Monitoring Your Algorand Node

Introduction

Nodes in the Algorand network are comprised of two distinct types, relays and non-relays. For a complete description of the various configurations and their use-cases, see the following link: https://developer.algorand.org/docs/run-a-node/setup/types/. Whether a node is participating in consensus, supporting a dApp, or acting as a MainNet relay, it is important to keep it online and in service. While there are a variety of tools – open-source or paid – that could be combined effectively, we focus on widely-used open-source software and services that can achieve reliable results in the Linux operating environment. The tools used in this guide consume minimal resources, and will work on almost any hardware.

Our monitoring solution combines Prometheus, Node Exporter, Push Gateway, and Grafana. Prometheus is a time-series database that efficiently stores metrics from a variety of sources. Node Exporter collects hardware and kernel metrics from the host and exposes them to Prometheus. Push Gateway provides a REST endpoint that allows service-level metrics to be cached and published to Prometheus. Grafana is an open-source visualization platform with a suite of related tools that can use Prometheus as a data source.

- Prometheus to store time-series data
- Node Exporter to monitor host metrics
- Push Gateway to record service metrics
- Grafana and a pre-built dashboard to visualize the data

This combination provides the core functionality of an "observability stack" that can be expanded to include logs, traces, alerts, and other capabilities. The core tools configured in this guide provide near-real-time performance monitoring for your node from any web client. These tools can assist with troubleshooting through data discovery – for example by allowing the observation of historical trends prior to an event. If you configure alerting, you can also be proactively informed when metrics exceed critical thresholds, potentially avoiding failures before they occur, or significantly reducing your response time after a failure.

In order to proceed, you must have:

- A configured Algorand node running Ubuntu
- Shell access to the host machine with administrative privileges

The steps in this guide should work with some modification for other Linux distributions and MacOS. It is possible to host an Algorand node on Windows, and these tools will also run in that environment, if Windows Exporter is substituted for Node Exporter. However, the installation

steps and dashboard would require significant modification. An alternative would be to use InfluxDB, Telegraf, and Grafana.

The purpose of this guide is to present a simplified and functional method to monitor your node or nodes. There are of course other options that could be explored. For example, InfluxDB could be used instead of Prometheus, and Netdata to collect host metrics instead of Node Exporter. You could also set up alerts that trigger when metrics cross defined thresholds, using Prometheus Alert Manager and Grafana Alerting. If your node doesn't have much locally-available storage or compute, or if you have several nodes to monitor at once, you could also opt to use Grafana Cloud for both storage and visualization. Feel free to experiment with more advanced configurations to achieve your specific goals.

None of the software used in this guide is known to create specific security risks, but always exercise caution and follow best practices to properly secure your host. Make sure to follow each step in this guide in order, and verify each step, as directed. We recommend you strictly limit access to the ports exposed by these services, and make sure to use secure, unique and periodically-rotating passwords for any privileged accounts.

Installer Script

We have created a shell script to simplify the process of installing the monitoring tools. To begin, first connect to your host machine, and then run the following commands:

```
$ dir=algorand-monitoring; sudo mkdir -pm744 /etc/${dir}; cd /etc/${dir};
$ sudo wget -nd https://raw.githubusercontent.com/node-ops-llc/algorand-monitoring/main/monitoring-installer.sh;
$ sudo chmod 744 *.sh;
```

Now that the script is downloaded, it can be run to install the monitoring components. Run the script once per component. To see the help menu, run the following command:

```
$ ./monitoring-installer.sh --help
```

The following text should display:

```
Usage: ./monitoring-installer.sh [--1|--2|--3|--4|--5|--help]

Options:
  --help  Show the help menu
  --1     Install Prometheus
  --2     Install Node Exporter
  --3     Install Push Gateway
  --4     Install Grafana
  --5     Install Algorand dashboard
```

Install Prometheus

To install Prometheus using the installer script, run the following command:

```
$ ./monitoring-installer.sh --1
```

The script will perform the steps necessary to install Prometheus on the host machine. Verify the service is running by following the instructions displayed at the end of the installation process. First, run the command:

```
$ sudo systemctl status prometheus
```

The service status should display:

```
prometheus.service - Prometheus
    Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
    Active: active (running) since Mon 2022-12-19 21:38:44 UTC; 1 day 23h ago
      Docs: https://prometheus.io/docs/introduction/overview/
  Main PID: 3759528 (prometheus)
  ...
```

Next, access the query endpoint at http://<your-host-ip>:9090/prometheus on a web browser from any external client, or by using curl or lynx on your host. The Prometheus query interface should display. Be sure to restrict access to the service port using Uncomplicated Firewall (ufw), or iptables.

Install Node Exporter

To install Node Exporter, run the following command:

```
$ ./monitoring-installer.sh --2
```

The script will perform the steps necessary to install Node Exporter on the host machine, including configuring Prometheus. Verify the service is running by following the instructions displayed at the end of the installation process. You should be able to access the metrics endpoint at http://<your-host-ip>:9101/metrics. You should also be able to access Prometheus at http://<your-host-ip>:9090/prometheus. Use the Metrics Explorer (the globe icon next to the query button) to view the available metrics, and run some ad-hoc queries to understand how Prometheus and Node Exporter work together. Our instance of Node Exporter includes the textfile collector, providing a simple and effective method for recording metrics using text files that contain properly-formatted data. Take note that the Algorand package installs an instance of Node Exporter in /usr/bin. We collect data from its endpoint at http://<your-host-ip>:9100/metrics, and also augment the available collectors by installing a separate instance of Node Exporter in /usr/local/bin, which is exposed on port 9101. If Algorand were to remove its instance of Node Exporter, then the instance we installed can be reconfigured to include any metrics needed for the dashboard.

Install Push Gateway

The Push Gateway is an extension that allows service-level metrics to be reported from sources that cannot be scraped as a normal Prometheus data source. These metrics can be derived from short-lived jobs, services, or commands, and are reported to a REST-like endpoint.

To install Push Gateway, run the following command:

```
$ ./monitoring-installer.sh --3
```

The script will perform the steps necessary to install Push Gateway on the host machine, including configuring Prometheus. Verify the service is running by following the instructions displayed at the end of the installation process.

Install Grafana

To install Grafana, run the following command:

```
$ ./monitoring-installer.sh --4
```

The script will perform the steps necessary to install Grafana on the host machine, including configuring the Prometheus data source. Verify the service is running by following the instructions displayed at the end of the installation process. You should be able to access Grafana at http://<your-host-ip>:3000. Log in with the username "admin" and the password "admin". You will be prompted to set up a new password. Make sure to use a password that is complex, unique, and secure.

To verify your Prometheus sources are up and running, access http://<your-host-ip>:9090/prometheus/targets. You should see your sources enabled, with the state "UP".

Install Dashboard

To install the Algorand Dashboard, run the following command:

```
$ ./monitoring-installer.sh --5
```

The script will perform the steps necessary to install the Algorand Dashboard in Grafana. Login to Grafana at http://<your-host-ip>:3000 with your username and password, and verify that the dashboard displays in the dashboards list. Select the dashboard and verify the elements populate with data. Bookmark the dashboard for quicker access in the future.

<Dashboard Screen Shot TKTK>

Conclusion

This guide and the associated resources provide a simple and effective method to monitor your Algorand node – you now have a complete monitoring dashboard for a single node that can be accessed from any desktop or mobile web client.

To gain a better understanding of how each service is installed and configured, step through the commands in the installation script. If you have any feedback on this guide, or issues with the script or instructions, reach out on the Node Runners channel on Algorand Discord. If you wish to experiment with more advanced configurations, feel free to discuss and share your results.

Happy node running!