syslog for any devices, then we can see it in log section and if we are enabled NetFlow then we can see it in Network Flows.



**Within Navigator view:**

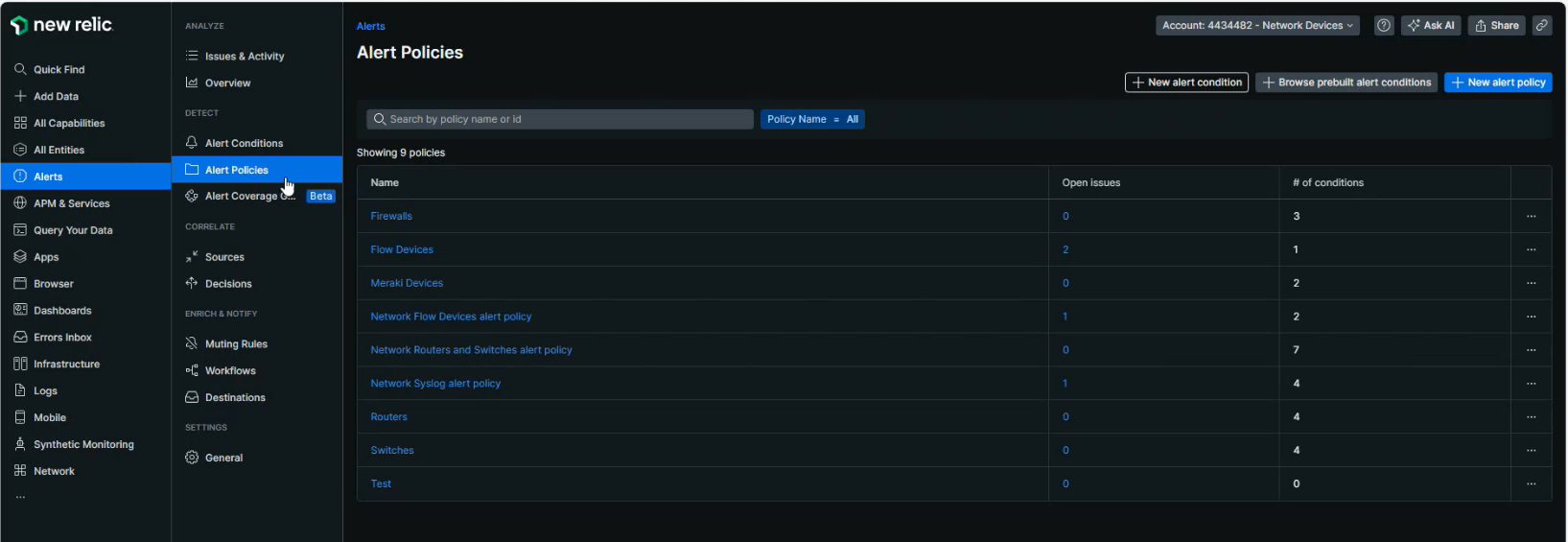
* If the color of devices (represented in hexagon) is green, means it’s in healthy state.
* If color of devices is in red means it has some critical violation and if it is in gray color means it is not going monitored.

A screenshot of a computer

Description automatically generated

**Alerts in New Relic:**

* There are different alert policies for different devices. You can see it under the alert policies section.
* We have different policies for different devices.
* We have created policies for Firewall, Flow devices, and Meraki devices.
* We have different policies for Network routers and Switches alert policy and Network Syslog alert policy.



* We have CPU and Memory utilization alert policies for Routers and Switches.
* If you want to create an alert, then you have to write the NRQL query.

**Notifications of Flow Devices Under Alert Policies:**

* If we are talking about notification under the Alert policies, here you can see ServiceNow and email.
* I’ve already integrated this with ServiceNow QA. Also, I have added 3 destinations for email.

A screenshot of a computer

Description automatically generated

**Set up of Notification Message:**

* You can set up the email notification format as per your need.

**Email destination-**

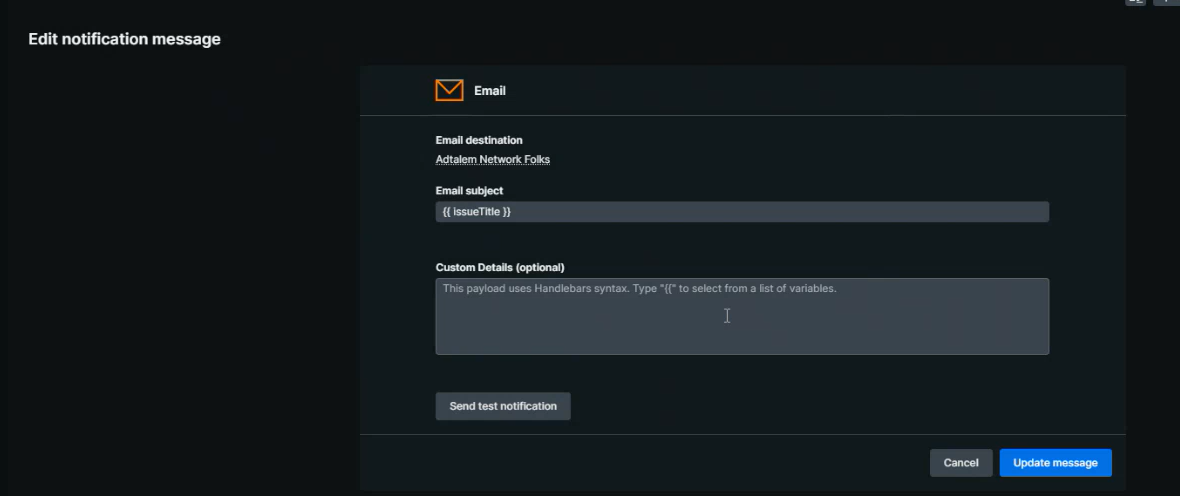
* In email destination you can add those, for which you want to send notification.

**Email Subject-**

* Write your issue title in email subject.

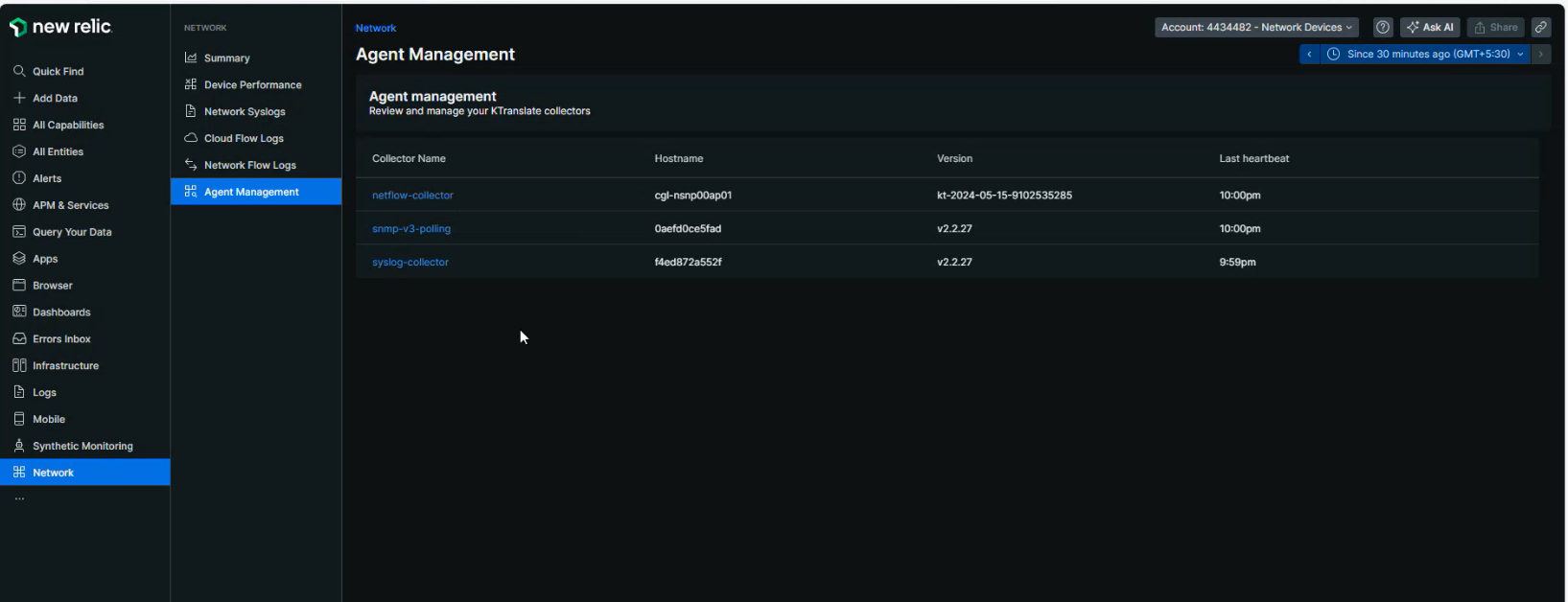
**Custom Details-**

* You can write custom messages here. You can also add variables here.
* You can write all your messages which you want in your alerts.



**Agent Management under Network:**

* We have one server, and all the collectors are in this server. In case of any issue on this server or if there is any critical issue on the datacenter in which the server is, then all notifications of all those devices which are monitored under this server will be stopped.
* To overcome this problem, we need to create 3 nodes within 3 different datacenters at different locations.
* If one node is getting down, then traffic will be shifted to the other two nodes that will be managing through the Kubernetes.
* We have added monitoring for 3 collectors under alert management.



**Thank You…**