

# Lesson Description - High Availability

---

A single Prometheus server could become a single point of failure if it goes down, causing you to be unable to access critical metric data when you need it. The simplicity of Prometheus' design makes it relatively easy to create new Prometheus servers for high availability. In this lesson, we will discuss what high availability looks like for Prometheus, and we will demonstrate it by building a second Prometheus server.

## Relevant Documentation

- [Can Prometheus be made highly available?](#)

## Lesson Reference

If you already built a second server for an additional Alertmanager instance, feel free to use that same server for this lesson.

Otherwise, you will need to create a second server with the following settings:

- Distribution: **Ubuntu 18.04 Bionic Beaver LTS**
- Size: **Small**
- Tag: **Prometheus 2**

## Install Prometheus on the Second Server

Log in to your second 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
```

## Copy All Prometheus Configurations from Server 1 to Server 2

Log in to your first Prometheus server.

Edit your Prometheus config to change any `localhost` references that might not work properly on the new server:

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

Change references to `localhost` *except* for the target of the `prometheus` job to the private IP address of your first Prometheus server.

Copy `prometheus.yml` to Prometheus Server 2:

```
scp /etc/prometheus/prometheus.yml
cloud_user@<PROMETHEUS_SERVER_2_PRIVATE_IP>:/home/cloud_user
```

Copy your rules files to Prometheus Server 2:

```
scp /etc/prometheus/rules/*
cloud_user@<PROMETHEUS_SERVER_2_PRIVATE_IP>:/home/cloud_user
```

Log in to Prometheus Server 2 and move `prometheus.yml` to the appropriate location:

```
sudo mv ~/prometheus.yml /etc/prometheus/prometheus.yml
```

Create the rules directory, and then move the rules file to the appropriate location.

```
sudo mkdir -p /etc/prometheus/rules
```

```
sudo mv ~/*.yml /etc/prometheus/rules
```

### Start the Second Prometheus Instance

On Prometheus Server 2, start and enable Prometheus:

```
sudo systemctl enable prometheus
```

```
sudo systemctl start prometheus
```

Access Prometheus Server 2 in a browser at [http://<PROMETHEUS\\_SERVER\\_2\\_PUBLIC\\_IP>:9090](http://<PROMETHEUS_SERVER_2_PUBLIC_IP>:9090). Run a query to verify it is working:

```
up
```

You should see **up** data for all jobs that were previously set up on Prometheus Server 1.

You can also click **Alerts** to verify the alerts from Prometheus Server 1 appear.