New Relic Prometheus for Redis Enterprise

Installation Instructions

Get a New Relic token and Configure Prometheus

1. From the left-hand menu, click 'Add Data'.
2. Enter 'Prometheus' in the search bar and then select 'Prometheus Remote Write Integration'.
3. Enter a name for the installation (e.g., prometheus-server-prod) and click 'generate'. This will produce the remote write configuration with a bearer token.

4. Open the attached prometheus.yml file and find the 'remote_write' configuration directive.

5. Under 'url', replace '<YOUR CLUSTER REFERENCE>' with the installation name you provided when generating your token.

6. Next, paste the value from the bearer_token field in place of the 'XXXX…' under 'remote_write' -> 'authorization' -> 'credentials'.

7. Finally, find the 'scrape_config' where 'job_name' is "redis-enterprise". Under 'static_configs', set the target IP address or hostname and port to that of your Redis installation.

Run a Prometheus Server via Docker

You're going to run a Prometheus Docker container that forwards Prometheus metrics from Redis Enterprise to New Relic.

1. Ensure that the VM running this container is VPC-peered to your cluster's VPC.

2. Create the initial peering request from your Redis Cloud cluster's UI. This will produce a command that you can execute to peer your VPCs. Run the peering command as follows:

```
gcloud compute networks peerings create <peering label> --project
<source project> --network <source network> --peer-network=<target
network> --peer-project=<target project>
```

You can test this peering by pinging an address for your Redis cluster from a VM on the peered network. Once that is successful, validate REST API access:

```
curl  -vk -u user:password  https://34.162.55.19:9443/v1/license
```

If this is successful, then proceed to create a Dockerfile and container image.

1. Create a folder for the project files (.e.g, redis-newrelic-config)

2. Create a file `redis-newrelic-config/Dockerfile` containing the next two lines:

```
FROM ubuntu/prometheus:latest
ADD prometheus.yml /etc/prometheus/prometheus.yml
```

3. Build and tag the image

```
docker build -t <image name> .
```

4. Run the image from the container we built:

```
docker run \
    -d \
    --name prometheus-exporter \
    --network=host \
    --cap-add=SYS_PTRACE \
    --privileged \
    --pid=host \
    --cgroupns=host
    <image name>
```

Verify this after five minutes by opening New Relic and clicking on 'Query Your Data' in the left menu. In the 'Metric' search bar, enter 'bdb_up', which should then appear in the results. You can then click to launch a NRQL query, and this should return a value corresponding to the number of databases.

Install the dashboards

See the attached the dashboard files (newrelic-*.json). In order to install them, click on the 'Dashboards' link in the left menu, and then click 'Import Dashboard' in the top right. Then paste the contents of the json file.

Once you've installed the dashboards, open the cluster dashboard and select any dropdown values at the top. For example, you should see values in the 'cluster' dropdown and be able to select one, at which point the graph should have active data; if not, set the time range to last 10 minutes at the top right.