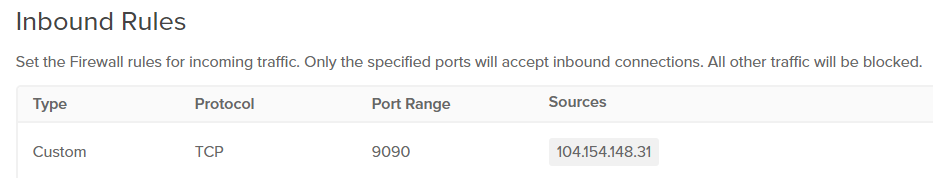
This guide is intended for Radix validator node operators running their nodes with the Docker configuration who would like to use Grafana Cloud instead of the local Grafana container that is co-located with the Radix Core, Nginx and Prometheus containers. Using Grafana Cloud instead of the local co-located Grafana container provided by the Radix team will enable alerts for complete node outages, such as server crashes (which would also bring down the co-located Grafana container running on the same server), therefore this is a recommended configuration that provides more reliable monitoring of the Radix validator node.

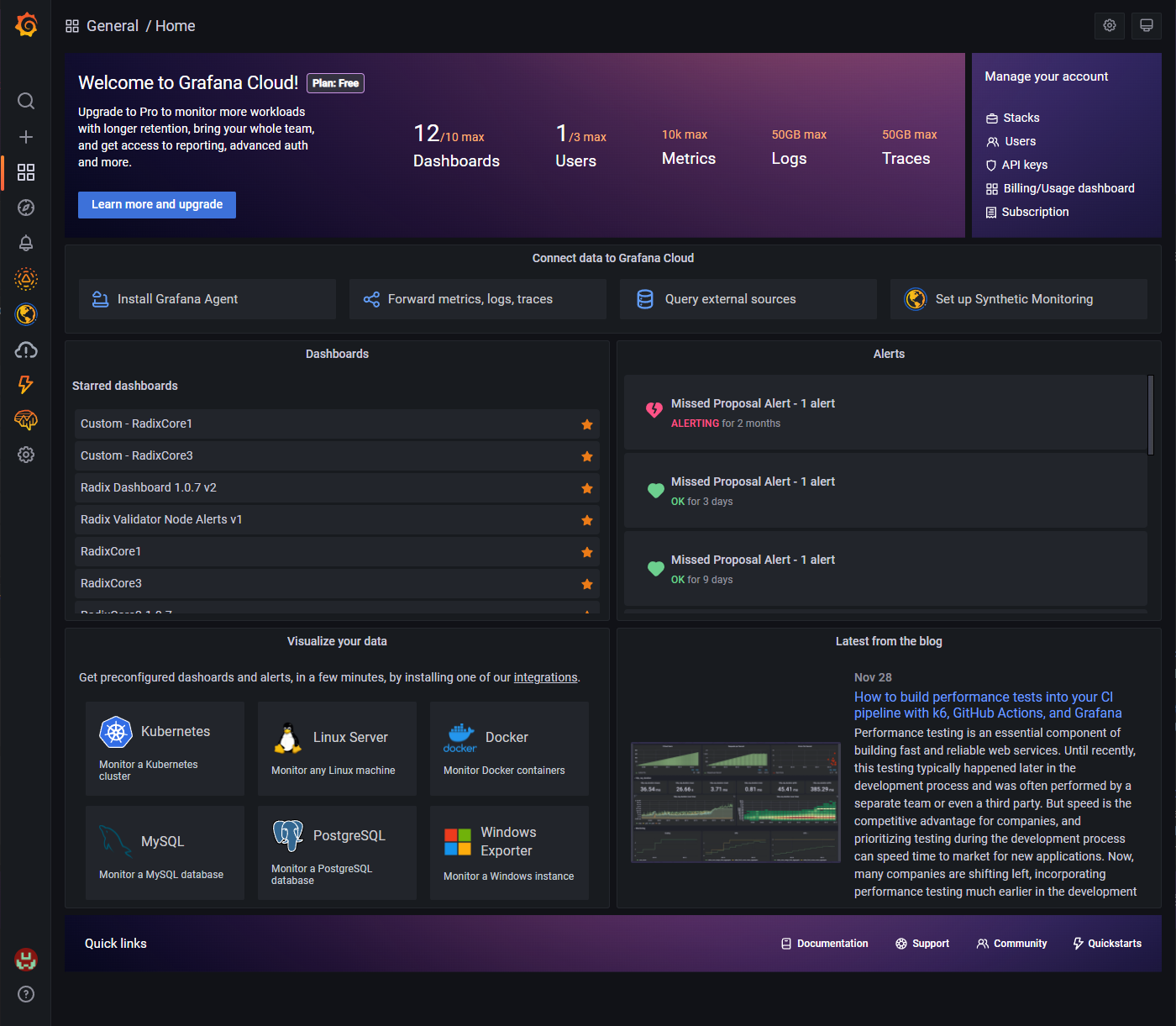
Steps to configure Grafana Cloud:

1. Start by running radixnode monitoring setup and answering “yes” in the prompt in the Linux server, if you haven’t already
2. Add the firewall rules needed to enable connectivity from Grafana Cloud to your Radix node.
   1. Navigate to <https://grafana.com/api/hosted-grafana/source-ips.txt> to obtain a list of the Grafana Cloud source IP addresses.
   2. Configure an inbound firewall rule for each IP address, using port 9090. The detailed steps depend on the environment; below is an example using Digital Ocean:

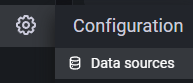


**Warning:** given the large number of unique IP addresses (111), and the fact that it’s not possible to simply configure a CIDR range to cover all 111 IP addresses, it will be tempting to simply open port 9090 to all incoming requests. However, this will create a security risk by exposing your server to adversaries who may be able to exploit security vulnerabilities, therefore we highlight recommend for you create individual firewall rules similar to the one depicted in the figure above for each of the IP addresses listed in <https://grafana.com/api/hosted-grafana/source-ips.txt>.

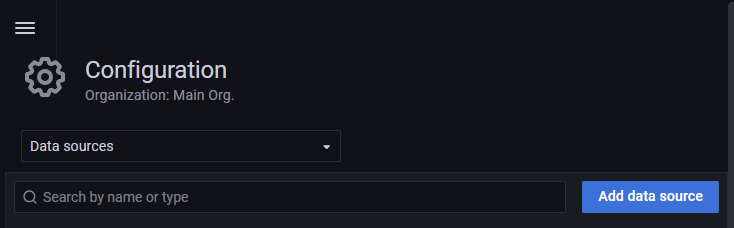
1. Create and test data source in Grafana Cloud.
   1. Log into your Grafana Cloud account and navigate to your home page (e.g., [https://*your\_org\_name*.grafana.net](https://your_org_name.grafana.net)). Your home page should look similar to the one below:



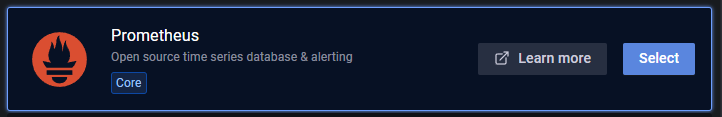
* 1. Click on Configuration > Data Sources



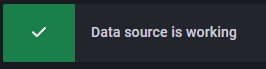
* 1. Click on the “Add data source” button



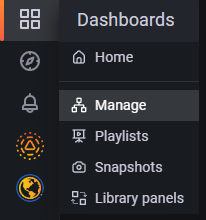
* 1. Click the Select button for Prometheus



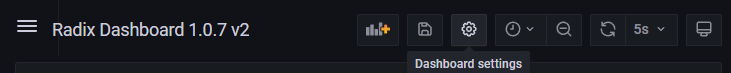
* 1. Change the following values for your data source:
     1. Change the name using your custom data source naming convention
     2. In the URL field, enter a URL in the following format: <http://external_IP_address_of_your_node:9090/prometheus>
     3. Change HTTP Method to GET instead of POST
     4. Click the “Save & test” button. If the configuration was successful you’ll get a message saying “Data source is working”.



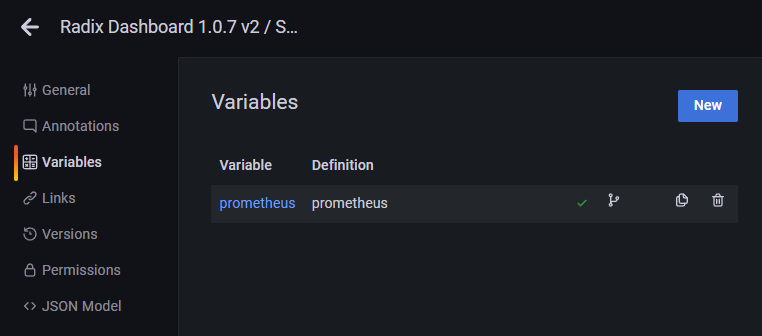
1. Importing and Configuring Grafana Dashboards
   1. Download the Grafana dashboard JSON files from our repository at <https://github.com/RadicalStaking/Grafana>.
   2. Navigate to Dashboards > Manage



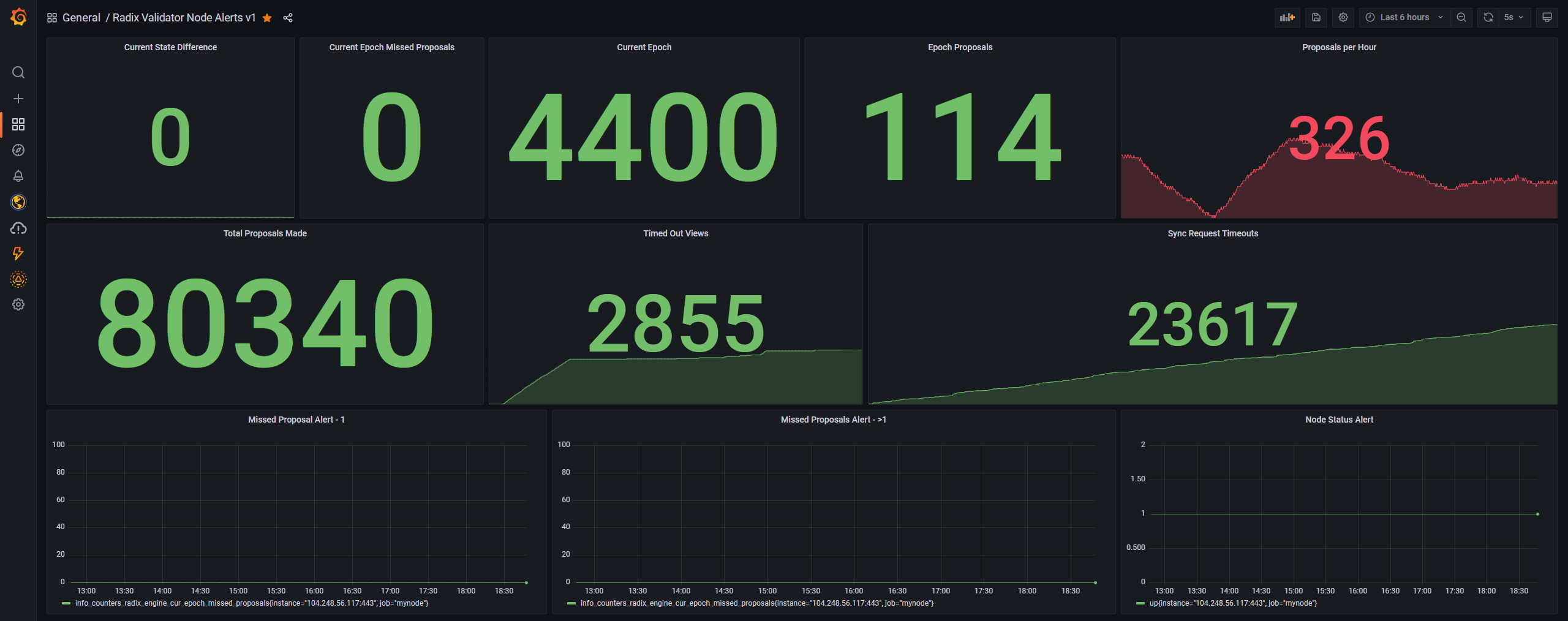
* 1. Click the Import button, and select radix\_dashboard\_1.0.7\_v2.json. Repeat this step for radix\_validator\_node\_alerts\_v1.json
  2. After the dashboards are imported, click on the dashboard link for Radix Dashboard 1.0.7 v2 in the General folder in the Dashboards page
  3. When the dashboard comes up, click on the “Dashboard settings” button

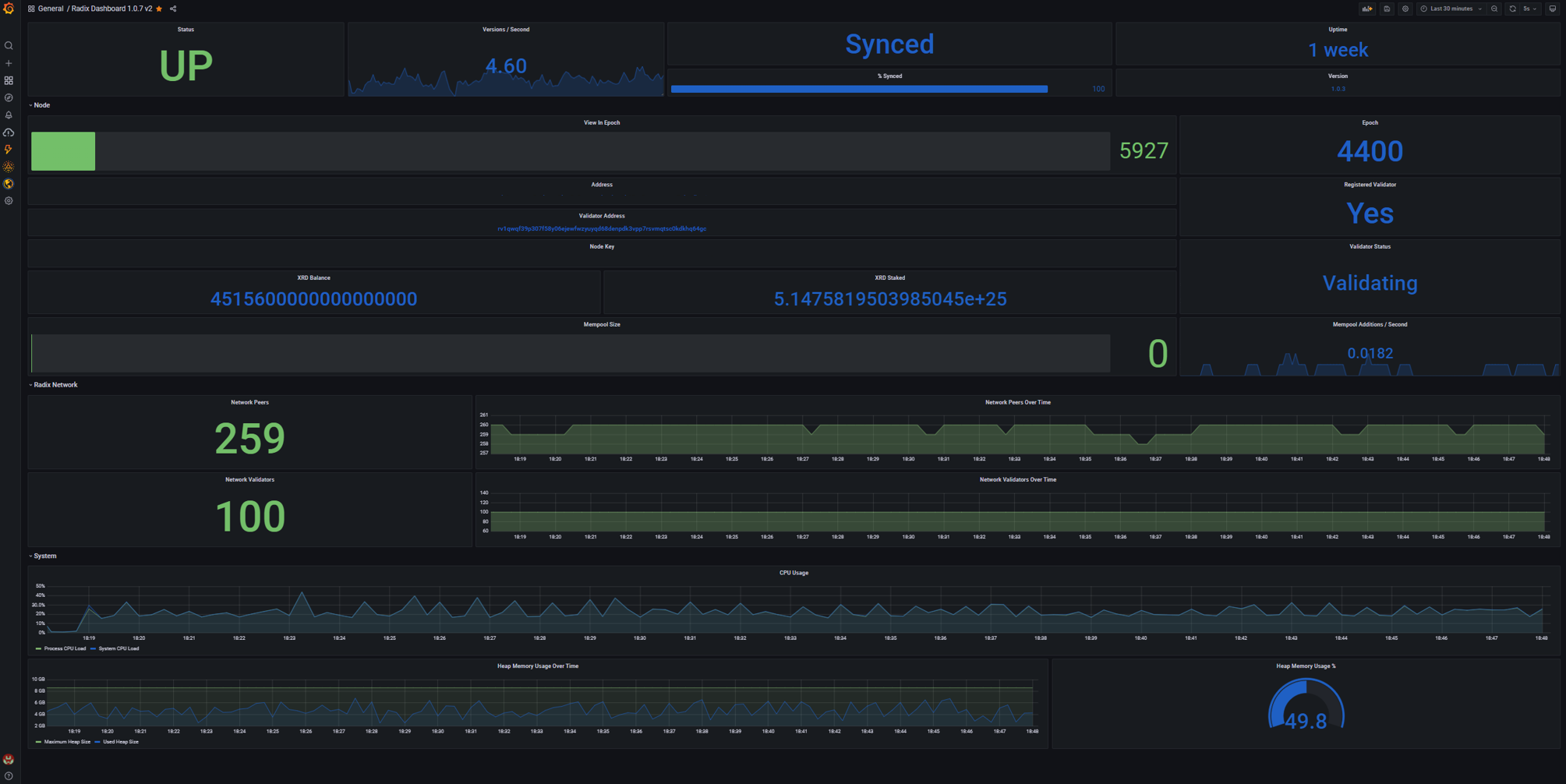


* 1. Click on Variables and click on the Prometheus variable

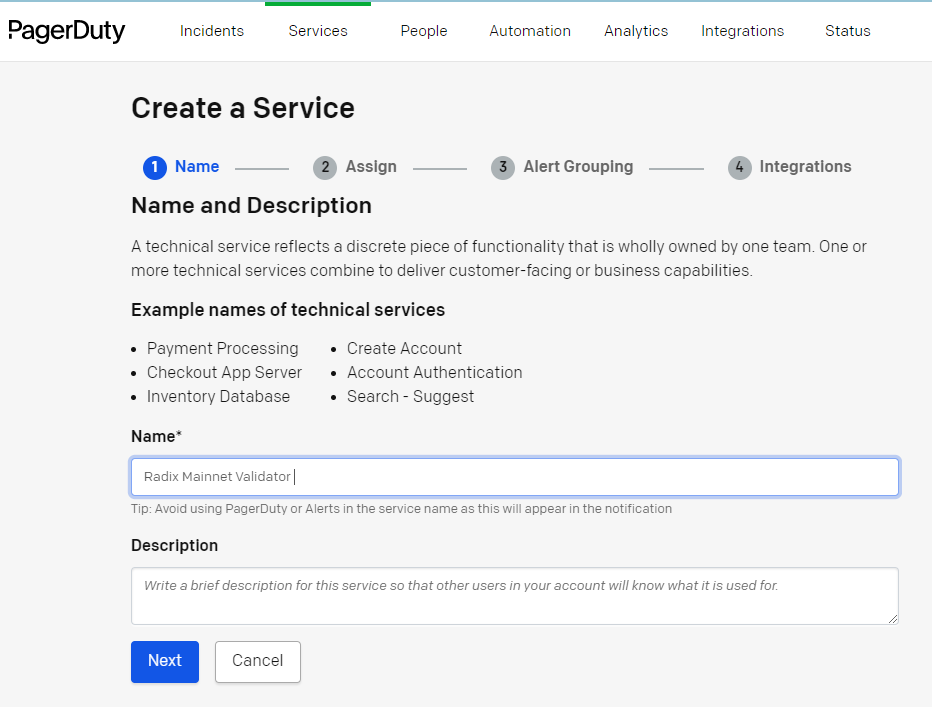


* 1. Change the value in the “Instance name filter” to the data source name you set in step 3.e.1.
  2. Click on Update, and Click on “Save dashboard”
  3. Repeat steps d. through h. for the Radix Validator Node Alerts v1 dashboard.
  4. After a few seconds the dashboards will populate the data, pulling it from the Prometheus container running in your node. Your dashboards should look similar to the following:

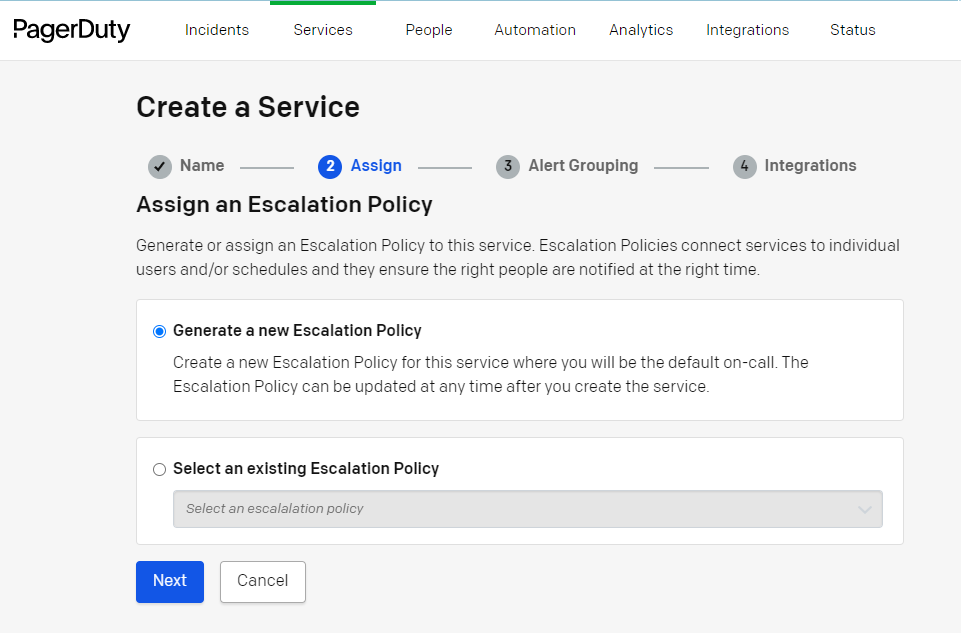




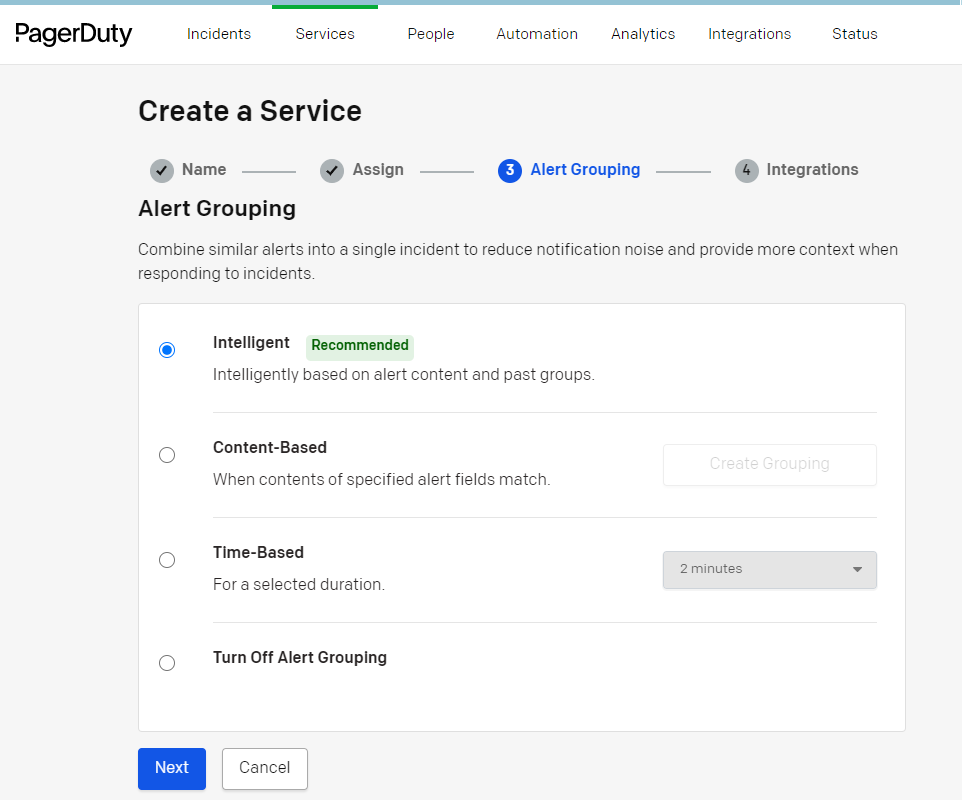
1. Configure alerts for missed proposals and node uptime in the Radix Validator Node Alerts v1 dashboard. In this example we configure alerts through PagerDuty.
   1. Create a free tier account at <https://www.pagerduty.com>
   2. Log into your account and create a new service in the Services tab



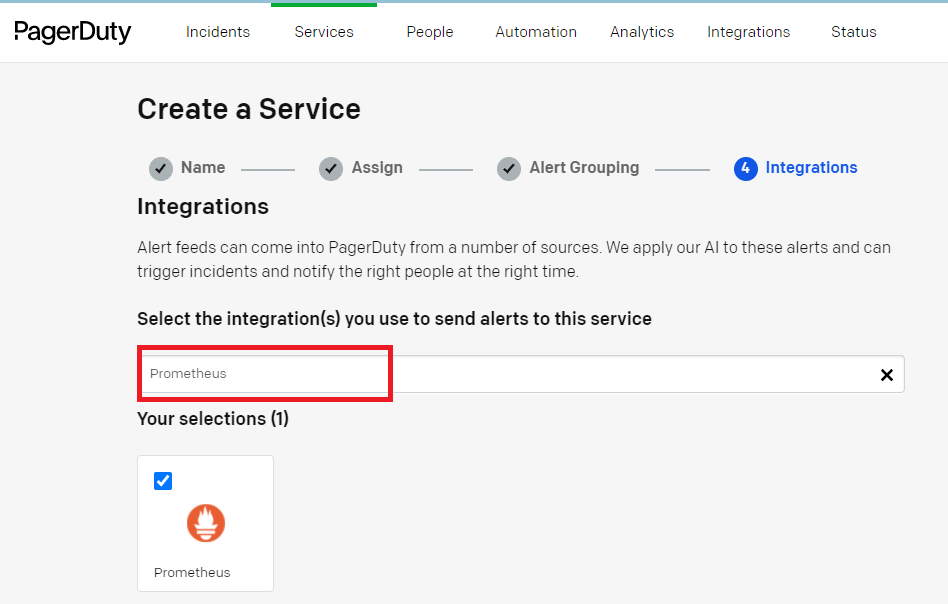
* 1. Set the default Escalation Policy



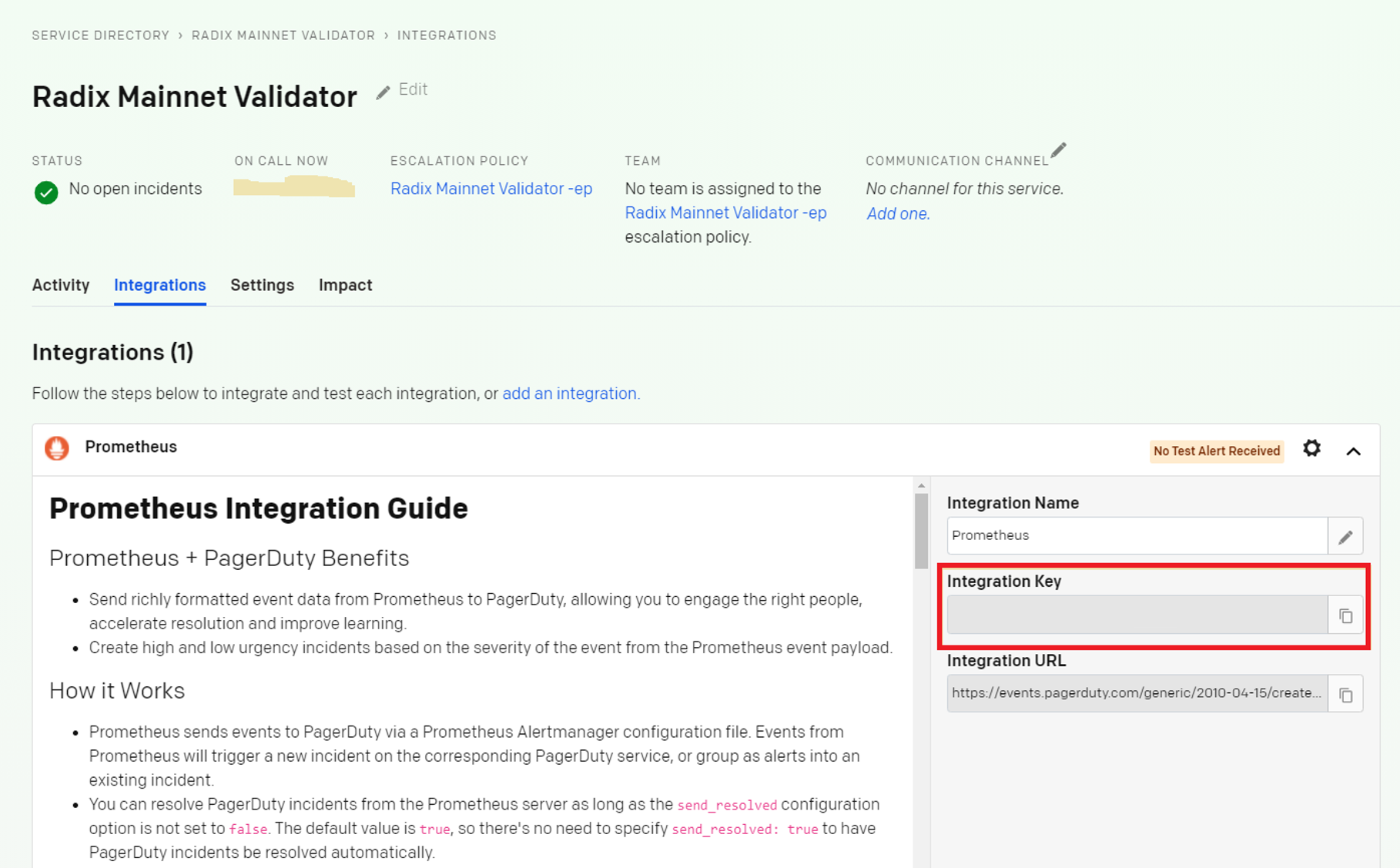
* 1. Set the default Intelligent alert grouping



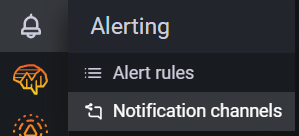
* 1. Select Prometheus in the Integrations step



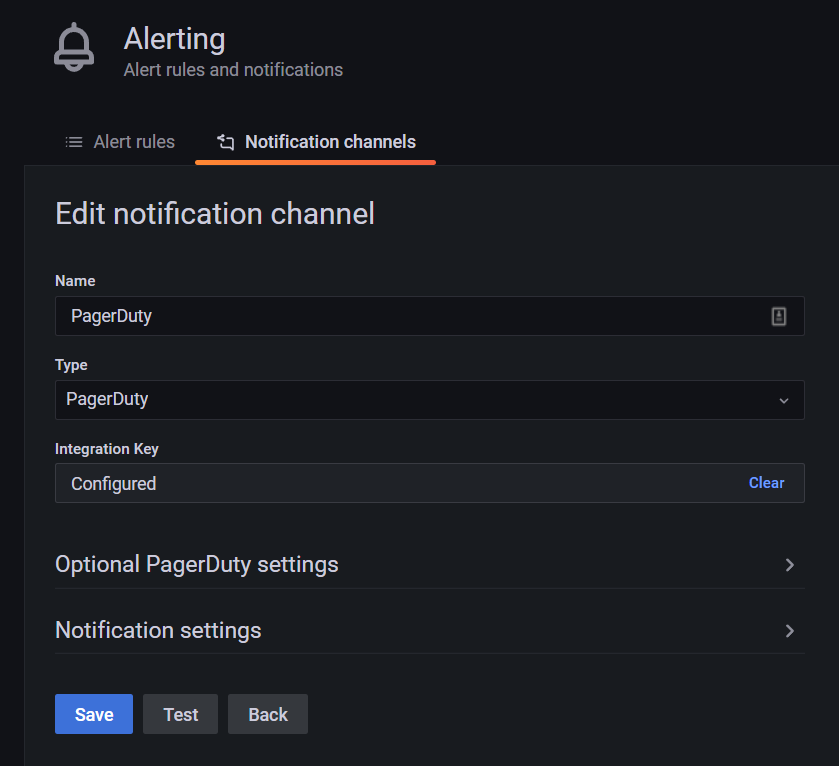
* 1. Copy the Integration Key and save it somewhere safe.



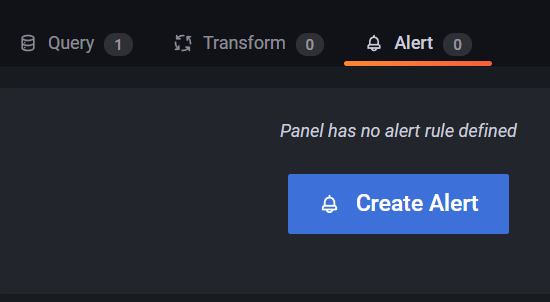
* 1. Click on your profile avatar in the top right hand corner and select My Profile.
  2. In the Contact Information tab, add your mobile number and/or an email address to receive SMS, Phone call and email alerts as desired.
  3. In Grafana Cloud, navigate to Alerting > Notification channels



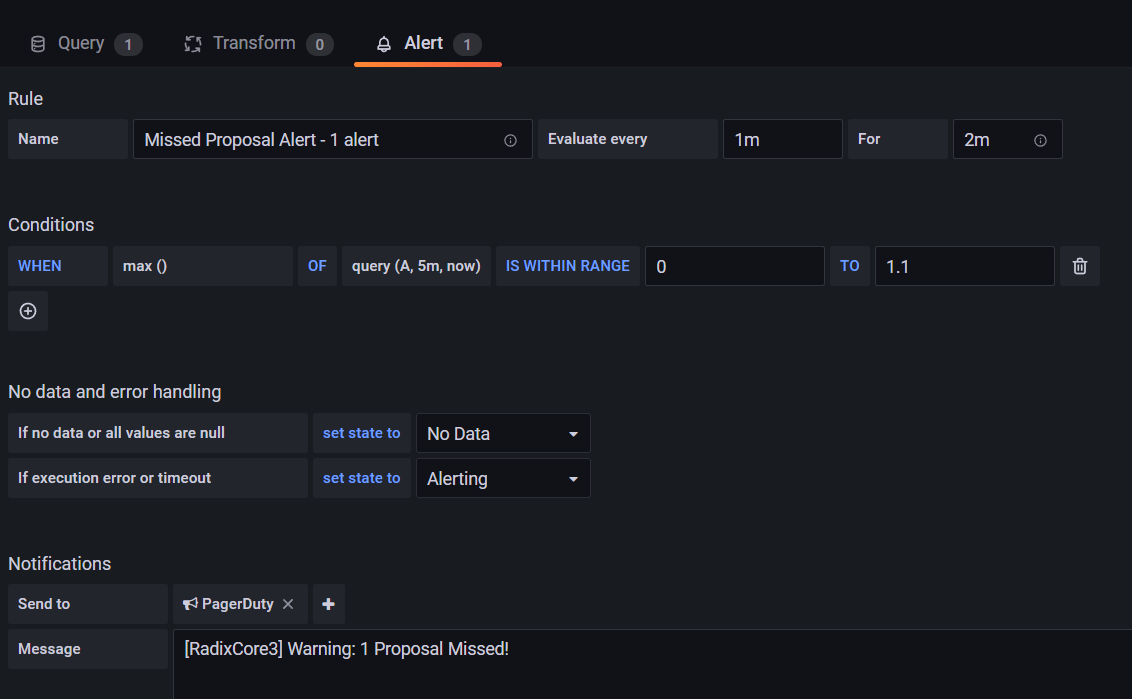
* 1. Create a Notification Channel for Pager Duty by adding your Integration Key, then save and test.



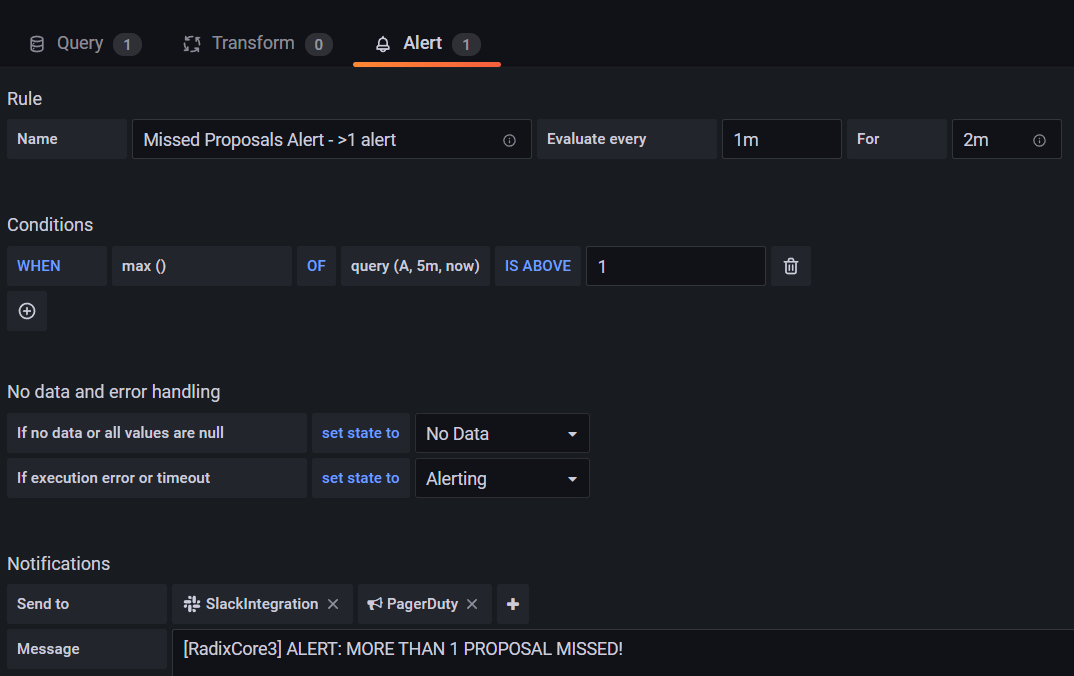
* 1. In the Radix Validator Node Alerts v1 dashboard, click on Missed Proposals Alert – 1 and select Edit
  2. Click on the Alert tab and click on Create Alert



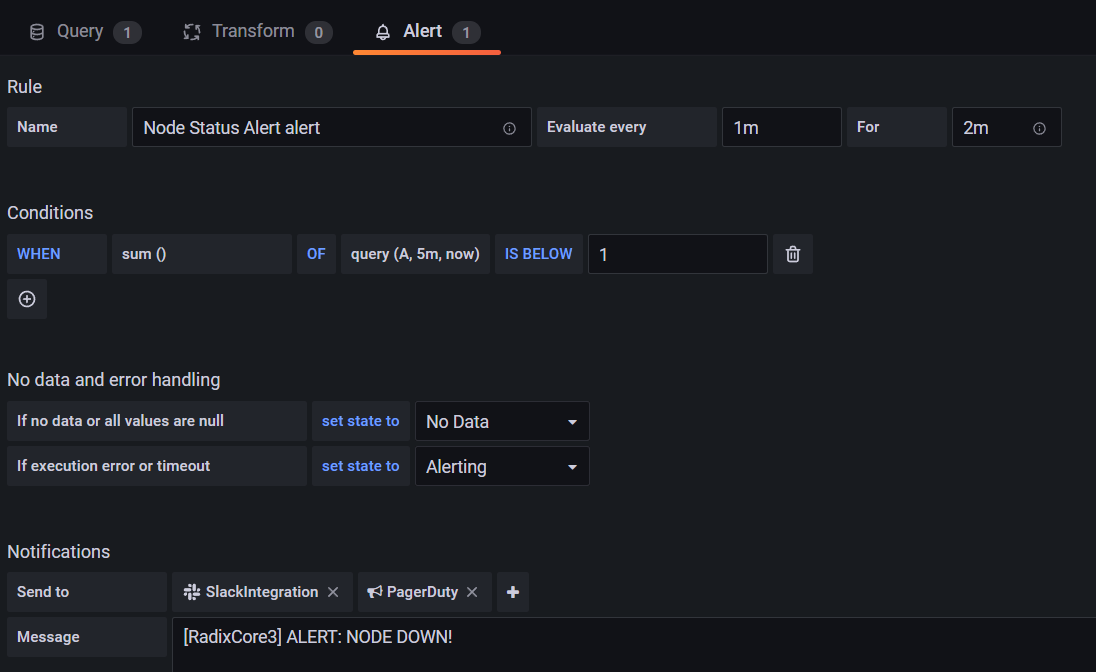
* 1. Configure the alert as shown in the screenshot below, replacing RadixCore3 with your server name:



* 1. Repeat steps k. through m. for Missed Proposals Alert - >1, configuring the alert as shown below:



* 1. Repeat steps k. through m. for Node Status Alert, configuring the alert as shown below:



* 1. Download the PagerDuty app to your iPhone or Android, configure notifications for the app, and test the alerts to confirm that they are working.
  2. (optional) bring down the Grafana container still running locally in your node by running the command “docker ps”, and then running “docker stop *container\_id*” using the Container ID number shown in the first column of the output of the docker ps command.