# By UTD Spring 2024 AR Project Batch

# 

[**By UTD Spring 2024 AR Project Batch 1**](#_heading=h.gjdgxs)

[**Grafana Setup 1**](#_heading=h.3znysh7)

[Introduction 1](#_heading=h.2et92p0)

[Security Groups related to Grafana 2](#_heading=h.tyjcwt)

[Local Installing Grafana - Docker build 2](#_heading=h.3dy6vkm)

[AWS Installation 2](#_heading=h.1t3h5sf)

[Consume AR API from Grafana 2](#_heading=h.4d34og8)

[Configure AR API Server 2](#_heading=h.f6d5je6qcv12)

[Consume API Query 2](#_heading=h.2s8eyo1)

[Connecting Grafana to Postgres - Local Docker environment 3](#_heading=h.17dp8vu)

[Install grafana 3](#_heading=h.3rdcrjn)

[Obtain docker container IP 3](#_heading=h.26in1rg)

[Connecting postgres DataSource 4](#_heading=h.lnxbz9)

[Importing Dashboards 5](#_heading=h.35nkun2)

[Consume Cloud Watch metrics from Grafana 6](#_heading=h.1ksv4uv)

[IAM Policy 6](#_heading=h.44sinio)

[Create policy 6](#_heading=h.2jxsxqh)

[Sample Policy 6](#_heading=h.z337ya)

[Connecting CloudWatch to Grafana 7](#_heading=h.3j2qqm3)

[Create Cloud-Watch connection 7](#_heading=h.1y810tw)

[Import Ec2 metrics Dashboards 8](#_heading=h.4i7ojhp)

[**Pending DevOps Items 8**](#_heading=h.1ci93xb)

[**Final step 9**](#_heading=h.2bn6wsx)

[Validate all data sources 9](#_heading=h.qsh70q)

[Dashboards 9](#_heading=h.3as4poj)

# 

# Grafana Setup

## Introduction

This document outlines creating dashboards in Grafana using the following datasources

* AR JSON API
* Postgress
* CloudWatch

## Security Groups related to Grafana

Port 3000 to access grafana.

We can change the port as needed.

Update docker-compose.yaml - Section:grafana or other installation mop.

## Local Installing Grafana - Docker build

Local development - see the docker file. Refer docker-compose.yml - Grafana Section

## AWS Installation

Refer to Grafana-installation\_V1

## Consume AR API from Grafana

### Configure AR API Server

The following steps are required,

1.Install JSON API plugin

2. Configure JSON API plugin with Connection URL. [https://{AR\_API\_URL](about:blank)}}

Example; https://api.athletereserve.com

2a) Http Headers: AR API is secured via “x-api-key”. Http Headers - provide the header -x-api-key and value from “????????” Refer API AR to find the latest API KeY

Local Installation: [http://web.8000](about:blank) ( local installation). Ensure docker network is created.

### Consume API Query

3.Grafana: Query tab

Path=\api\athletes

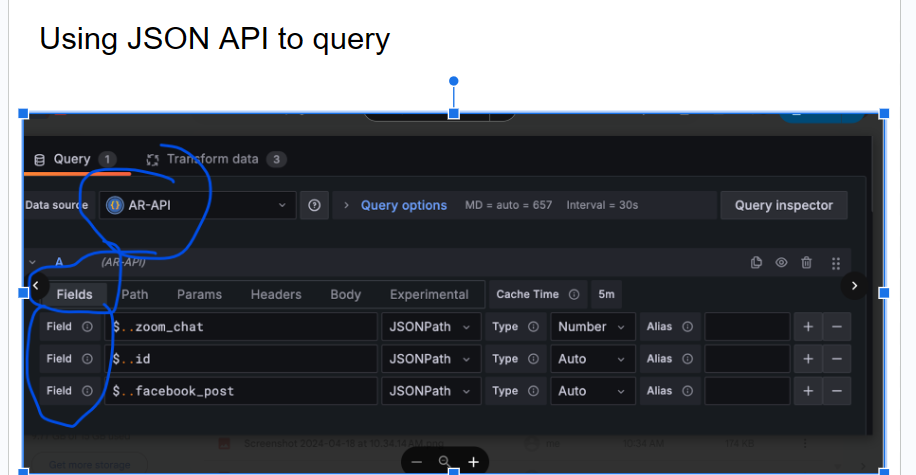
field= $..first\_name

You see the table data. If you want to perform transformation , go transformation tab.

A sample FAQ is updated in GIT\server\grafana\AR-JSONAPI.ppt

4. Import AR\_API\_Dashboards.json.

5. Updated/delete dashboard as needed.

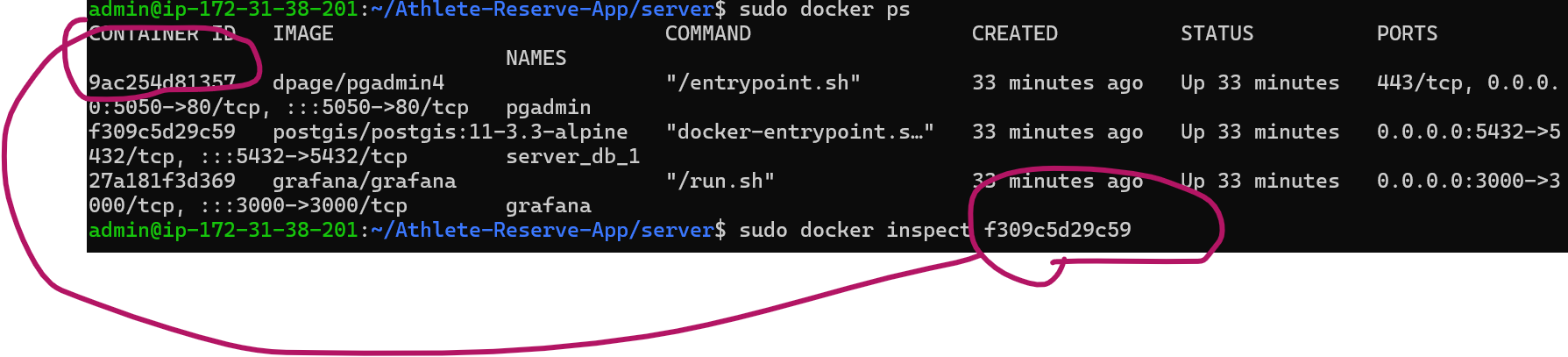


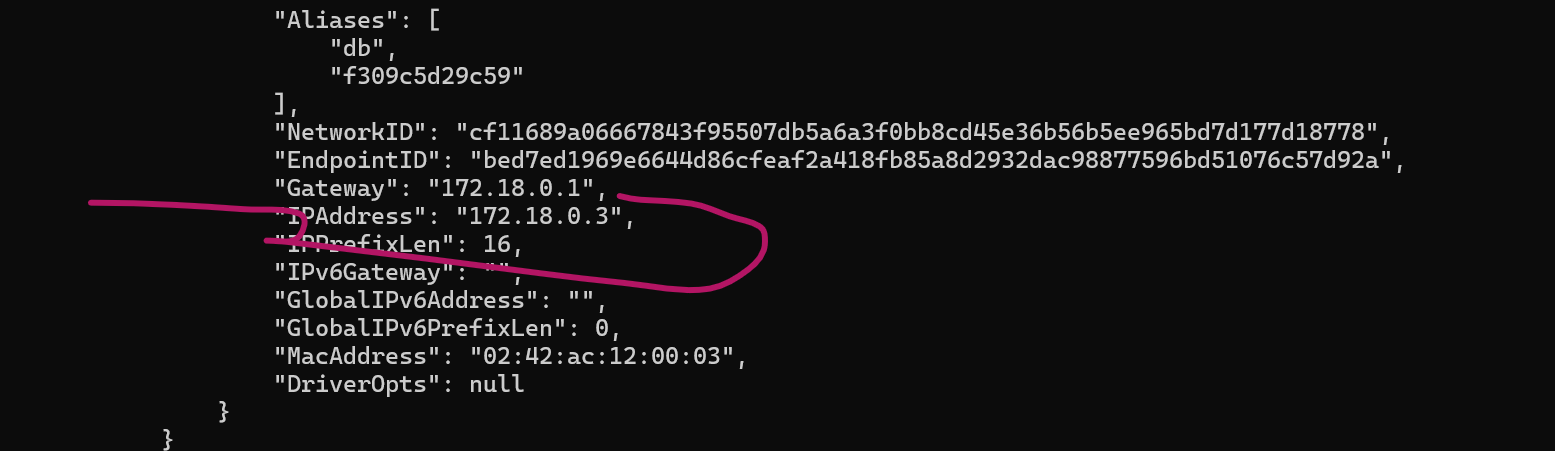
## Connecting Grafana to Postgres - Local Docker environment

### Install grafana

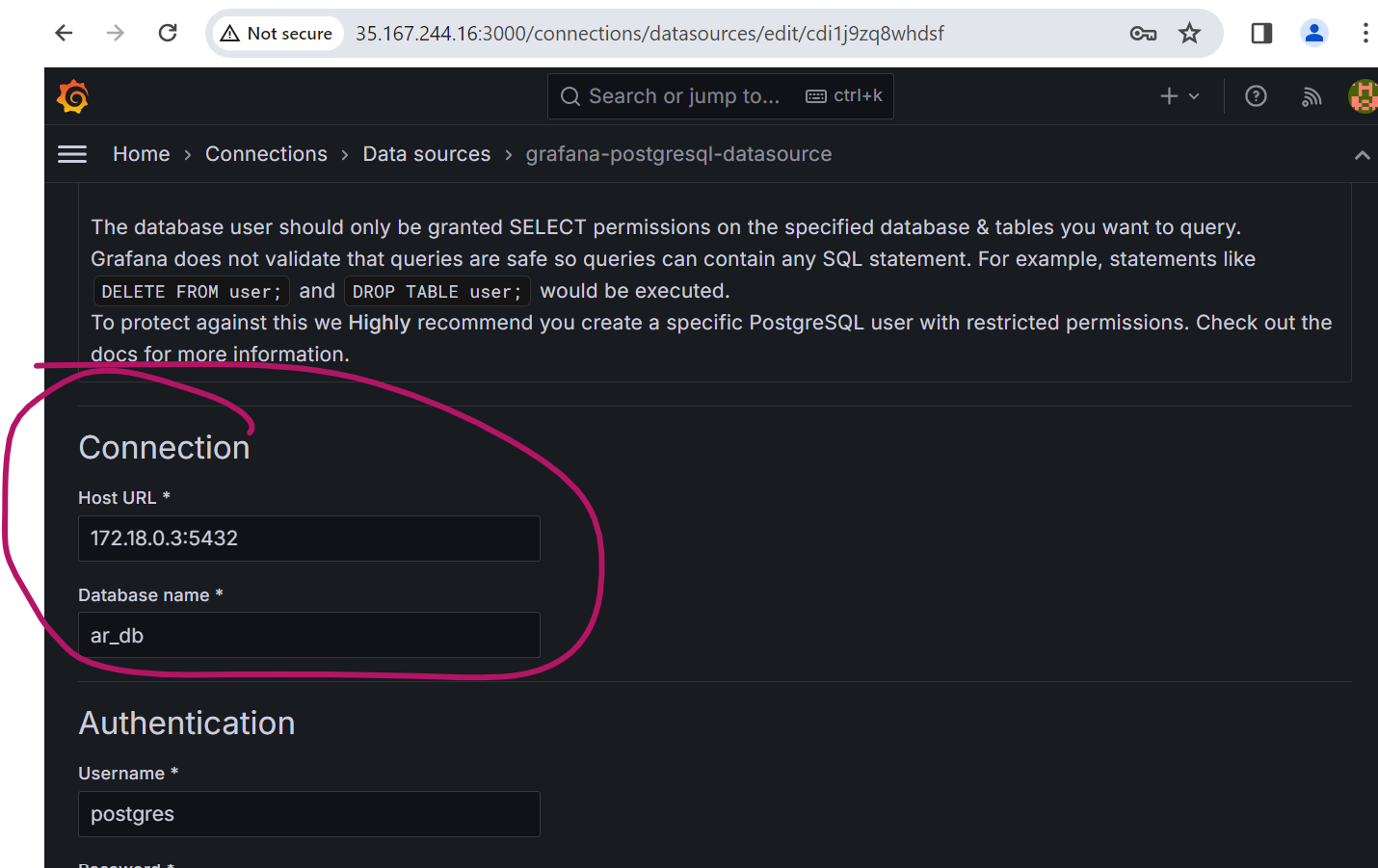
* Docker-compose build
* Docker-compose up

### Obtain docker container IP



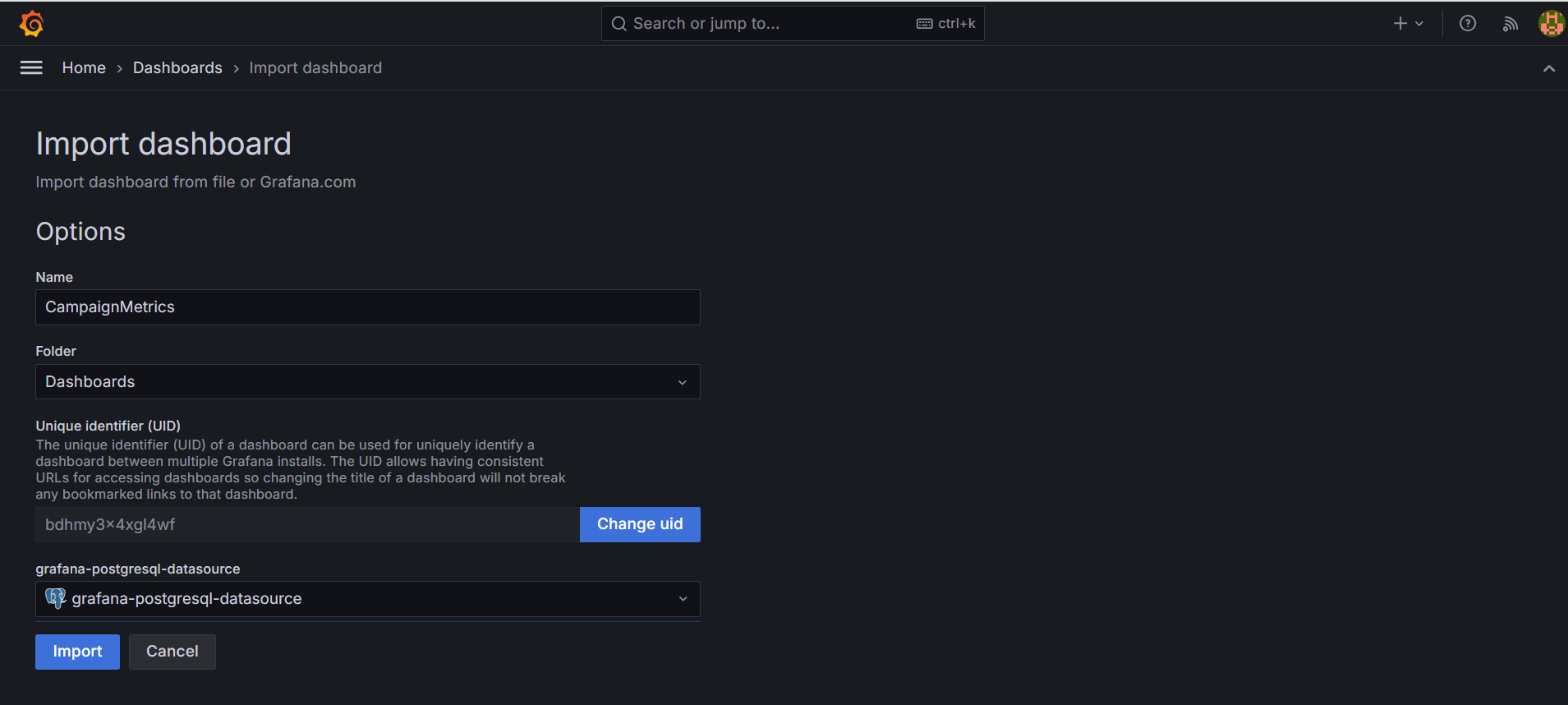


### Connecting postgres DataSource



### Importing Dashboards

Import->Upload JSON File("\server\grafana\AR\_BusinessMetrics.json , AR\_CampaignMetrics.json)

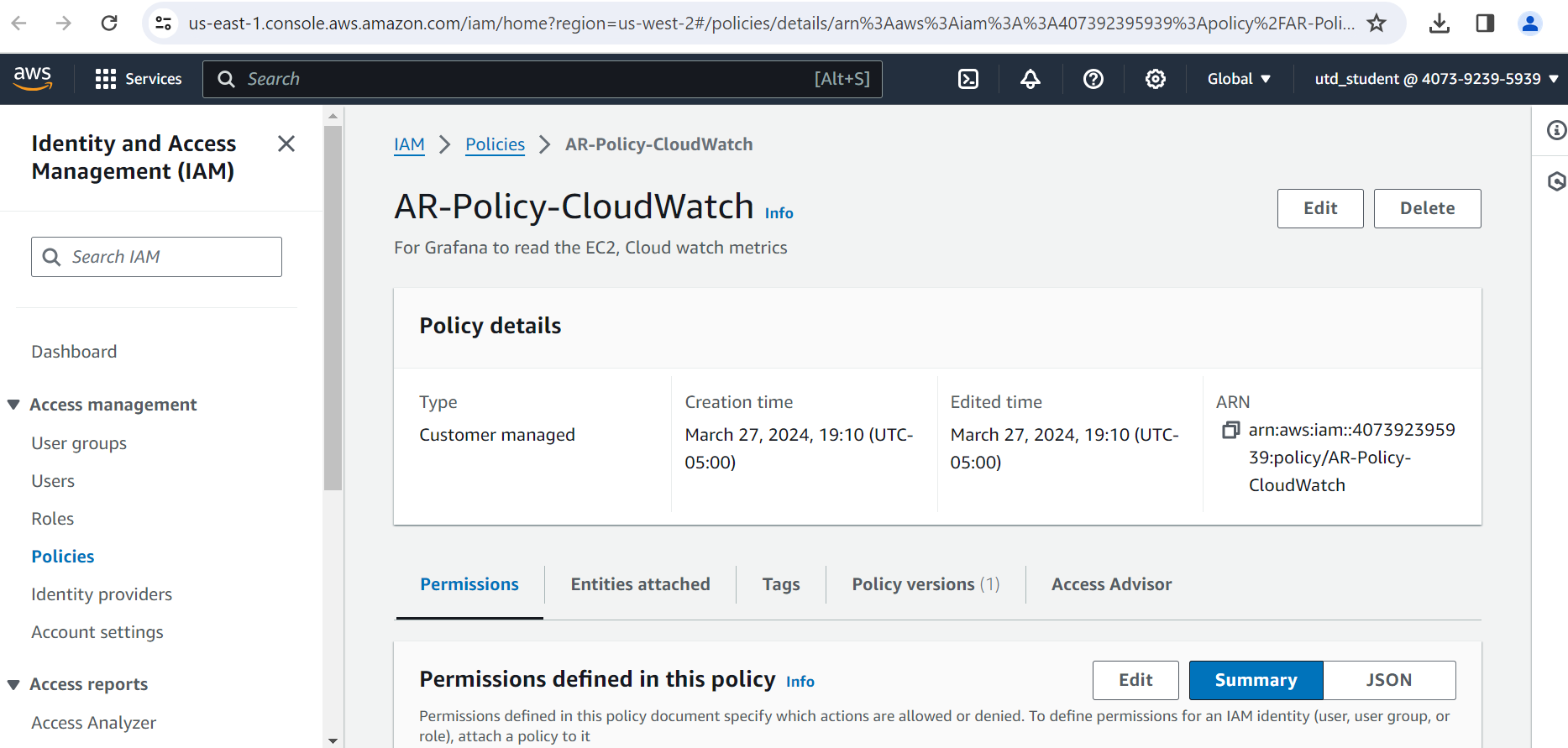


## Consume Cloud Watch metrics from Grafana

### IAM Policy

An IAM policy is required for Grafana to consume Cloud Watch Metrics.

#### Create policy



#### Sample Policy

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AllowReadingMetricsFromCloudWatch",

"Effect": "Allow",

"Action": [

"cloudwatch:ListMetrics",

"cloudwatch:GetMetricStatistics",

"cloudwatch:GetMetricData"

],

"Resource": "\*"

},

{

"Sid": "AllowReadingTagsInstancesRegionsFromEC2",

"Effect": "Allow",

"Action": [

"ec2:DescribeTags",

"ec2:DescribeInstances",

"ec2:DescribeRegions"

],

"Resource": "\*"

}

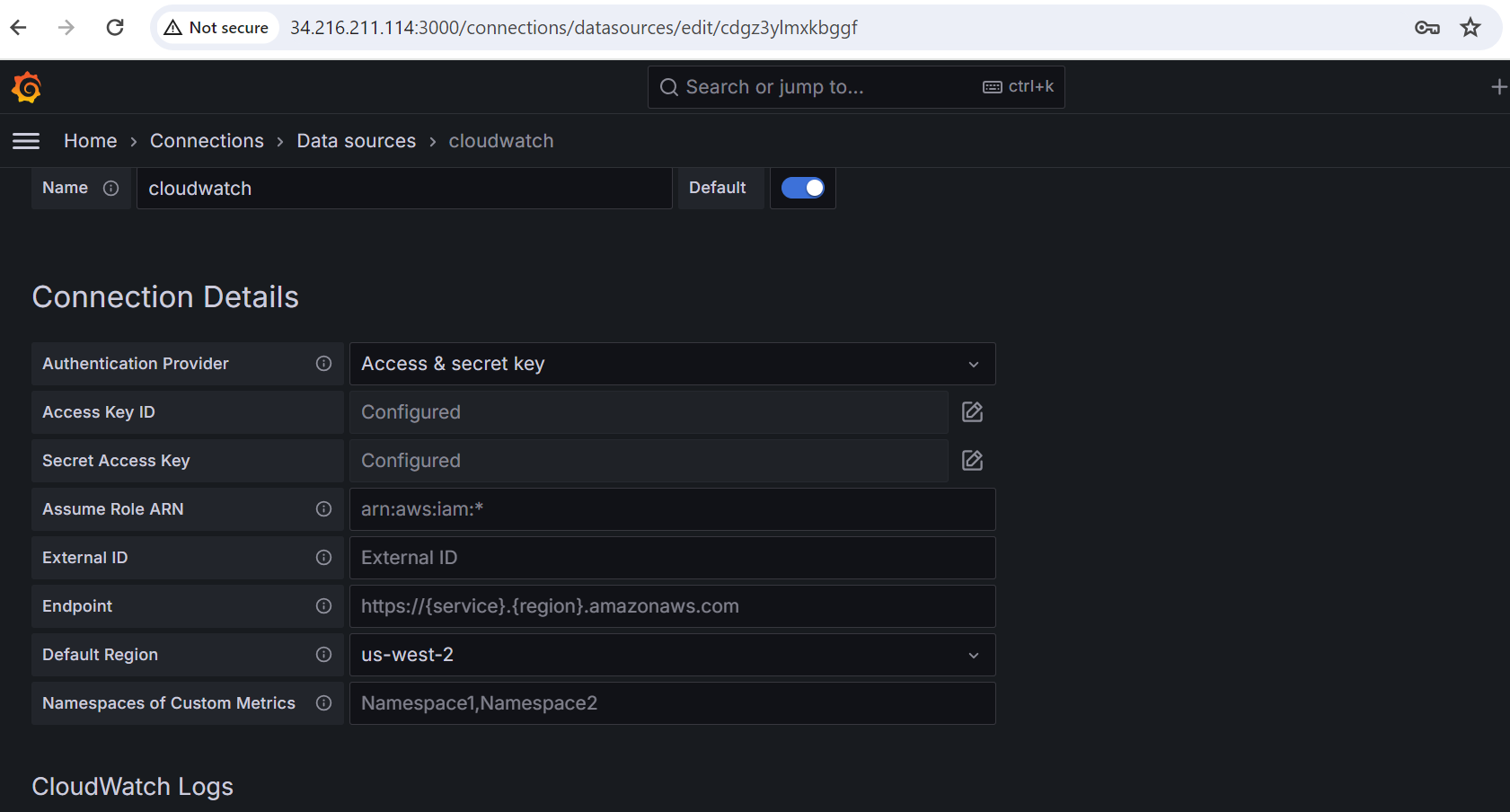
]

}

### Connecting CloudWatch to Grafana

#### Create Cloud-Watch connection

Keep handy Access and Secret key.



### Import Ec2 metrics Dashboards

* Import the gitProject\..\server\grafana\AR-EC2-SystemMetrics.json

# 

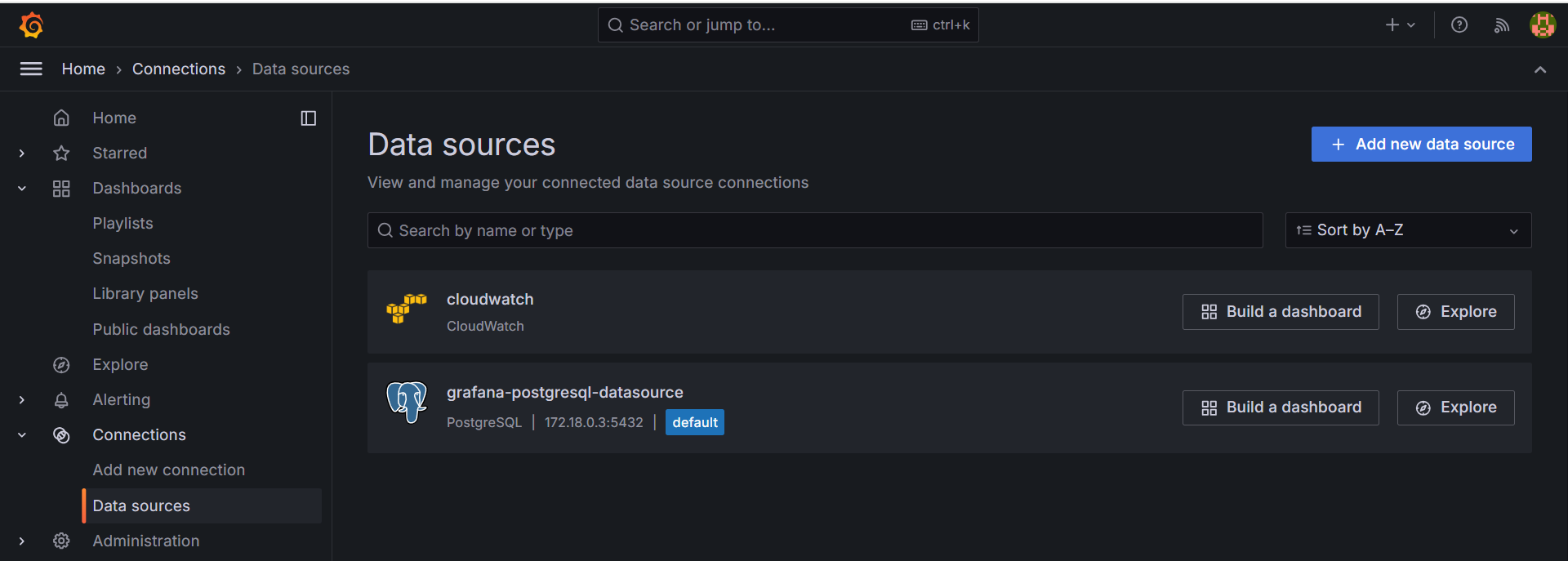
# Pending DevOps Items

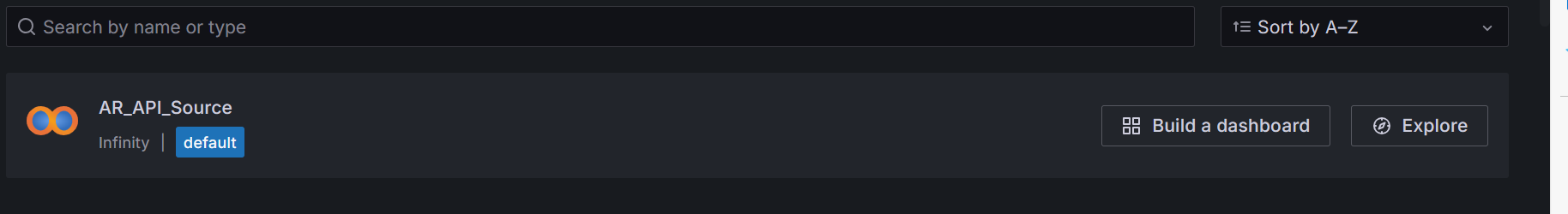
* Automate iam policy
* Automate sg groups related to grafana
* Automate creations of connections
* Automate installing dashboards in server/grafana/\*.json files

# 

# Final step

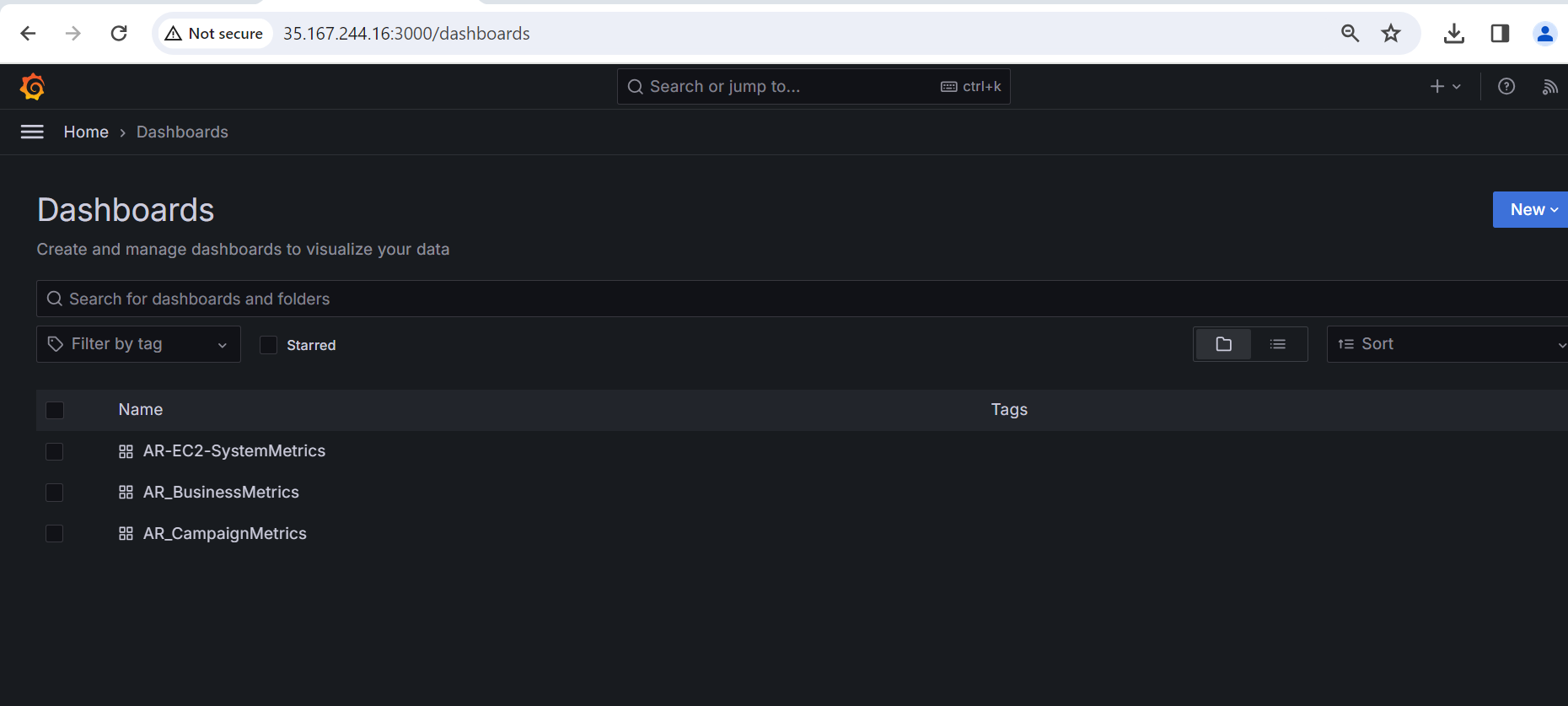
## Validate all data sources





## Dashboards

Import all json listed in server/grafana/\*.json



# 

# 