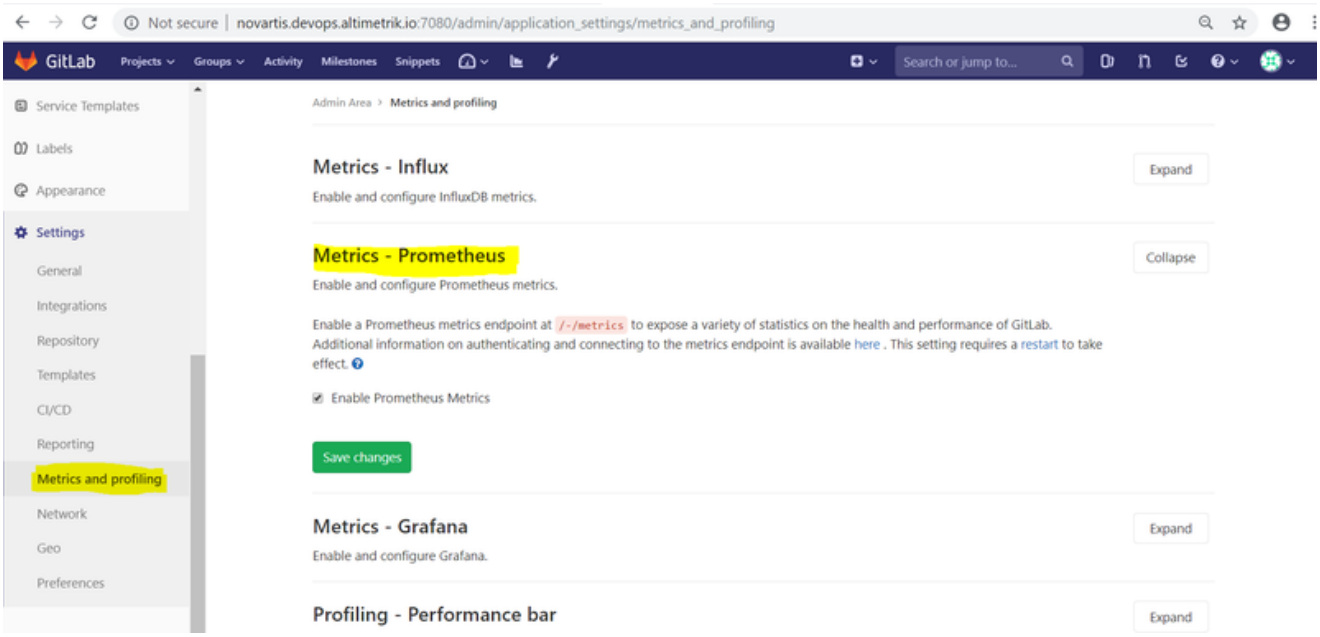
Configure Prometheus and Grafana for GitLabEE

Installation and upgrade:

1. Log into GitLab as an administrator, and go to the Admin area.
2. Navigate to GitLab's **Settings > Metrics and profiling**.
3. Find the **Metrics - Prometheus** section, and click **Enable Prometheus Metrics**.
4. Restart GitLab for the changes to take effect



Configuring Prometheus

1. Edit /data/gitlabee/config/`gitlab.rb`.

   Disable the bundled Prometheus:

```
prometheus['enable'] = false
```

2. Changing the port and address Prometheus listens on

   Edit /data/gitlabee/config/`gitlab.rb`

```
prometheus['listen_address'] = ':9090'
# or
prometheus['listen_address'] = '0.0.0.0:9090'
```

   Save the file and reconfigure GitLab for the changes to take effect

3. Set each bundled service's exporter to listen on a network address, for example:

```
gitlab_monitor['listen_address'] = '0.0.0.0'
sidekiq['listen_address'] = '0.0.0.0'
gitlab_monitor['listen_port'] = '9168'
node_exporter['listen_address'] = '0.0.0.0:9100'
redis_exporter['listen_address'] = '0.0.0.0:9121'
postgres_exporter['listen_address'] = '0.0.0.0:9187'
gitaly['prometheus_listen_addr'] = "0.0.0.0:9236"
gitlab_workhorse['prometheus_listen_addr'] = "0.0.0.0:9229"
```

4. To scrape nginx metrics, you'll also need to configure nginx to allow the Prometheus server IP. For example:

```
nginx['status']['options'] = {
      "server_tokens" => "off",
      "access_log" => "off",
      "allow" => "192.168.0.1",
      "deny" => "all",
}
```

Usage:

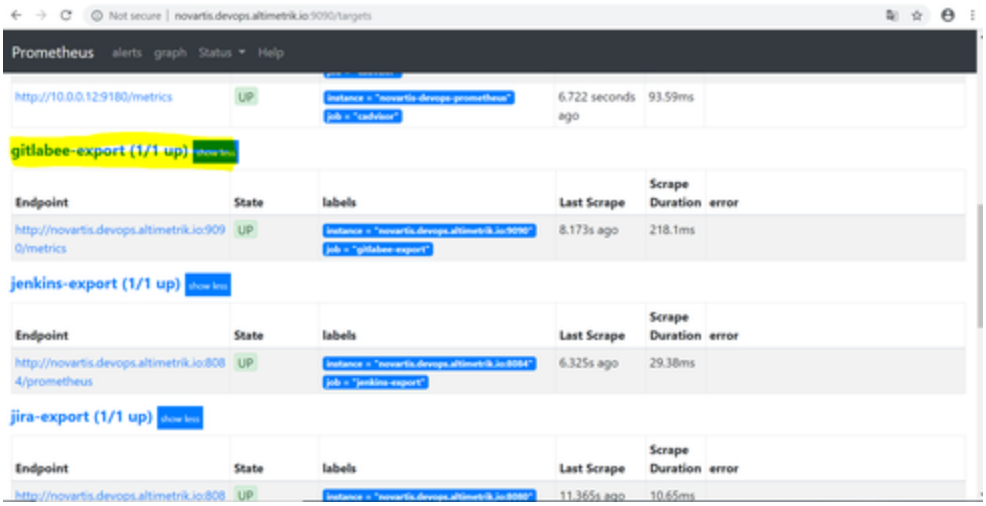
Add each node's exporters to the Prometheus server's ,The *prometheus.yml* settings looks like:

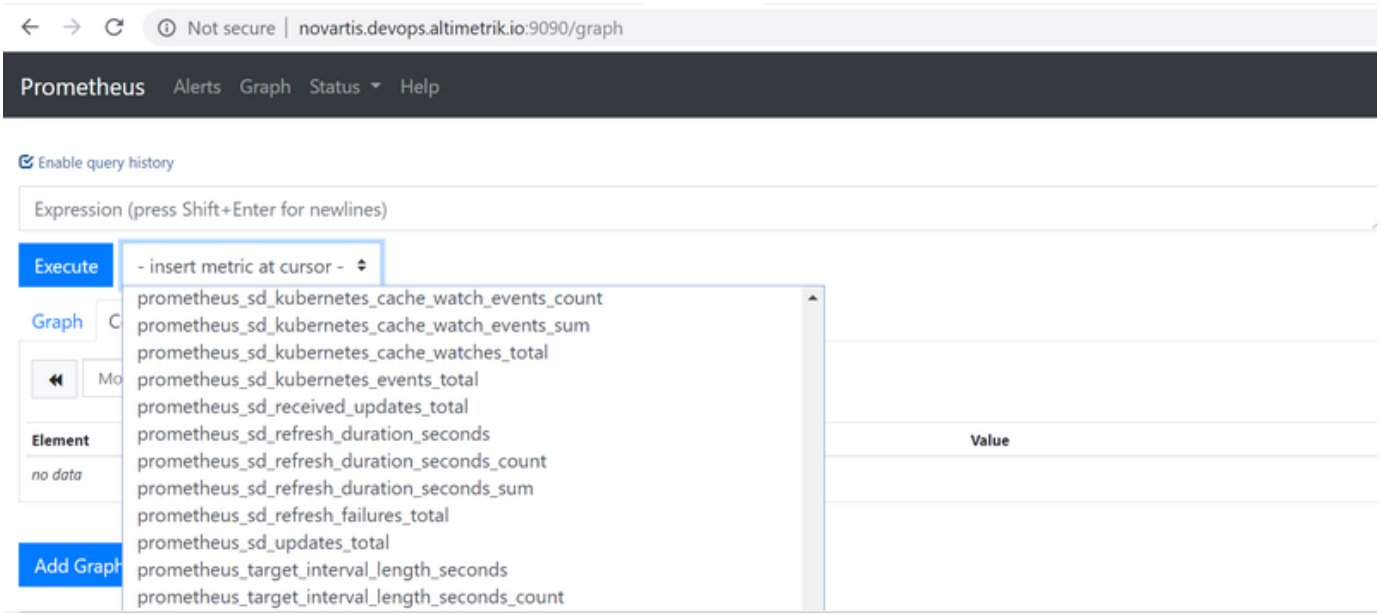
/data/prometheus/config/prometheus.yaml

```
- job_name: gitlabee-exporter
  honor_timestamps: true
  scrape_interval: 15s
  scrape_timeout: 10s
  #metrics_path: /metrics
  scheme: http
  ec2_sd_configs:
  - endpoint: ""
    region: us-east-1
    access_key: AKIAXZXIY4Q4M3JKI4TO
    secret_key: <secret>
    profile: arn:aws:iam::536285340728:user/devplatarn
    refresh_interval: 1m
    port: 9090
    filters: []
  relabel_configs:
  - source_labels: [__meta_ec2_tag_Name]
    separator: ;
    regex: ^novartis-devops-bitbucket$
    replacement: $1
    action: keep
  - source_labels: [__meta_ec2_public_ip]
    separator: ;
    regex: (.+)
    target_label: __address__
    replacement: novartis.devops.altimetrik.io:9090
    action: replace
```

Reload the Prometheus server.

Installation and setup is finished. You can collect metrics. Prometheus will be available at http://novartis.devops.altimetrik.io:9090/

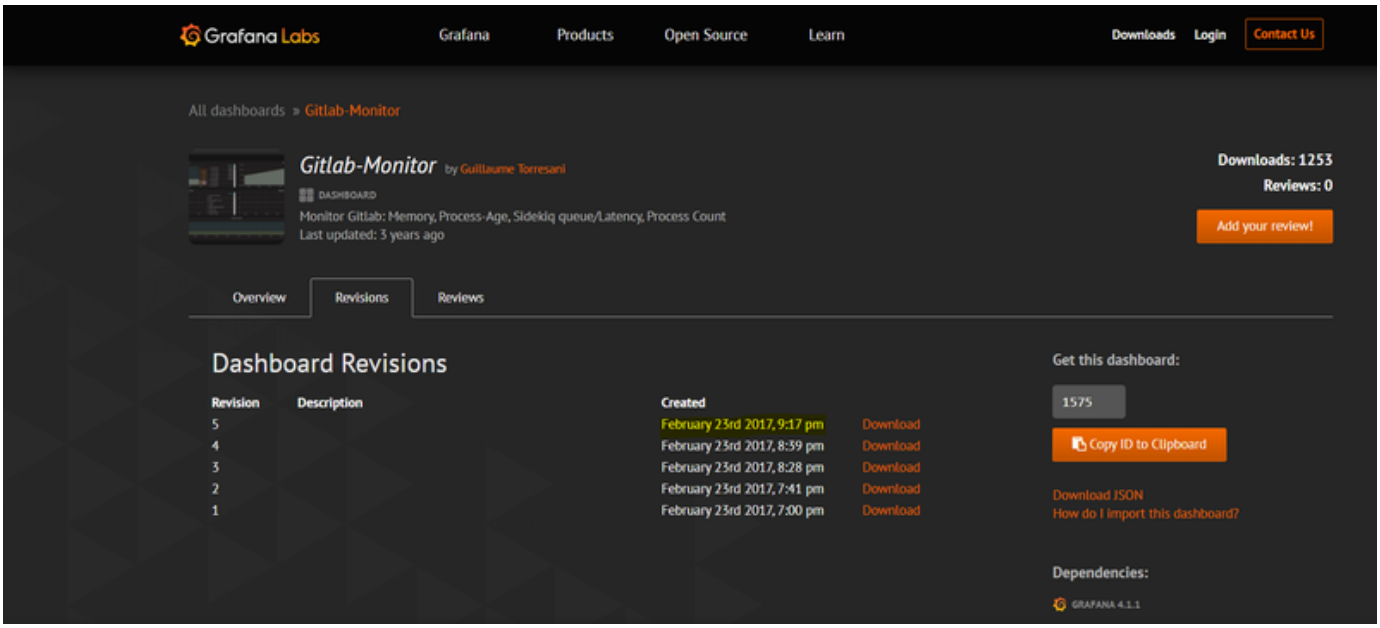




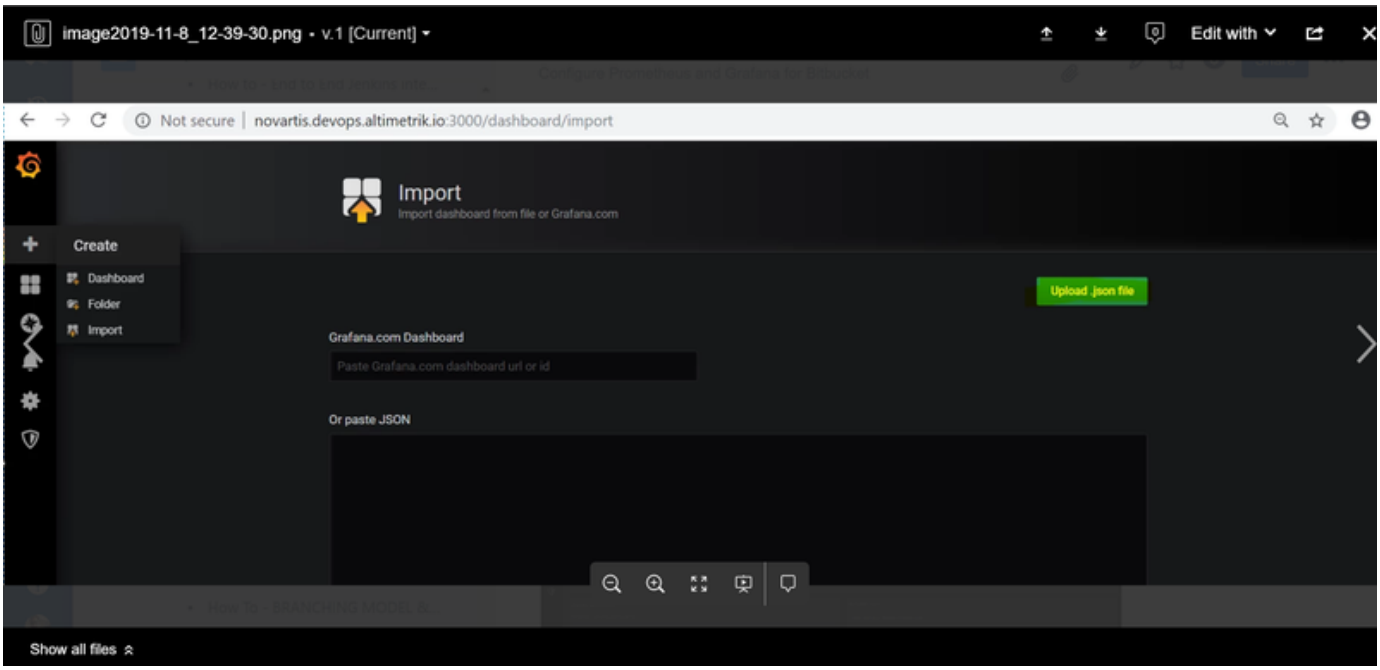
Grafana Configuration

We will use Grafana to visualize metrics stored in Prometheus. There are a couple of example dashboards in the official site https://grafana.com/grafana/dashboards

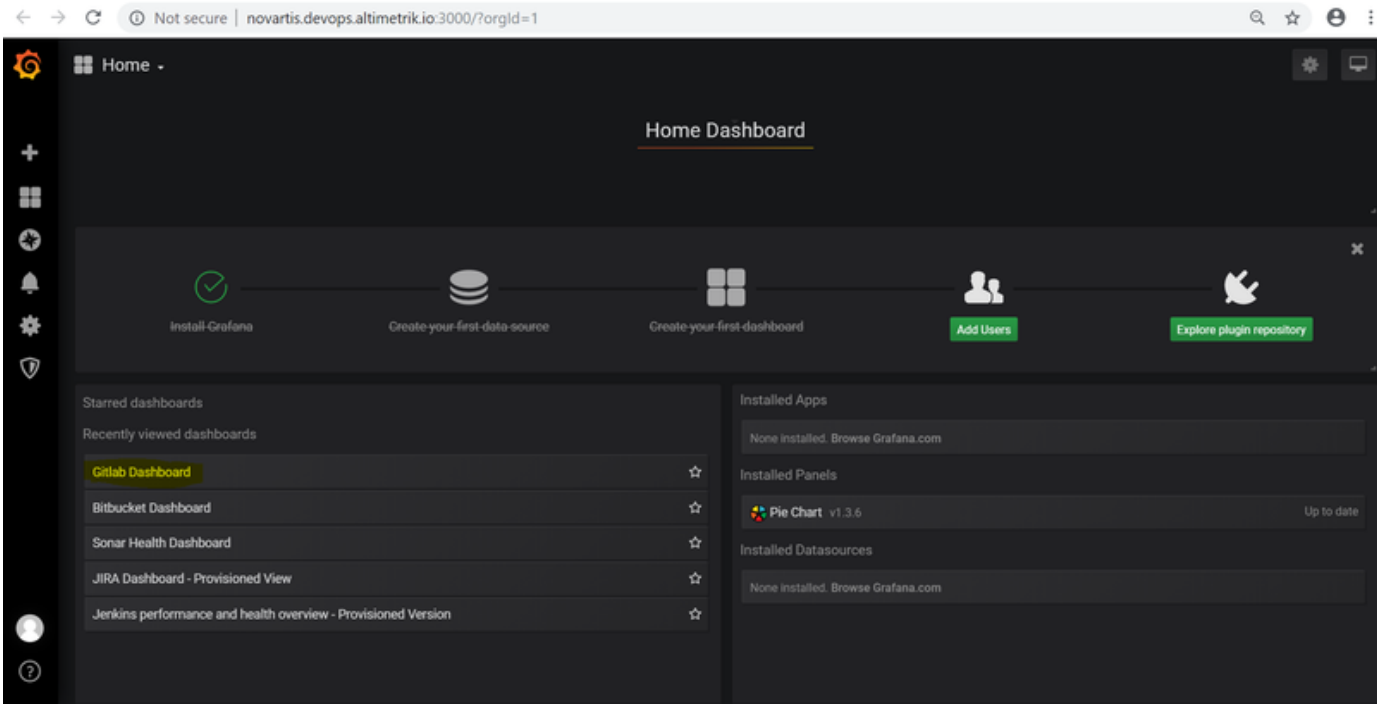
Download the latest release of Grafana for your platform, then extract it:



Go to Grafana interface at http://novartis.devops.altimetrik.io:3000/dashboard/import

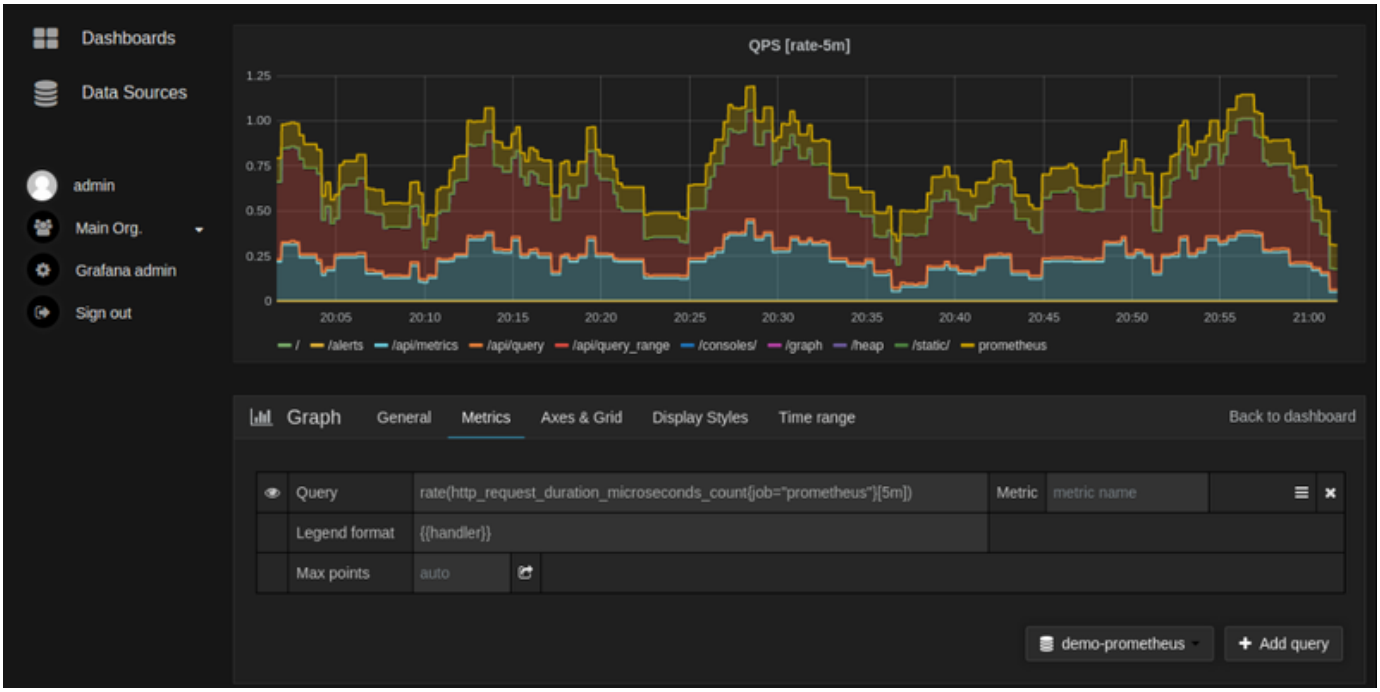


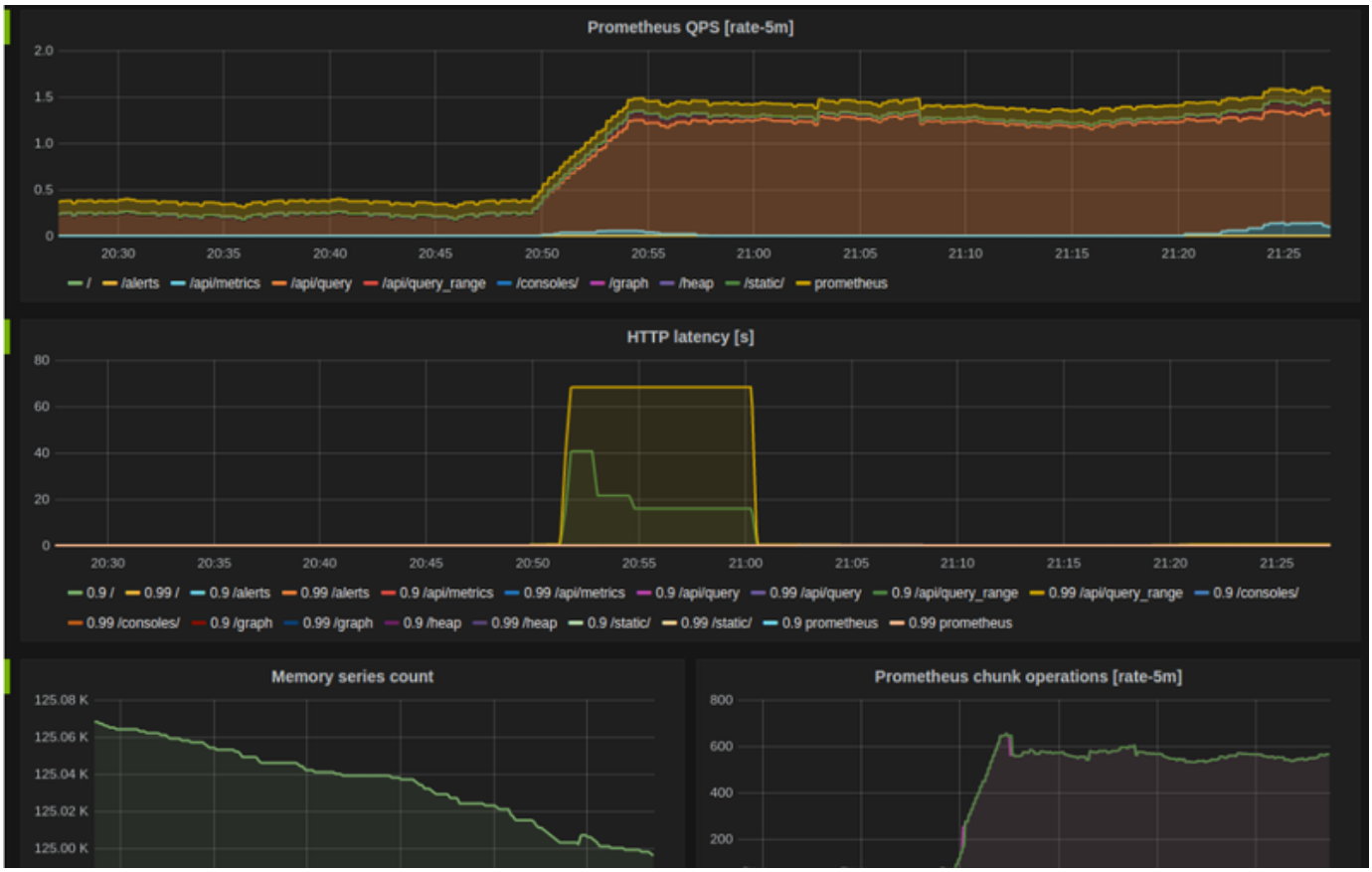
Then, you can add a new Dashboard that will get the data from the previous created Data-Source.



After a couple of minutes you will be able to view your metrics on the Dashboard. You can also add new panels to the Dashboard.

You can also build one Dashboard to tack all your Atlassian tools in one single Dashboard.




Prometheus QPS [rate-5m]
2.0
1.5
1.0
0.5
0
20:30
20:35
20:40
20:45
20:50
20:55
21:00
21:05
21:10
21:15
21:20
21:25
/
/alerts
/api/metrics
/api/query
/api/query_range
/consoles/
/graph
/heap
/static/
prometheus
HTTP latency [s]
80
60
40
20
0
20:30
20:35
20:40
20:45
20:50
20:55
21:00
21:05
21:10
21:15
21:20
21:25
0.9 /
0.99 /
0.9 /alerts
0.99 /alerts
0.9 /api/metrics
0.99 /api/metrics
0.9 /api/query
0.99 /api/query
0.9 /api/query_range
0.99 /api/query_range
0.9 /consoles/
0.99 /consoles/
0.9 /graph
0.99 /graph
0.9 /heap
0.99 /heap
0.9 /static/
0.99 /static/
0.9 prometheus
0.99 prometheus
Memory series count
125.08 K
125.06 K
125.04 K
125.02 K
125.00 K
Prometheus chunk operations [rate-5m]
800
600
400
200