K8s Monitoring Guide

<?xml version="1.0"?><DocumentBlank xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <GraphicCharterDefinitionId>0</GraphicCharterDefinitionId> <TemplateBaseTypeId>0</TemplateBaseTypeId> <CompanyId>1</CompanyId> <ConfidentialId>0</ConfidentialId> <ConfidentialDescription /> <CountryId>0</CountryId> <PageSizeId>1</PageSizeId> <PageOrientationId>1</PageOrientationId> <PrePrintedStationary>false</PrePrintedStationary> <Project /> <Reference>20191007-151044-AZ</Reference> <TemplateType>1</TemplateType> <CultureId>en-US</CultureId> <LanguageId>8</LanguageId> <Customer /> <DocumentDate>2019-10-10T12:49:13.9373864+05:30</DocumentDate> <FirstPageHeader>AWS Landing Zone Shared Services</FirstPageHeader> <FirstPageSubHeader /> <FirstPageTitle\_Blank>AWS Landing Zone Shared Services</FirstPageTitle\_Blank> <FirstPageSubtitle\_Blank /> <Saved>false</Saved> <IsValid>true</IsValid> <FirstPageCover>false</FirstPageCover> <IsNew>true</IsNew> <Title>AWS Landing Zone</Title></DocumentBlank>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
|  |  |  |  | |
|  |  | * Prometheus * Grafana * AlertManager * Node exporter | | |
|  |  |  | | |
|  |  |  | | |
|  |  |  | |  |

# **OBJECTIVE :**

**Prometheus**

Monitoring infrastructure, EC2 instance, application URL/Services/Log, databases health, docker process/status, kubernetes clusters and more….

**Grafana**

Grafana allows us to visualize, alert[s], log[s] on grafana dashboard no matter where they are stored. Create, explore, and share dashboards with team and foster a **data driven culture**.

# **OVERVIEW of Tools :**

**Prometheus**

[Prometheus](https://github.com/prometheus) is an open-source system monitoring and alerting toolkit originally built at [Sound Cloud](https://soundcloud.com). Since its inception in 2012, many companies and organizations have adopted Prometheus, and the project has a very active developer and user [community](https://prometheus.io/community). It is now a standalone open source project and maintained independently of any company. To emphasize this, and to clarify the project's governance structure, Prometheus joined the [Cloud Native Computing Foundation](https://cncf.io/) in 2016 as the second hosted project, after [Kubernetes](https://kubernetes.io/).

Prometheus is a monitoring system and time series database, with powerful query language, aggregation, and alerting capabilities.

Our infrastructure and operations team uses Prometheus in multiple ways. We use it to monitor resource usage and service performance, as well as data pipeline processing queues. Additionally, we keep track of Kubernetes deployment status and metrics.

Devops and analytics team uses Prometheus for several artificial intelligence-related metrics.

**Grafana**

Grafana is an open source metric analytics & visualization suite. It is most commonly used for visualizing time series data for infrastructure and application analytics but many use it in other domains including industrial sensors, home automation, weather, and process control.

## Features :

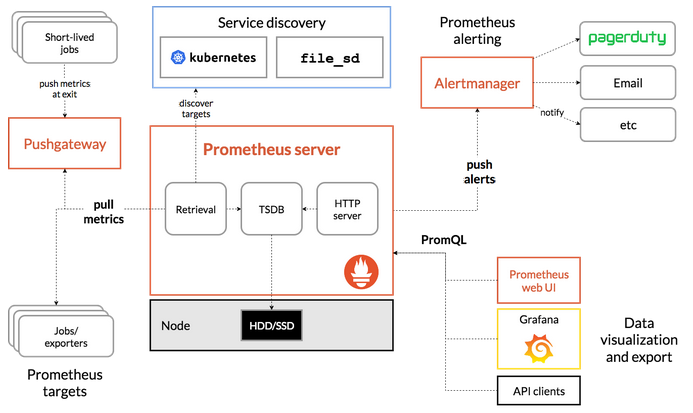
**Prometheus**

* A multi-dimensional data model (time series defined by metric name and set of key/value dimensions)
* A flexible query language to leverage this dimensionality
* No dependency on distributed storage; single server nodes are autonomous
* Time series collection happens via a pull model over HTTP
* Pushing time series is supported via an intermediary gateway
* Targets are discovered via service discovery or static configuration
* Multiple modes of graphing and dash boarding support
* Support for hierarchical and horizontal federation

**Grafana**

* Visualize :-
  + Fast and flexible client side graphs with a multitude of options. Panel plugins for many different ways to visualize metrics and logs.
* Dynamic Dashboards :-
  + Create dynamic & reusable dashboards with template variables that appear as dropdowns at the top of the dashboard.
* Explore Metrics :-
  + Explore your data through ad-hoc queries and dynamic drilldown. Split view and compare different time ranges, queries and data sources side by side.
* Explore Logs :-
  + Experience the magic of switching from metrics to logs with preserved label filters. Quickly search through all your logs or streaming them live.
* Alerting :-
  + Visually define alert rules for your most important metrics. Grafana will continuously evaluate and send notifications to systems like **Slack**, **PagerDuty**, **VictorOps**, **OpsGenie**.
* Mixed Data Sources :-
  + Mix different data sources in the same graph! You can specify a data source on a per-query basis. This works for even custom data sources.
* Annotations :-
  + Annotate graphs with rich events from different data sources. Hover over events shows you the full event metadata and tags.
* Ad-hoc Filters :-
  + Ad-hoc filters allow you to create new key/value filters on the fly, which are automatically applied to all queries that use that data source.

## Architecture



# **GETTING STARTED**

## PRE-REQUISITES

* Linux Server is required to install Prometheus & Grafana .
* Ansible Server install on Centos and Amazon Ami like (ami-0a887e401f7654935)

# **PROMETHEUS**

## Installation

* cd /opt/
* Git clone <https://github.com/chauhanudham/interview.git>
* cd /opt/interview/monitoring\_prometheus
* ansible-playbook ansible\_config.yml

## Tree structure of source codes.

monitoring\_prometheus/

│   ├── ansible\_config.yml

│   └── roles

│   └── prometheus

│   ├── files

│   │   └── grafana.db

│   ├── tasks

│   │   ├── alertmanager\_installation.sh

│   │   ├── grafana\_installation.sh

│   │   ├── grafana.db.txt

│   │   ├── grafana.db.txt1

│   │   ├── grafana.ini

│   │   ├── main.yml

│   │   ├── node\_exporter\_installation.sh

│   │   ├── prometheus\_installation.sh

│   │   └── restart\_services.sh

│   └── templates

│   └── hi

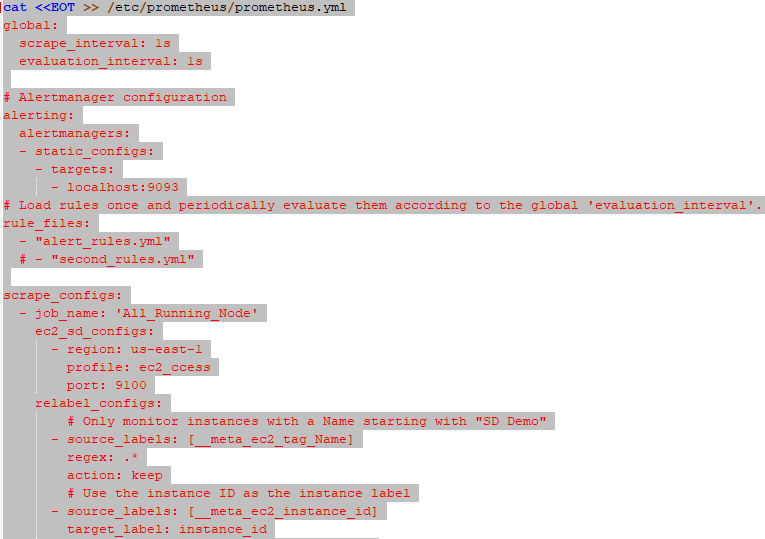
## Role of Ansible

* Deploy Prometheus, Grafana, NodeExporter and Nginx using ansible role. Inside the “/opt/interview/monitoring\_prometheus/roles/prometheus/tasks” all installation scripts are present.
  + prometheus\_installation.sh
  + grafana\_installation.sh
  + alertmanager\_installation.sh
  + node\_exporter\_installation.sh
  + restart\_services.sh

## Configurable Paramètres

### Prometheus

* + Inside « prometheus\_installation.sh” all the configurations are define. All feature, like auto discover service, grouping server, port access and more…



Follow official document for more details and custmization.

<https://prometheus.io/docs/prometheus/latest/getting_started/>

* + Nginx setting is present inside “prometheus\_installation.sh” file.



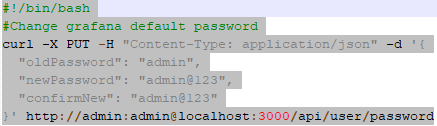


* + Create htpassword for Nginx proxy URL access.



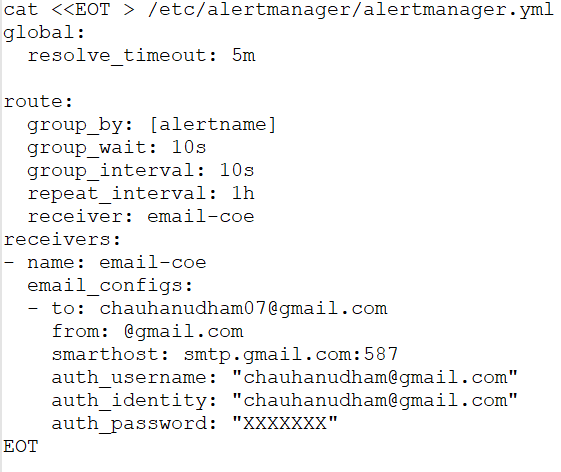
### Grafana

* + Inside « grafana\_installation.sh” all the configurations are define. All the configuration saved inside “grafana.db.txt” which is copied during execuation.
  + Some dashkboard are allready configured like “infra monitring”, “Docker monitring”, “kubernetes monitring”, and more…
  + To change default password of user “admin” use api inside “restart\_services.sh”

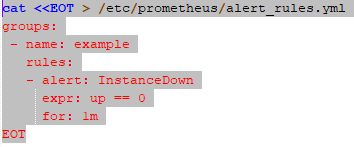


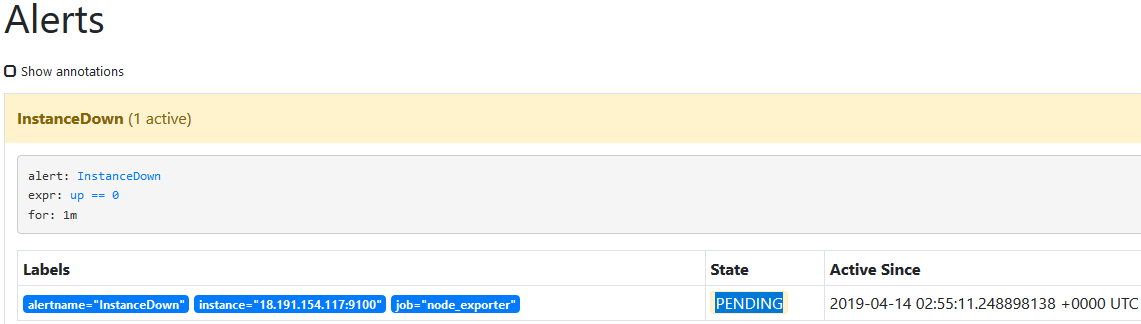
### AlertManager

* + Inside « alertmanager\_installation.sh” all the configurations are define. It is use to send notification to end user by Email. Need to update SMTP configuration inside configuration file.
  + Change/Add SMTP host, auth\_password and mailing details as per we have information.

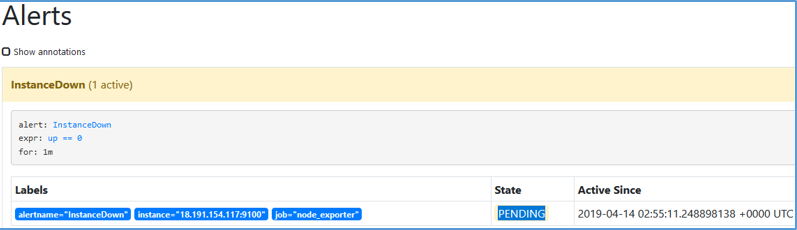


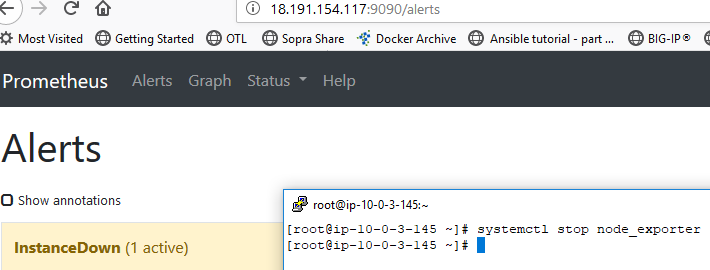
* + Change/Add alerting rules here. We can also update alerting rules after deployment.



**Rules and Alerts are on prometheus interface.**

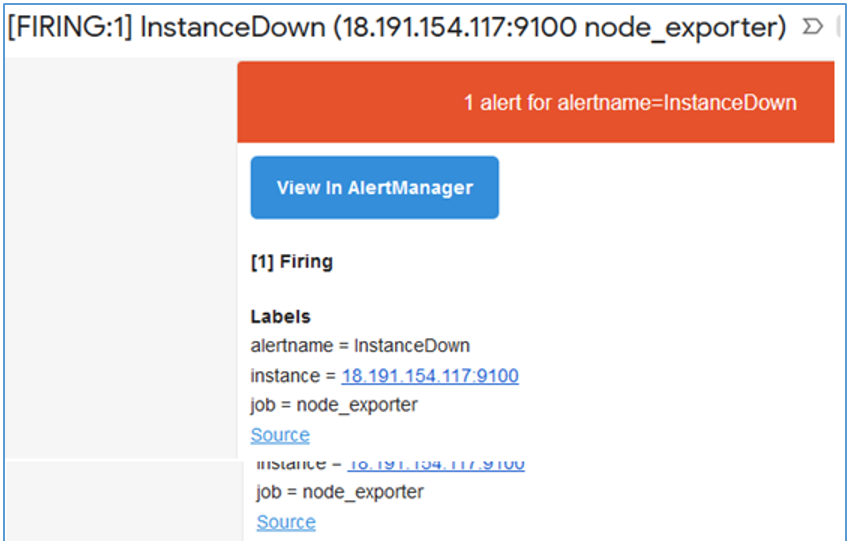
**Testing, manully stop service on node.**

****

 **We have mail alert on mail ID.**

### NodeExporter

* + Node Exporter is a Prometheus exporter for hardware and OS metrics with pluggable metric collectors.
  + It allows to measure various machine resources such as memory, disk and CPU utilization.
  + Inside « node\_exporter\_installation.sh” all the configurations are define. It is use to monitor infra on prometheus server.



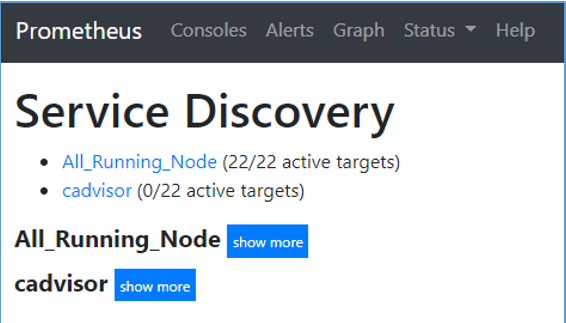
### RestartService

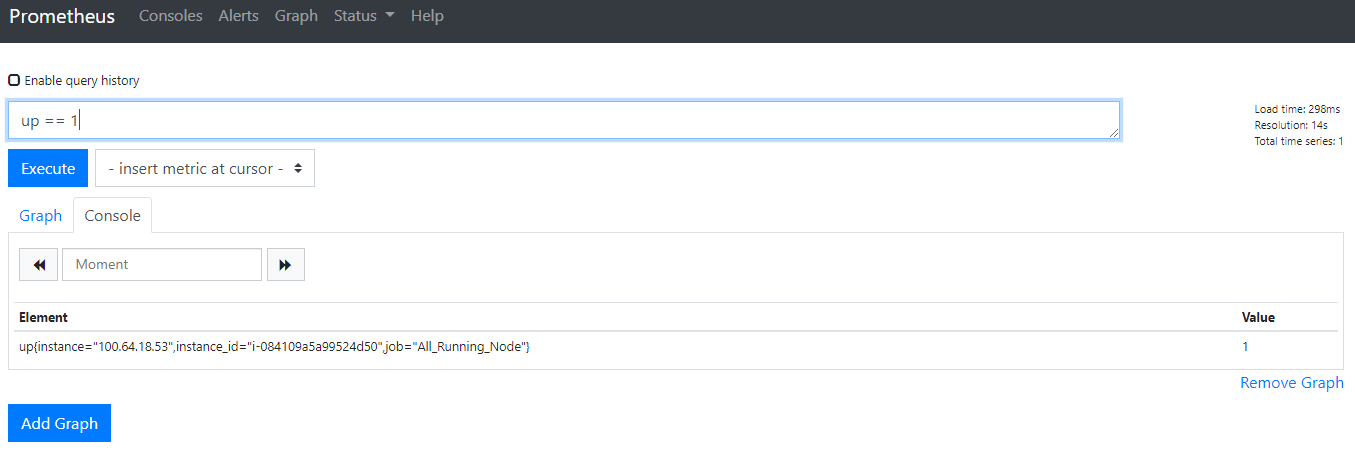
* + Inside « restart\_services.sh” define restart sequences after all deployment.

# Prometheus Web UI

Access Prometheus web UI using domain name that is “grafana.XXXX.com”. and <http://X.X.X.X:9090/>

**Check service discover and other functionality are working.**

****



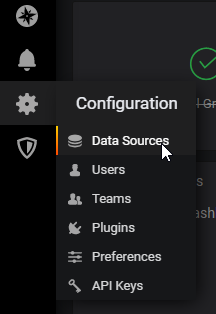
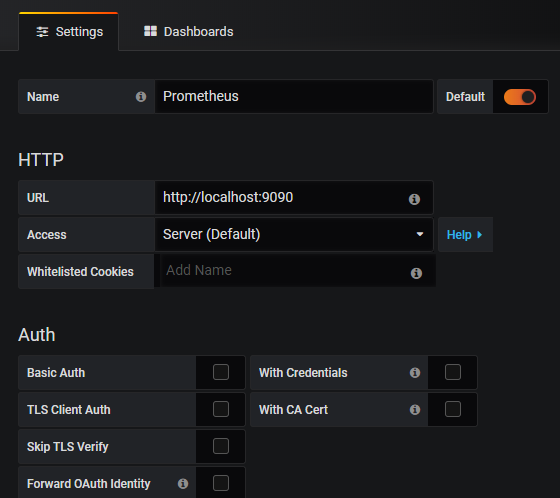
**For adding, any new Service/Alert/Node/NodeExporter update “prometheus.yml” file, which is present on server, inside “/etc/prometheus/” Check service discover and other functionality are working.**

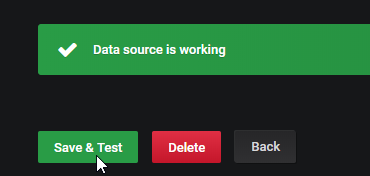
# Grafana Web UI

Access Grafana web UI using domain name that is “grafana.XXXX.com:3000”.

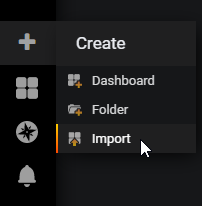


### Prometheus as a data source.

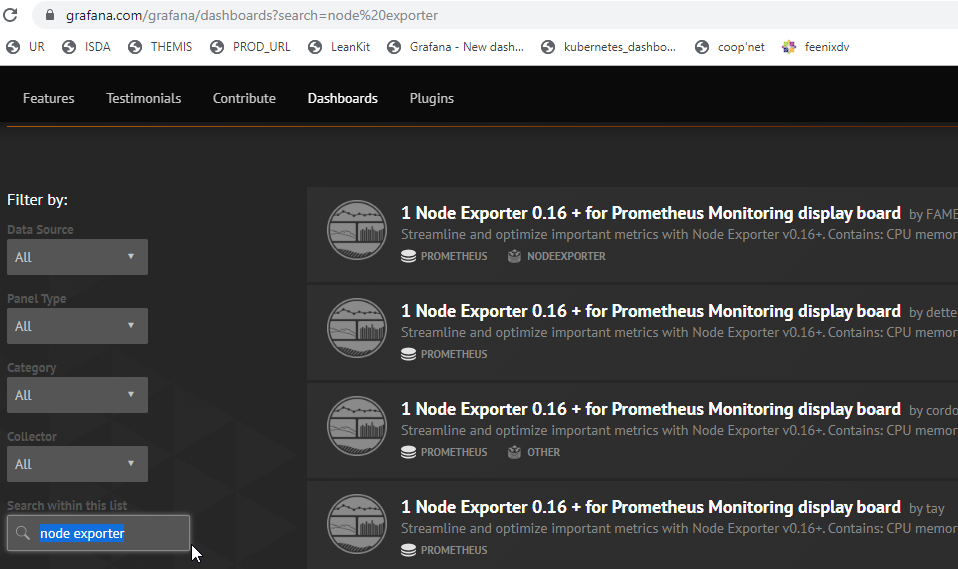
 



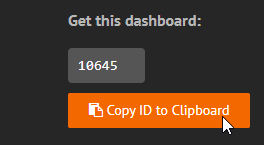
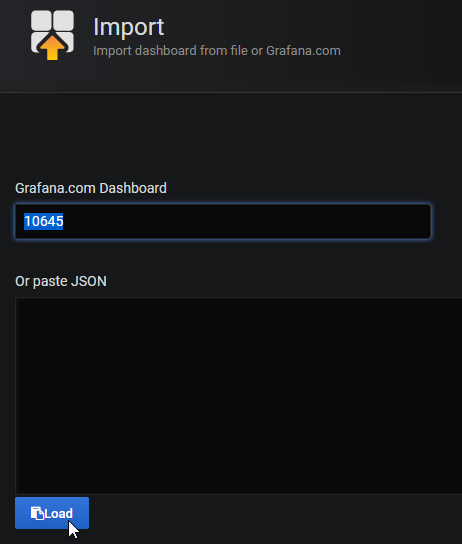
Now add dashboard. Click on Import.



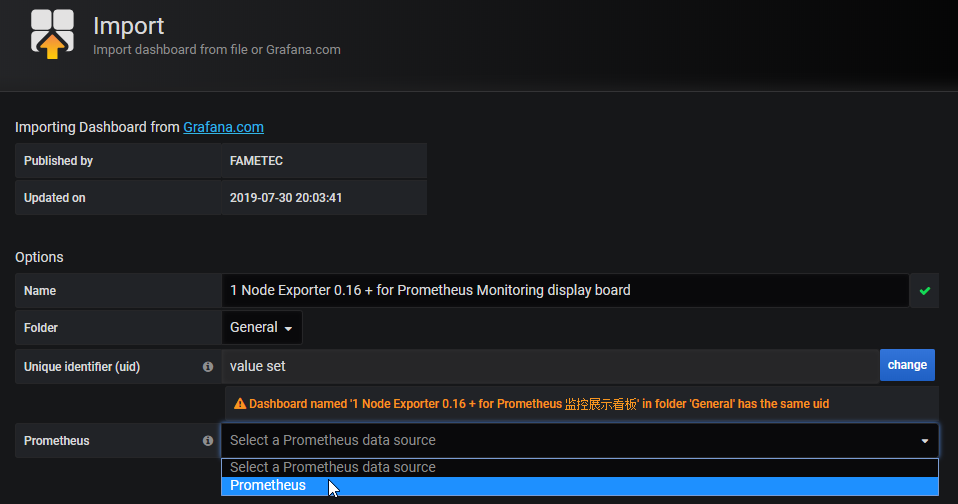
Now visit Grafana official web site for dashboard “<https://grafana.com/grafana/dashboards>” and search dashboard with exporter name.



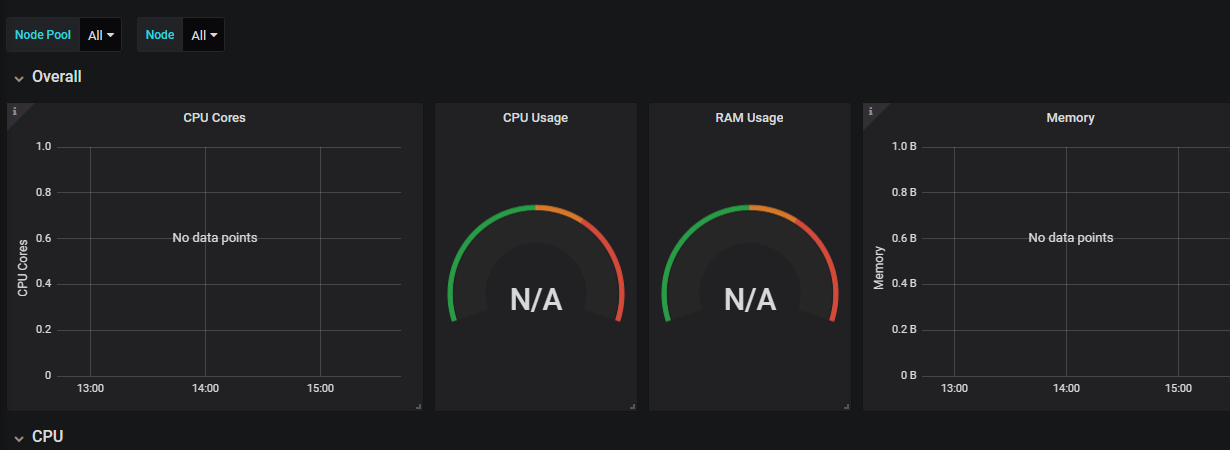
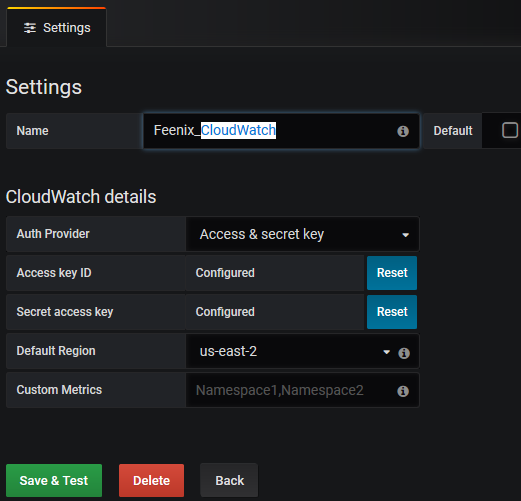
**After selecting dashboard copy dashboard ID and apply on Grafana.**

**Import dashboard.**



**After that dashboard available for monitoring.**



**To add new data source follow same process and import dashboard ID from official site.**

### CloudWatch as a data source for AWS.

**Using credentials file.**

#### Access key and secret key configuration.

root@knode1 grafana]# pwd

/usr/share/grafana

[root@knode1 grafana]# cat .aws/credentials

[default]

aws\_access\_key\_id = XXXXXX

aws\_secret\_access\_key = XXXXXXXX

region = us-east-2

## Manage Organazation In Grafana

### Two levels of administrators:

* Organizational administrators: These admins can manage users within specific organizations in a particular Grafana installation.

As an Organizational administrator, you can add Data Sources, add Users to your Organization and modify Organization details and options.

* Grafana administrators: These super admins can manage users across all organizations in a Grafana installation. They can also change and access system-wide settings.

As a Grafana Administrator, you have complete access to any Organization or User in that instance of Grafana. When performing actions as a Grafana admin, the sidebar will change its appearance as below to indicate you are performing global server administration.

* From the Grafana Server Admin page, you can access the System Info page, which summarizes all of the backend configuration settings of the Grafana server.

**Organizational view on Grafana interface.**

