# Installing Grafana locally

Grafana is the primary tool that is used to build monitoring and operational dashbaords for our services.

ATG provides a dev and production version of Grafana via OMF.

There are occasions where one might want to pursue work locally with Grafana and Prometheus. This document outlines how to setup grafana and. promethues locally.

There are multiple ways of going about installing Grafana and Prometheus.  This document outlines one such way - using docker.

## Docker - Install and setup

1. Refer to Docker Getting Started for more details on docker setup
2. Install DockerDesktop on your system as a pre requisite.
3. Verify that docker is up and running

## Grafana - Install and setup

1. Setup a docker volume where grafana can store data.  This is an optional step and is required only if you want to persist grafana data across restarts.
    a. docker volume grafana-storage
2. Use the latest image of grafana to startup a docker container
    a. docker run d— name grafana -p 3000:3000  -v grafana-storage:/var/lib/grafana grafana/grafana
3. Point your browser to http://localhost:3000 to access grafana
4. Remember to change the default password for admin (admin/admin).
5. Once Prometheus is setup, we will come back and setup a data source.

## Prometheus - Install and setup

1. Setup a docker volume where prometheus can store data. This is an optional step and is required only if you want to persist prometheus data across restarts.
    a. docker volume prometheus-data
2. Create a prometheus configuration file (say in ~/prometheus.yml)
    a. Sample prometheus file is provided below.  Change the target as appropriate to point to the right micro service that we want to scrap metrics from

**~/prometheus.yml**

```
# my global config
global:
  scrape_interval:     15s # Set the scrape interval to every 15 seconds. Default is every 1
minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
  - static_configs:
    - targets:
      # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: 'prometheus'

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
    - targets: ['localhost:9090']
```

3. Use the latest image of prometheus to startup a docker container (
    a. docker run -d --name prometheus -p 9090:9090 -v ~/prometheus.yml:/etc/prometheus/prometheus.yml -v promethus-data:/prometheus prom/prometheus
4. Point. your browser to http://localhost:9090 to access prometheus