

Grafana High Availability (HA) test setup

A set of docker compose services which together creates a Grafana HA test setup with capability of easily scaling up/down number of Grafana instances.

Included services

- Grafana
- Mysql - Grafana configuration database and session storage
- Prometheus - Monitoring of Grafana and used as data source of provisioned alert rules
- Nginx - Reverse proxy for Grafana and Prometheus. Enables browsing Grafana/Prometheus UI using a hostname

Prerequisites

Build grafana docker container

Build a Grafana docker container from current branch and commit and tag it as grafana/grafana:dev.

```
$ cd <grafana repo>
$ make build-docker-full
```

Virtual host names

Alternative 1 - Use dnsmasq

```
$ sudo apt-get install dnsmasq
$ echo 'address=/loc/127.0.0.1' | sudo tee /etc/dnsmasq.d/dnsmasq-loc.conf > /dev/null
$ sudo /etc/init.d/dnsmasq restart
$ ping whatever.loc
PING whatever.loc (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.076 ms
--- whatever.loc ping statistics ---
1 packet transmitted, 1 received, 0% packet loss, time 1998ms
```

Alternative 2 - Manually update /etc/hosts

Update your `/etc/hosts` to be able to access Grafana and/or Prometheus UI using a hostname.

```
$ cat /etc/hosts
127.0.0.1      grafana.loc
127.0.0.1      prometheus.loc
```

Start services

```
$ docker-compose up -d
```

Browse

- <http://grafana.loc/>
- <http://prometheus.loc/>

Check for any errors

```
$ docker-compose logs | grep error
```

Scale Grafana instances up/down

Scale number of Grafana instances to `<instances>`

```
$ docker-compose up --scale grafana=<instances> -d
# for example 3 instances
$ docker-compose up --scale grafana=3 -d
```

Test alerting

Create notification channels

Creates default notification channels, if not already exists

```
$ ./alerts.sh setup
```

Slack notifications

Disable

```
$ ./alerts.sh slack -d
```

Enable and configure url

```
$ ./alerts.sh slack -u https://hooks.slack.com/services/...
```

Enable, configure url and enable reminders

```
$ ./alerts.sh slack -u https://hooks.slack.com/services/... -r -e 10m
```

Provision alert dashboards with alert rules

Provision 1 dashboard/alert rule (default)

```
$ ./alerts.sh provision
```

Provision 10 dashboards/alert rules

```
$ ./alerts.sh provision -a 10
```

Provision 10 dashboards/alert rules and change condition to `gt > 100`

```
$ ./alerts.sh provision -a 10 -c 100
```

Pause/unpause all alert rules

Pause

```
$ ./alerts.sh pause
```

Unpause

```
$ ./alerts.sh unpause
```