A CLOUD GURU

Lesson Description - Federation

Prometheus supports the ability to pull metric data from one Prometheus server to another. This allows you to have local Prometheus servers monitoring a small set of applications and services, while also passing that data to other Prometheus servers for aggregation and/or centralization. This process is known as federation. In this lesson, we will discuss how federation can be used with Prometheus, and we will demonstrate how to federate data between Prometheus servers.

Relevant Documentation

- Federation
- Scaling and Federating Prometheus

Lesson Reference

To federate data, you will need to build a new Prometheus server with the following settings:

- Distribution: `Ubuntu 18.04 Bionic Beaver LTS`
- Size: `Small`
- Tag: `Federal Prometheus Server`

Note: You may need to delete an existing server to make room for this new server. You can delete either the Grafana or Prometheus 2 server.

Install Prometheus on the Federal Prometheus Server

Log in to your new Federal Prometheus Server.

Create a user, group, and directories for Prometheus:

```
sudo useradd —M —r —s /bin/false prometheus
```

```
sudo mkdir /etc/prometheus /var/lib/prometheus
```

Download and extract the pre-compiled binaries:

```
wget https://github.com/prometheus/prometheus/releases/download/
v2.16.0/prometheus-2.16.0.linux-amd64.tar.gz
```

```
tar xzf prometheus-2.16.0.linux-amd64.tar.gz prometheus-2.16.0.linux-
amd64/
```

Move the files from the downloaded archive to the appropriate locations and set ownership:

```
sudo cp prometheus-2.16.0.linux-amd64/{prometheus,promtool} /usr/
local/bin/
```

```
sudo chown prometheus:prometheus /usr/local/bin/{prometheus,promtool}
```

```
sudo cp -r prometheus-2.16.0.linux-amd64/
{consoles,console_libraries} /etc/prometheus/
```

```
sudo cp prometheus-2.16.0.linux-amd64/prometheus.yml /etc/prometheus
/prometheus.yml
```

```
sudo chown -R prometheus:prometheus /etc/prometheus
```

```
sudo chown prometheus:prometheus /var/lib/prometheus
```

Briefly test your setup by running Prometheus in the foreground:

```
prometheus --config.file=/etc/prometheus/prometheus.yml
```

Create a `systemd` unit file for Prometheus:

```
sudo vi /etc/systemd/system/prometheus.service
```

Define the Prometheus service in the unit file:

```
[Unit]
Description=Prometheus Time Series Collection and Processing Server
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
    --config.file /etc/prometheus/prometheus.yml \
    --storage.tsdb.path /var/lib/prometheus/ \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

Add Configuration to Federate Data from Another Prometheus Server

Edit the Prometheus config on your Federal Prometheus Server:

```
sudo vi /etc/prometheus/prometheus.yml
```

Add the `/federate` endpoint on your first Prometheus server as a new scrape target. Make sure you use the private IP address of your first Prometheus server for the target:

```
scrape_configs:

  ...

  - job_name: 'federate'
    scrape_interval: 15s
    honor_labels: true
    metrics_path: '/federate'
    params:
      'match[]':
        - '{job!~"prometheus"}'
    static_configs:
```

```
    - targets:
      - '<PROMETHEUS_SERVER_1_PRIVATE_IP>:9090'
```

Start and enable Prometheus:

```
sudo systemctl enable prometheus
```

```
sudo systemctl start prometheus
```

Access your Federal Prometheus Server in a browser at `http://<FEDERAL_PROMETHEUS_SERVER_PUBLIC_IP>:9090`. Run a query to pull some data about your jobs:

```
up
```

You should see data about jobs run by your first Prometheus server.