+++ title = "Live HA setup" description = "Grafana Live HA setup guide" keywords = ["Grafana", "live", "guide", "ha"] weight = 130 +++

Configure Grafana Live HA setup

By default, Grafana Live uses in-memory data structures and in-memory PUB/SUB hub for handling subscriptions.

In a high availability Grafana setup involving several Grafana server instances behind a load balancer, you can find the following limitations:

- Built-in features like dashboard change notifications will only be broadcasted to users connected to the same Grafana server process instance.
- Streaming from Telegraf will deliver data only to clients connected to the same instance which received Telegraf data, active stream cache is not shared between different Grafana instances.
- A separate unidirectional stream between Grafana and backend data source may be opened on different Grafana servers for the same channel.

To bypass these limitations, Grafana v8.1 has an experimental Live HA engine that requires Redis to work.

Configure Redis Live engine

When the Redis engine is configured, Grafana Live keeps its state in Redis and uses Redis PUB/SUB functionality to deliver messages to all subscribers throughout all Grafana server nodes.

Here is an example configuration:

```
[live]
ha_engine = redis
ha_engine_address = 127.0.0.1:6379
```

For additional information, refer to the [ha_engine]({{< relref "../administration/configuration.md#ha_engine" >}}) and [ha_engine_address]({{< relref "../administration/configuration.md#ha_engine_address" >}}) options.

After running:

- All built-in real-time notifications like dashboard changes are delivered to all Grafana server instances and broadcasted to all subscribers.
- Streaming from Telegraf delivers messages to all subscribers.
- A separate unidirectional stream between Grafana and backend data source opens on different Grafana servers. Publishing data to a channel delivers messages to instance subscribers, as a result, publications from different instances on different machines do not produce duplicate data on panels.

At the moment we only support single Redis node.

> ***Note:*** *It's possible to use Redis Sentinel and Haproxy to achieve a highly available Redis setup. Redis nodes should be managed by Redis Sentinel to achieve automatic failover. Haproxy configuration example:*
>
> ```
> listen redis
>   server redis-01 127.0.0.1:6380 check port 6380 check inter 2s weight 1 inter 2s
> downinter 5s rise 10 fall 2 on-marked-down shutdown-sessions on-marked-up shutdown-
> backup-sessions
>   server redis-02 127.0.0.1:6381 check port 6381 check inter 2s weight 1 inter 2s
> downinter 5s rise 10 fall 2 backup
> ```

```
bind *:6379
mode tcp
option tcpka
option tcplog
option tcp-check
tcp-check send PING\r\n
tcp-check expect string +PONG
tcp-check send info\ replication\r\n
tcp-check expect string role:master
tcp-check send QUIT\r\n
tcp-check expect string +OK
balance roundrobin
```

Next, point Grafana Live to Haproxy address:port.