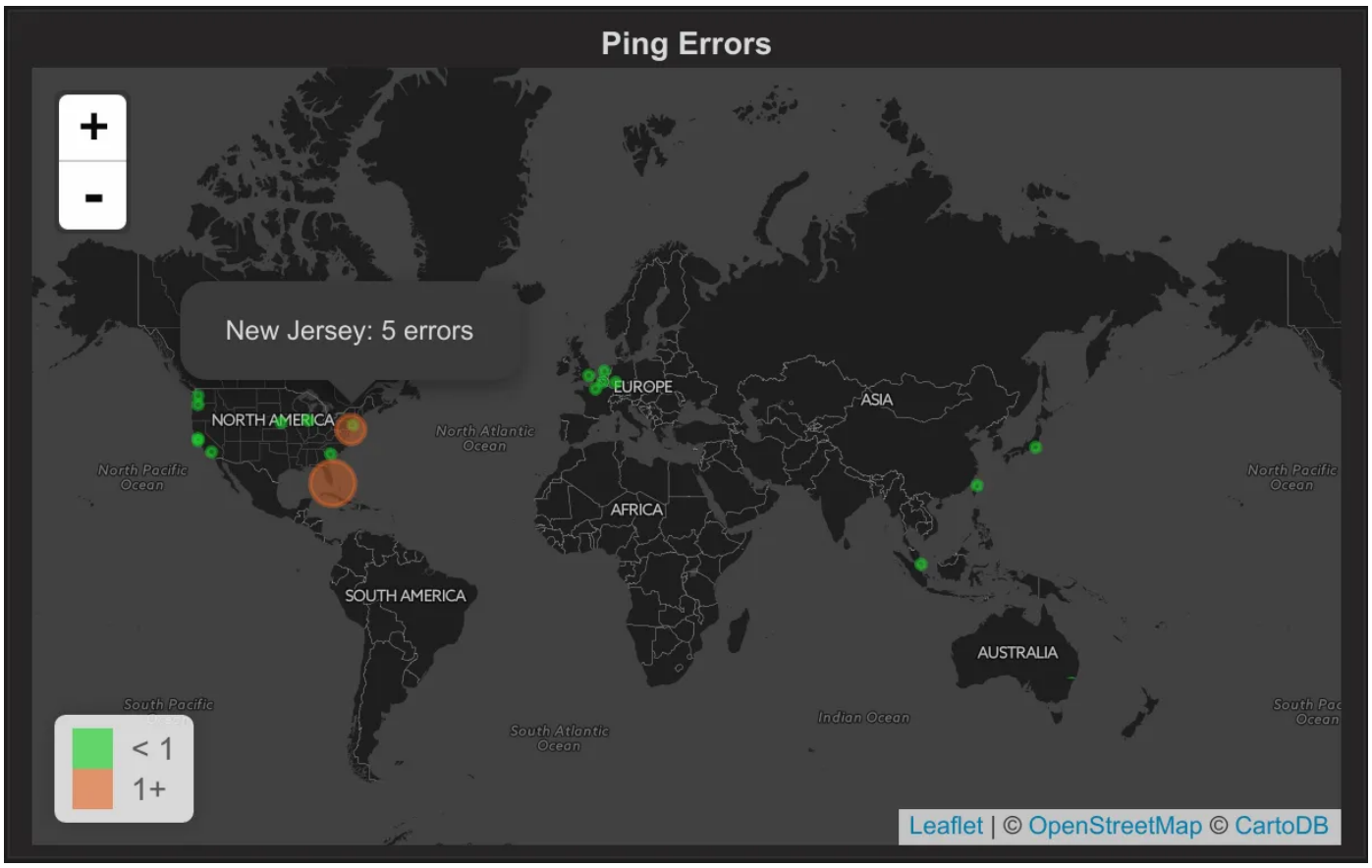
# Visualising a distributed network

## Using Worldmap Panel to pinpoint your nodes

Tom Burton · Follow
5 min read · Jul 28, 2021



When building a large distributed network for a blockchain, it's sometimes nice to see how your nodes are distributed around the world. Grafana has a plugin called Worldmap Panel. It allows you to provide a list of locations and watch them light up a map for your viewing pleasure.




https://grafana.com/grafana/plugins/grafana-worldmap-panel/

I'd like to share with you today how I generated a similar view using just the IP addresses of the nodes.

### Node Exporter, Prometheus, Grafana

Setting up and configuration of the above is outside the scope of this tutorial. We use the inbuilt Prometheus service discovery modules to automatically add nodes from the various cloud providers around the world.

### Worldmap Plugin

Depending on your Grafana setup, it may be as simple as

```
grafana-cli plugins install grafana-worldmap-panel
```

### JSON API Plugin

This plugin allows us to run JSON queries and store them as a table.

```
grafana-cli plugins install marcusolsson-json-datasource
```

### IP-API

This free API allows us to provide a list of IP addresses and get an array of latitude and longitudes in return. We can send 45 queries per minute for free, but I will show you how to make sure you don't hit that limit.

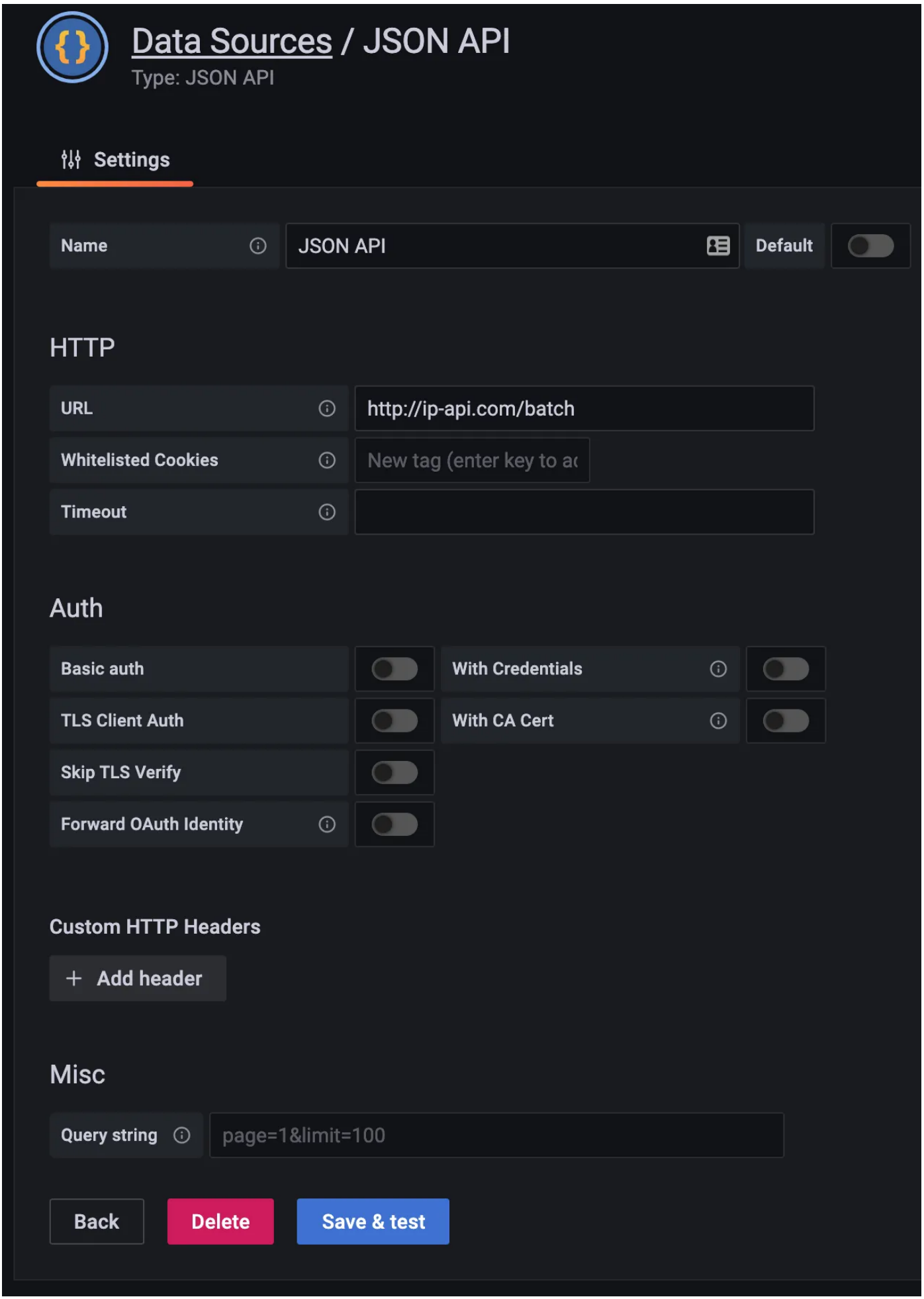
You can see the result here:

http://ip-api.com/json/1.1.1.1

### Configuring the data source

Let's glue these pieces together.

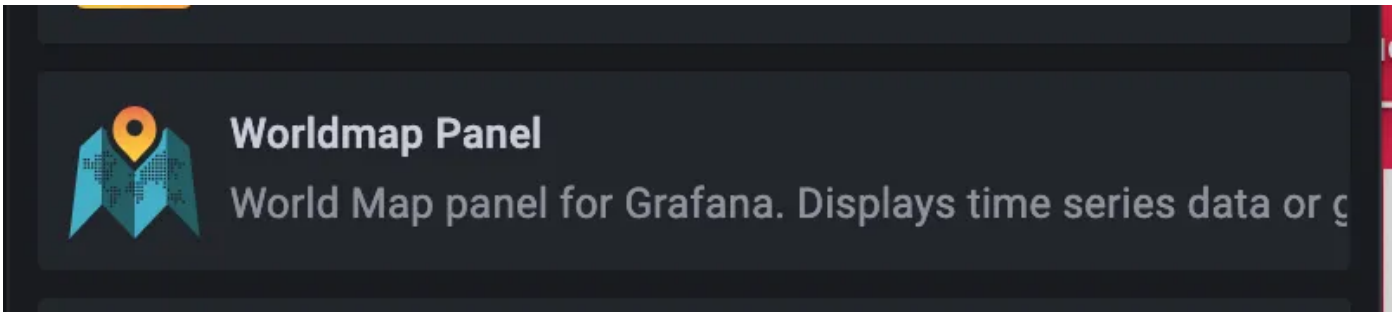
Head to the settings tab on Grafana and add a new data source. We want the JSON API data source. Fill out the details like so:


Data Sources / JSON API
Type: JSON API
Settings
Name
JSON API
Default
HTTP
URL
http://ip-api.com/batch
Whitelisted Cookies
New tag (enter key to ac
Timeout
Auth
Basic auth
With Credentials
TLS Client Auth
With CA Cert
Skip TLS Verify
Forward OAuth Identity
Custom HTTP Headers
+ Add header
Misc
Query string
page=1&limit=100
Back
Delete
Save & test

Note: you will get an error at this point. This is okay. We haven't sent data in the correct form yet.

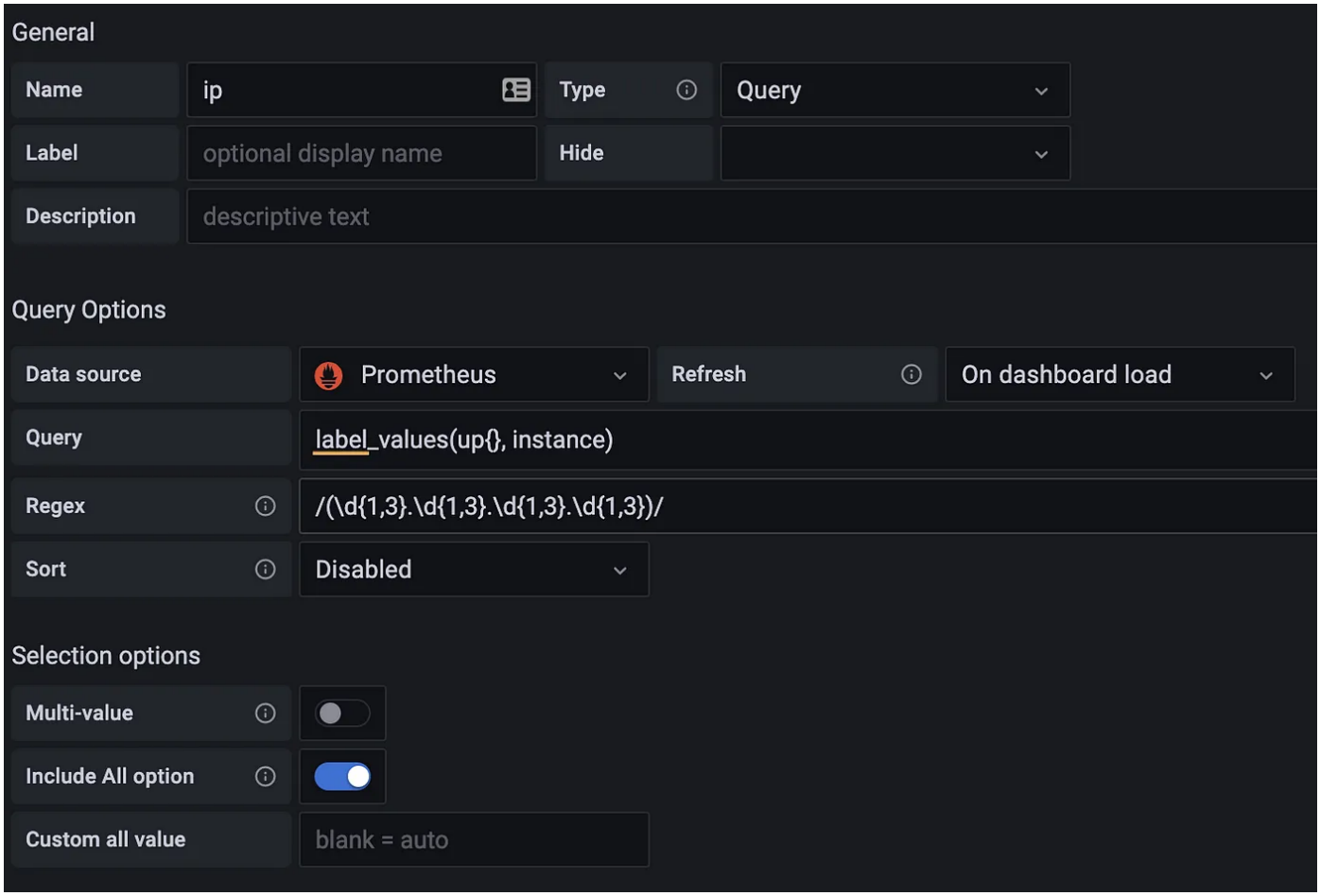
### Configuring the Dashboard

Head to the Create tab and start a new Dashboard. Add a new panel and add the Worldmap Panel.




We will need a list of IP addresses. The easiest metric to obtain this list is `up` as it includes the `instance` label. Head to your dashboard settings, scroll to variables and create a new one called `ip`. Use the `label_values` function to scrape the list of instance IPs. Use the REGEX below to drop the `:port` from the end of each IP.

```
/(\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})/
```




## Configuring the Panel

We can now start displaying this list. The Worldmap Plugin expects an array of IPs in the following format:

```
[
  {
    "key": "SE",
    "latitude": 60.128161,
    "longitude": 18.643501,
    "name": "Sweden"
  },
  {
    "key": "US",
    "latitude": 37.09024,
    "longitude": -95.712891,
    "name": "United States"
  }
]
```

However, the data we receive looks like this:

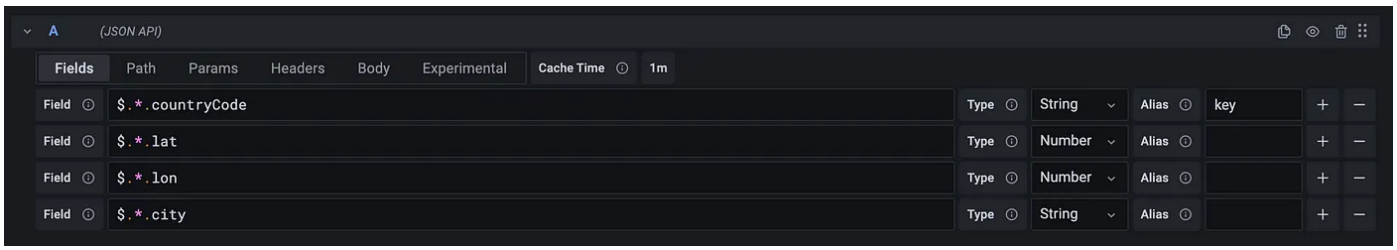

```
[
  {
    "status": "success",
    "country": "Australia",
    "countryCode": "AU",
    "region": "QLD",
    "regionName": "Queensland",
    "city": "South Brisbane",
    "zip": "4101",
    "lat": -27.4766,
    "lon": 153.0166,
    "timezone": "Australia/Brisbane",
    "isp": "Cloudflare, Inc",
    "org": "APNIC and Cloudflare DNS Resolver project",
    "as": "AS13335 Cloudflare, Inc.",
    "query": "1.1.1.1"
  }
]
```


The panel allows us to map the required fields based on the incoming data.

On your newly created dashboard, edit the Worldmap Panel. Add a new data source of the JSON API. Fill out each tab as follows:

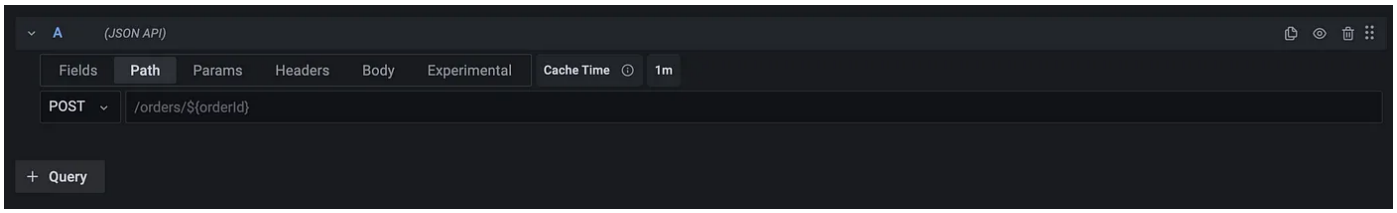
⚠️*: Make sure to set the Cache Time to 1m otherwise you will get rate limited by IP-API.*

**Fields**



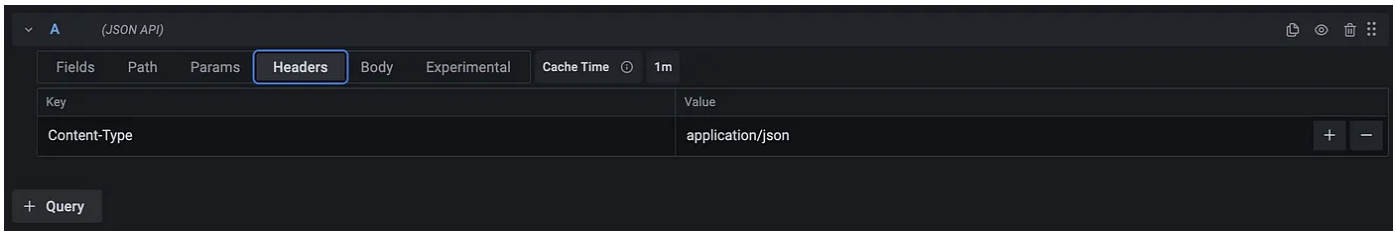
Notice the `Alias` for `countryCode`.

## Path



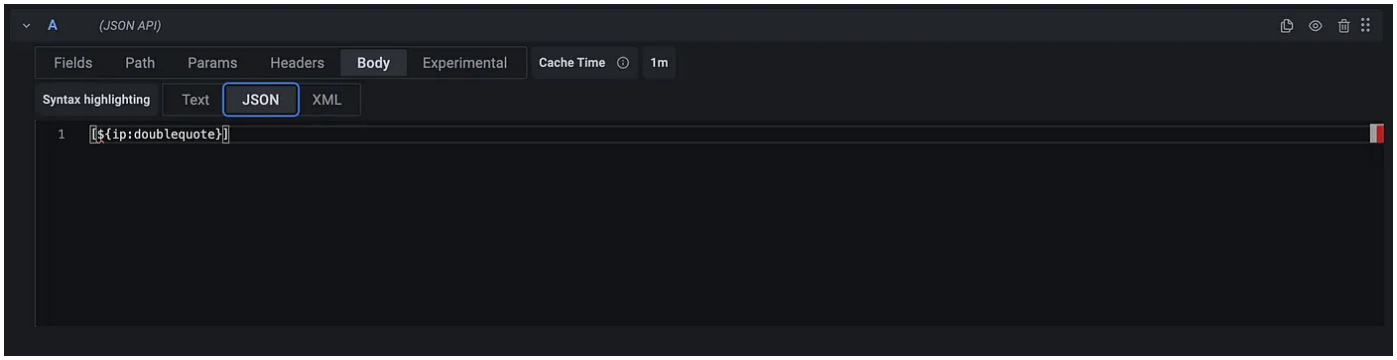

Make sure to set the request to `POST`.

## Headers




```
Content-Type=application/json
```

## Body



```
[${ip:doublequote}]
```

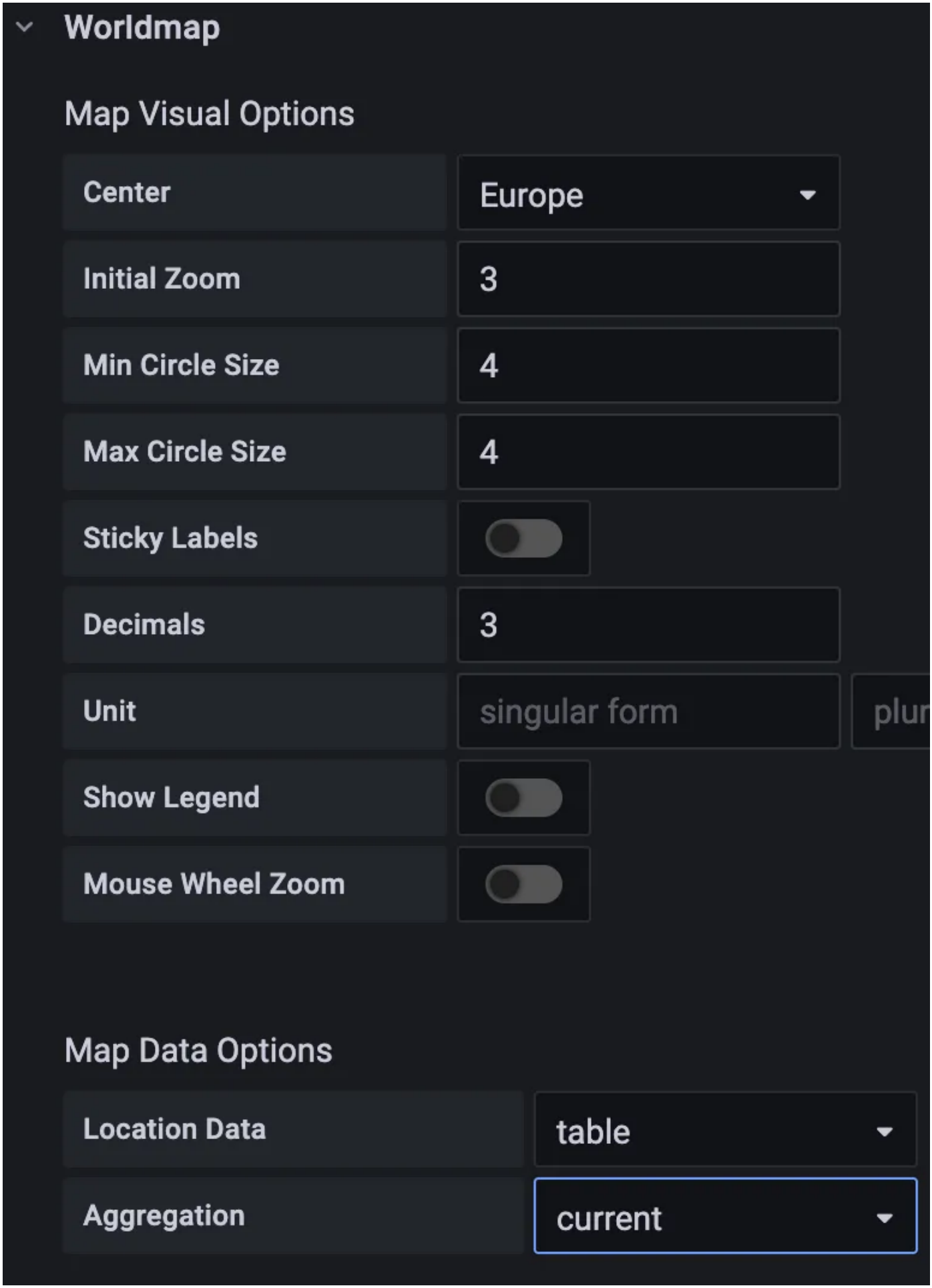
What we're doing here is transforming our list of IPs into a comma separated, double-quoted array. The variable we set earlier returns the list of IPs like so:

```
"123.32.424.24, 52.34.63.63"
```

After using the Grafana variable format manipulations, we get:

```
"123.32.424.24", "52.34.63.63"
```

We need to make some edits to the Panel itself. Fill out the following options:

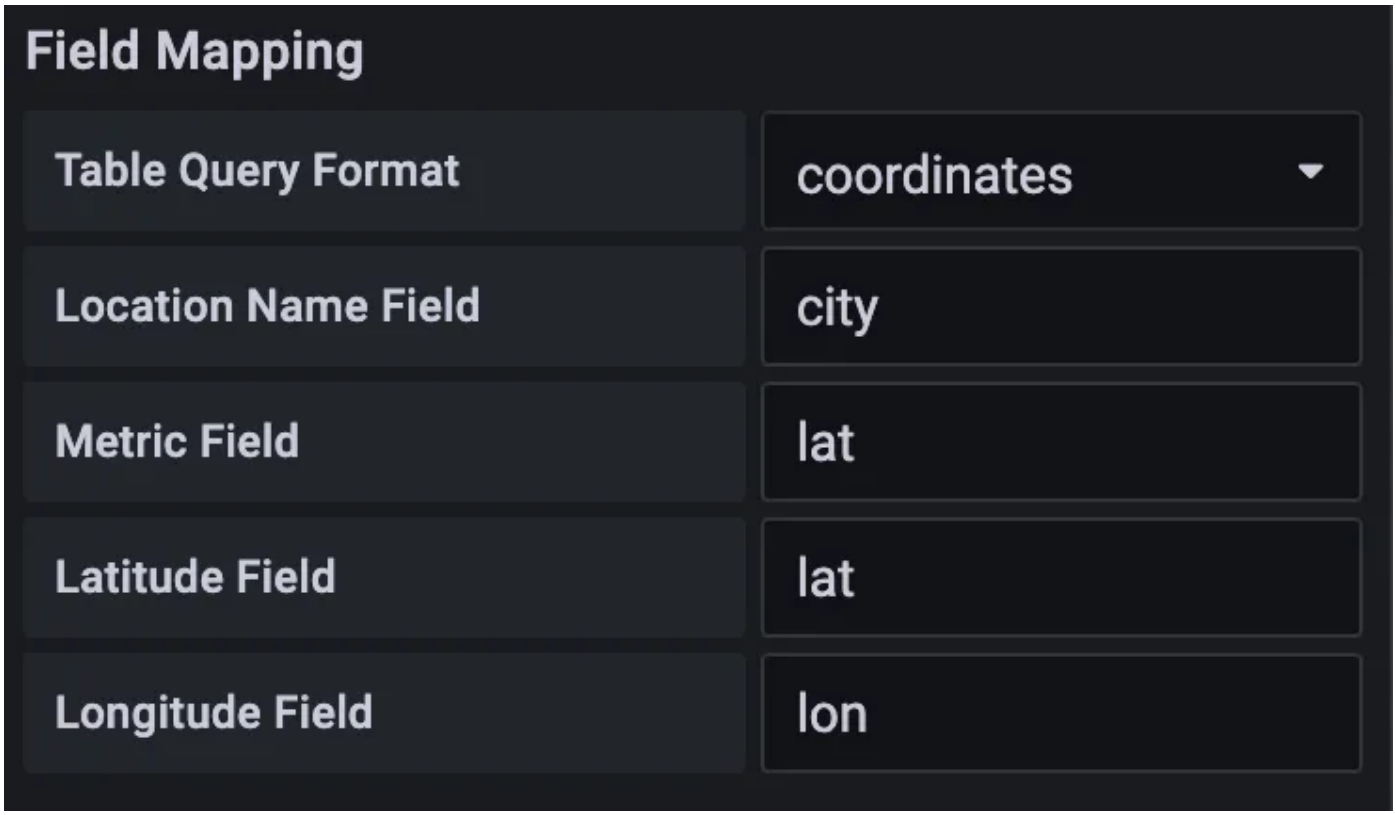



You can of course tweak these as you go, but the most important setting is

`Location Data` as this tells the panel what format to expect the data in.

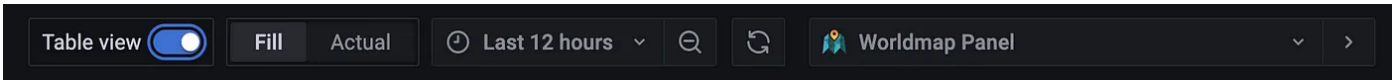
As the IP-API doesn't return any metric that you can use to define the size of the circle, I put the `Max Circle Size` and `Min Circle Size` to equal the same.

Finally, configure the `Field Mapping` section like so:

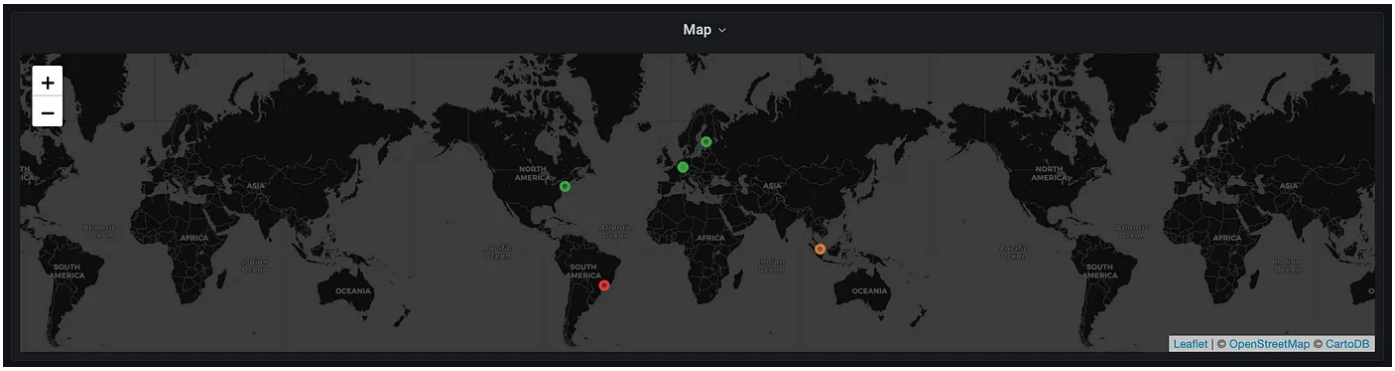



The `Metric Field` can be left blank if you wish.

Switch over to `Table View` to verify the data is coming in. If you still can't see the data, head over to your browsers `Inspect` window and check the `Network` tab. You should see a failing request. Most likely the data you're sending is malformed, or you have somehow hit the rate limit. Run through the instructions again.




If you've done everything correctly, you should end up with a lovely map like so:




Thanks for reading!

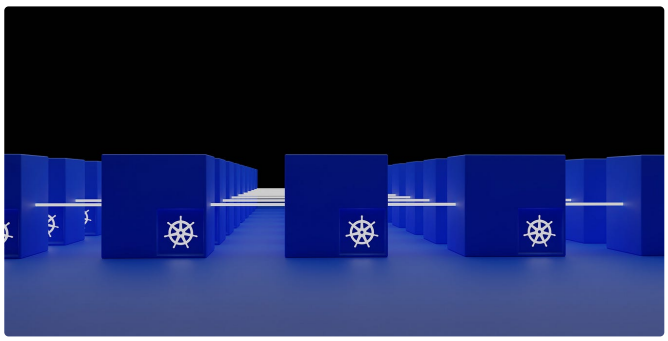
Grafana Maps Prometheus DevOps Monitoring

## Written by Tom Burton

9 Followers

Follow

## Recommended from Medium



Pavel Glukhikh

### Deploying a Production Kubernetes Cluster in 2023—A...

A batteries included guide on deploying a production-ready Kubernetes cluster with...

21 min read · Aug 25

82 1


Okan Okumuşoğlu in Valensas

### Monitoring—Basics Of Prometheus, Grafana

Introduction

7 min read · Aug 1

63

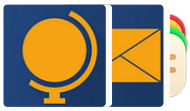
## Lists

## General Coding Knowledge

20 stories · 728 saves

## Icon Design

34 stories · 188 saves

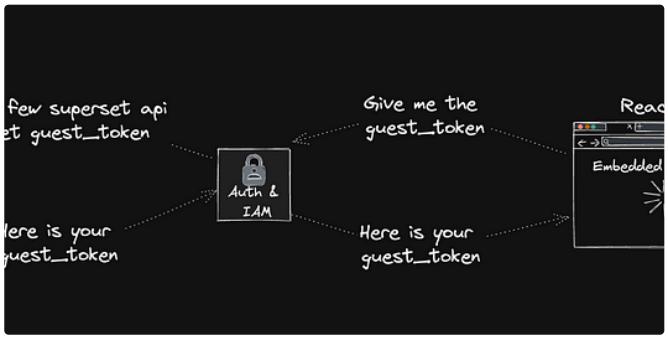


## Productivity

234 stories · 251 saves


## Staff Picks

544 stories · 577 saves




Michael Chen

## End-to-end example of setting up superset embedded dashboard

Hello, this marks the beginning of my journey in sharing insights on Machine...

8 min read · Aug 30

30 1


Rkssh - DevOps as a service plateform

## Simplify Log Analysis with Signoz: A Step-by-Step Guide

Introduction

3 min read · Aug 3

1 1

Samirsaha

## Observability—using Telegraf, InfluxDB & Grafana in Kubernetes

‘Keeping an eye on’ the application, product or platform works as a catalyst in service...

8 min read · Aug 21

1

Athan Bonis in tech.thesignalgroup

## Integration Tests with In-Memory MongoDB

Adding Integration Tests to your API is a crucial step towards a robust system....

9 min read · Dec 20

3

See more recommendations