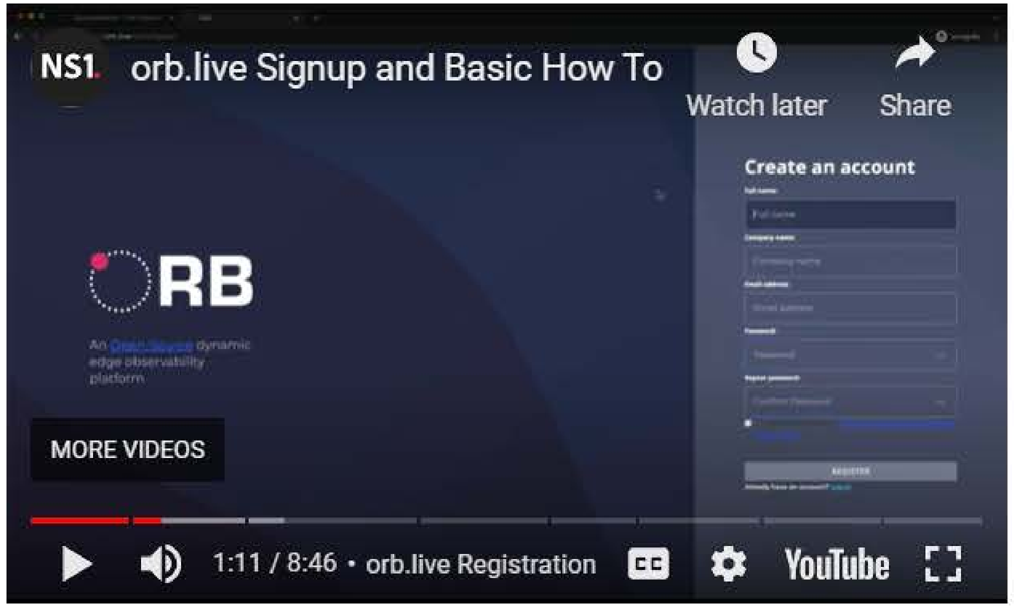
Try Orb now for free in under 10 minutes at orb.live community SaaS site (no CC required). Follow instructions here.

ORB

Open source, dynamic edge observability.

Get Started with Orb


NS1
orb.live Signup and Basic How To
Watch later
Share
Create an account
RB
MORE VIDEOS
1:11 / 8:46 • orb.live Registration
YouTube

Back to top

Why Orb?

Distributed deep network observability

Orb manages a fleet of **agents** deployed across distributed, hybrid infrastructure: *containers*, *VMs*, *servers*, *routers*, and *switches*. Agents tap into traffic streams and extract real-time insights, resulting in lightweight, actionable metrics. The result: faster time-to-action at a lower cost.

Streaming analysis at the edge

Based on the pktvisor observability agent, **Orb's goal is to push analysis to the edge**, where high-resolution data can be analysed in real time without the need to send raw data to a central location for batch processing. Current analysis focuses on L2-L3 Network, DNS, and DHCP with more analyzers in the works.

Real-time agent orchestration

Orb uses Internet of Things (IoT) principles to allow the **observability agents** to connect out to the Orb central **control plane**, avoiding firewall problems. Once connected, agents are controlled in real time from the Orb Portal or REST API, orchestrating observability policies designed to precisely extract the desired insights. Agents are grouped and addressed based on tags.

About

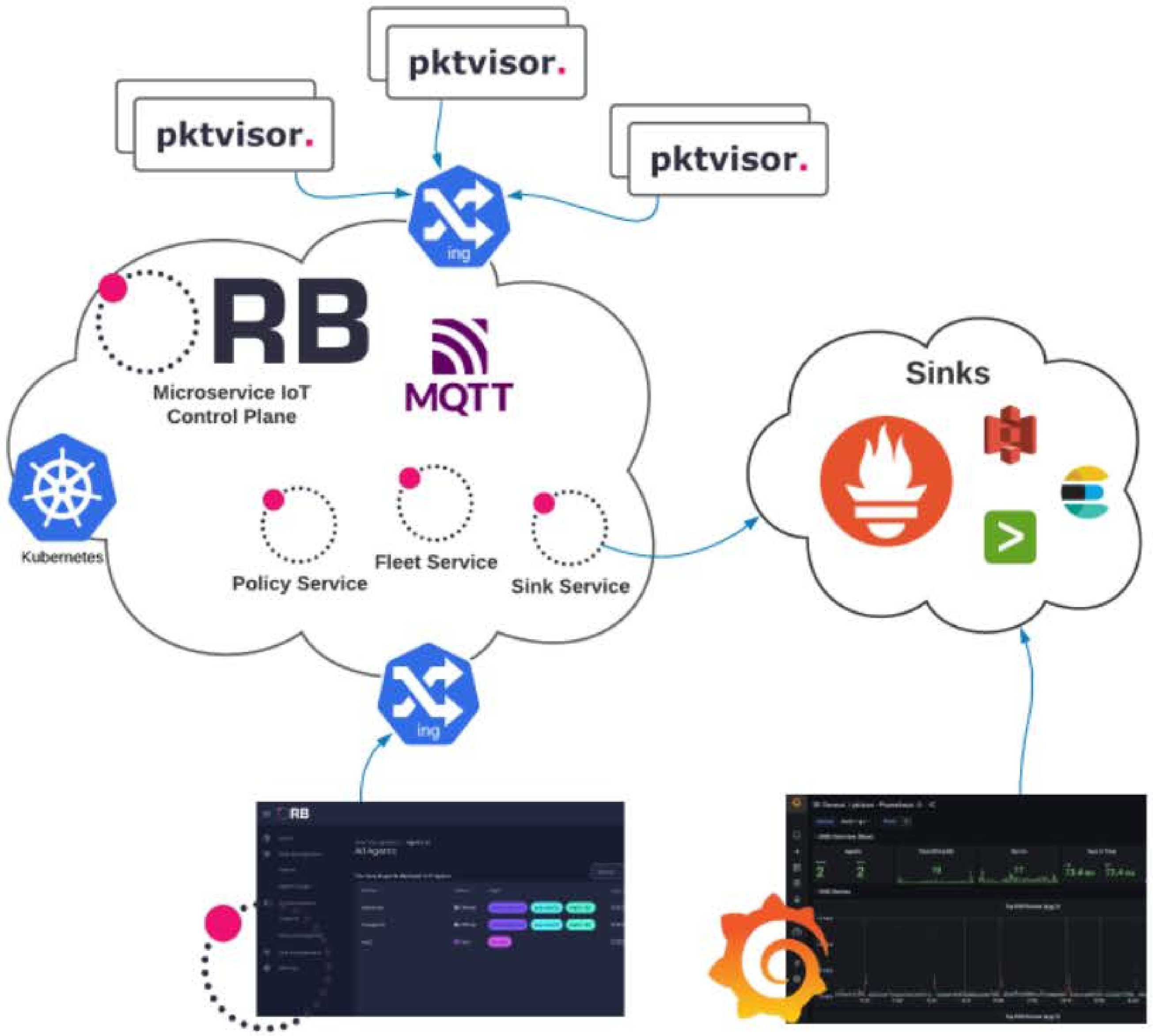


Table of contents

The story
The components
The agents
The control plane
The features
Orb + pktvisor
Core Concepts
Agent
Agent group
Fleet
Policies
Dataset
Sinks

Orb + pktvisor

Via **Orb's user interface**, you decide *what data* to extract from *which agents*. The resource-efficient, side-car style **pktvisor observability agent** performs edge analysis on network data streams.

This combination allows you to:

- Adjust analysis and collection parameters dynamically across the entire fleet via a powerful control plane
- Perform centralized fleet management, allowing you to configure heartbeats, tagging, and grouping for each of the pktvisor agents
- Orchestrate dataset policies that specify the type of data to extract from each agent

In terms of metrics, pktvisor can capture DNS, DHCP, and L2/L3 network data via packet capture, `dnstap`, `sflow`, among other input methods.

For a complete list of metrics currently collected by pktvisor, look here.

To view a Grafana dashboard for visualizing pktvisor Prometheus metrics, look here.

Table of contents

The story
The components
  The agents
  The control plane
The features
Orb + pktvisor
Core Concepts
  Agent
  Agent group
  Fleet
  Policies
  Dataset
  Sinks

Installation

Table of contents
orb.live
Self-host
Orb with Docker Compose
Requirements
Instructions
Orb Helm Chart
Requirements
Configuration

Orb consists of two major components:

1. The **Control Plane**—comprised of microservices, communication systems, databases, etc.—deploys to a central location (usually a cloud environment on Kubernetes).
2. The **Orb Agent**—a lightweight observability agent—deploys to all the infrastructure you wish to monitor.

Info

The instructions below are for installing the **Control Plane**. If you just need to install the **Orb Agent** (`orb-agent`), see these instructions instead.

The **Control Plane** can be self-hosted, or you can use our free Orb SaaS service. Self-hosting gives you full privacy and control but is more complex. On the other hand, our SaaS gets you up and running quickly since you only need to create a free account on orb.live and then install the **Orb Agent** to your infrastructure.

orb.live

The Orb SaaS platform (**orb.live**) is now in active development. This free-forever service allows you to enjoy the benefits of the Orb platform without having to run your own control plane.

If you need to install the **Orb Agent** to be used with orb.live, see these instructions.

Danger

The Orb SaaS service is still under active development and is not yet production ready. Please use it for non-production testing purposes only. The service may become unavailable, and your data may be reset without notice.

Self-host

There are two main deployment methods for those wanting to self-host:

- **Docker Compose** - This option is useful for developer or testing installations, allowing you to run both the Orb Control Plane and the Orb Agent on a single machine.
- **Helm Chart** - This option is intended for production deployments, requiring access to a Kubernetes cluster.

Follow the instructions below after choosing a self-host option.

Tip

If you're just interested in trying out Orb quickly to see what it's all about, use the Docker Compose method.

Table of contents

orb.live
Self-host
Orb with Docker Compose
Requirements
Instructions
Orb Helm Chart
Requirements
Configuration

Orb Helm Chart

Helm is a package manager for Kubernetes. A Helm Chart is a package that allows you to customize your deployment on Kubernetes.

To configure and deploy an Orb Helm Chart, follow the instructions below.

Requirements

- Helm v3

Configuration

This guide assumes installation into namespace `orb`. It requires a HOSTNAME over which you have DNS control. It uses Let's Encrypt for TLS certification management.

- cd to working directory `charts/orb`
- Add helm repos for dependencies.

```
helm repo add jaegertracing https://jaegertracing.github.io/helm-charts
helm repo add bitnami https://charts.bitnami.com/bitnami
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo add jetstack https://charts.jetstack.io
helm repo update
helm dependency update
```

- Create `orb` namespace.

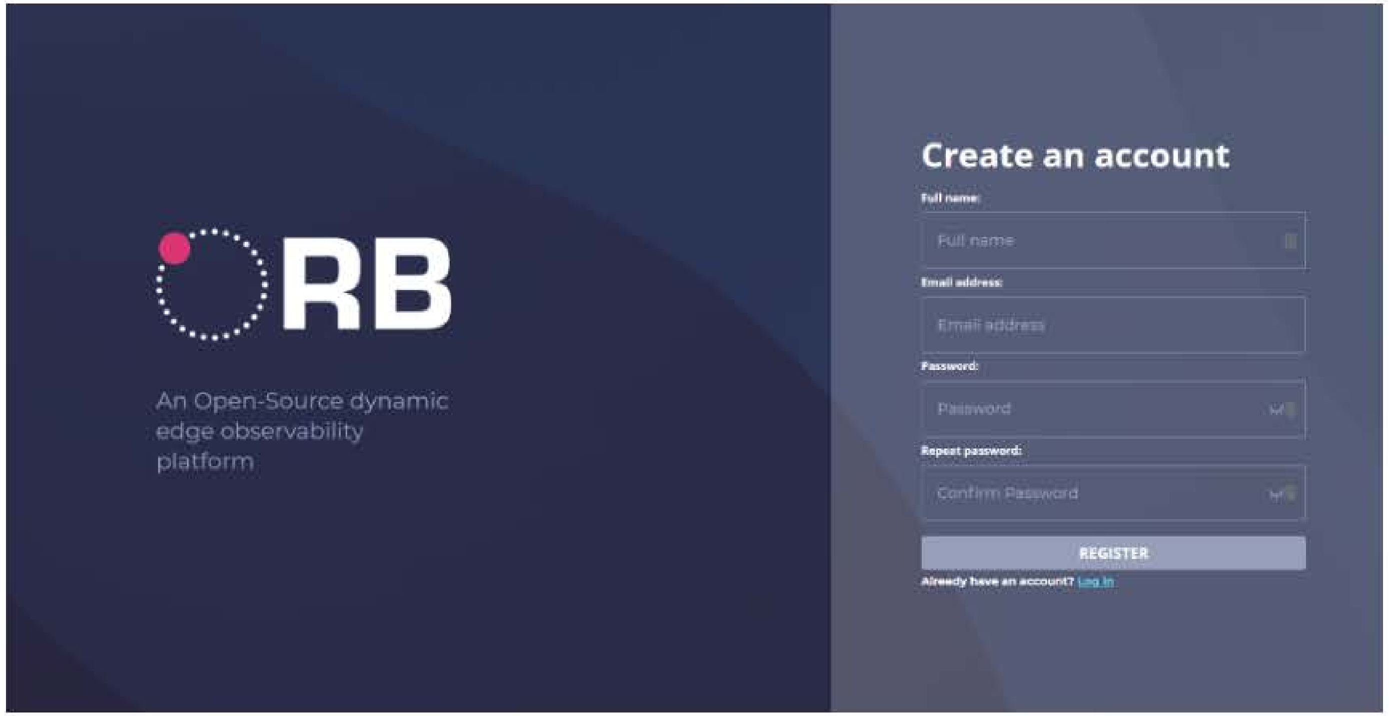
Table of contents

orb.live
Self-host
Orb with Docker Compose
Requirements
Instructions
Orb Helm Chart
Requirements
Configuration

Getting started

Follow the steps below after logging in to your Orb Portal to get an agent up and running.

Register a new account

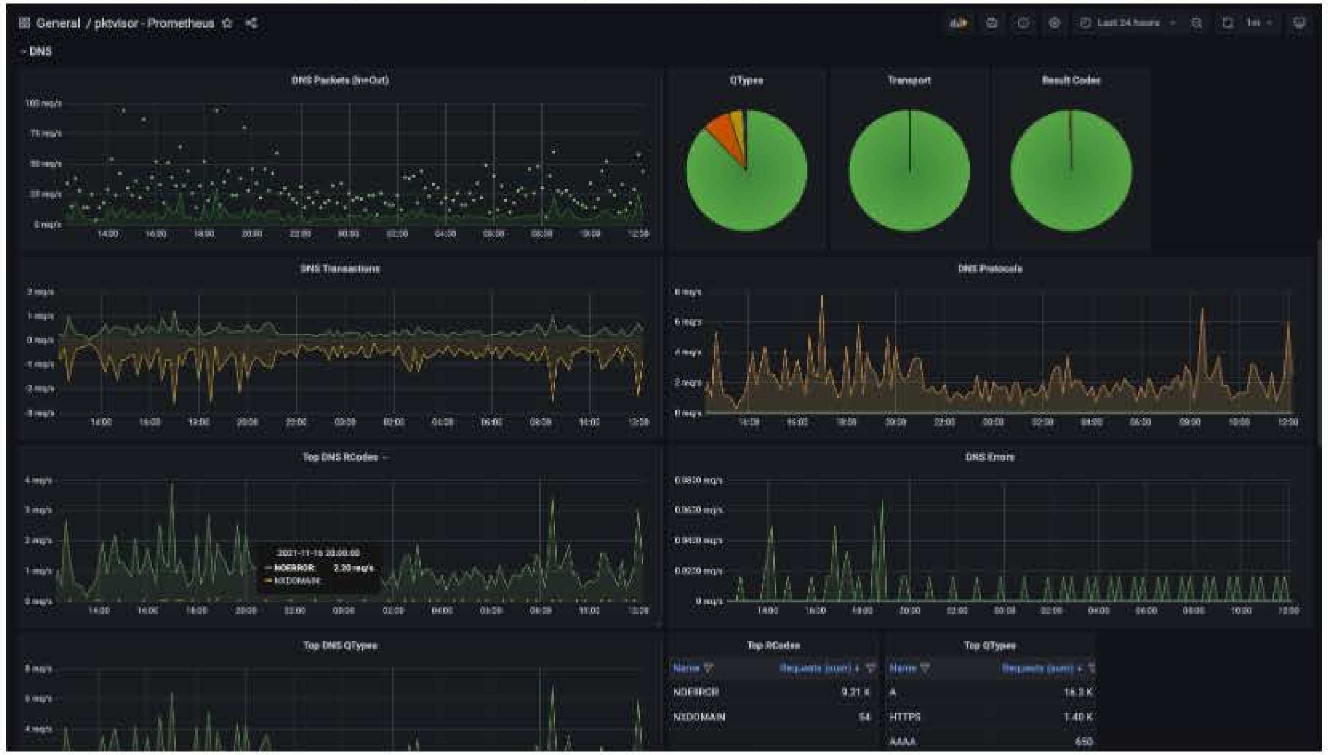


After registering, you should see the home page with a welcome message.

Getting started
Register a new account
Create an Agent
Create an Agent Group
Create a Sink
Create a Policy
Create a Dataset
Visualize and alert on your metrics
Running Orb Agent
Agent credentials
Sample provisioning commands
Configuration files
Advanced auto-provisioning setup
Working with API Docs

Visualize and alert on your metrics

1. Your agent should now be running the policy you created. After one minute of collection time, the metrics will be sent to your Prometheus sink.
2. You may use standard tools for visualizing and alerting on your Prometheus metrics. A popular option is Grafana.
3. A pre-made dashboard for visualizing Orb/pktvisor metrics is available for import here.



Name	Requests (sum)
NOERROR	9.21 K
NXDOMAIN	54

Name	Requests (sum)
A	16.3 K
HTTPS	1.40 K
AAAA	650

Getting started
Register a new account
Create an Agent
Create an Agent Group
Create a Sink
Create a Policy
Create a Dataset
Visualize and alert on your metrics
Running Orb Agent
Agent credentials
Sample provisioning commands
Configuration files
Advanced auto-provisioning setup
Working with API Docs

Running Orb Agent

An Orb agent needs to run on all the infrastructure (computers, servers, switches, VMs, k8s, etc.) to be monitored. It is a small, lightweight Docker process with an embedded pktvisor agent which connects into the Orb control plane to receive policies and send its metric output.

To run an agent, you will need:

1. Docker, to run the agent image (ns1labs/orb-agent:develop)
2. Agent Credentials, which are provided to you by the Orb UI or REST API after creating an agent
3. The Orb Control Plane host address (e.g. `localhost` or `orb.live`)
4. The network interface to monitor (e.g. `eth0`)

Tip

If you are unsure which network interface to monitor, you may list the available interfaces on your host. Note that to allow the agent access to these interfaces, you must run the container with `--net=host`

Linux

```
ip -stats -color -human addr
```

OSX

```
ifconfig
```

Getting started
- Register a new account
- Create an Agent
- Create an Agent Group
- Create a Sink
- Create a Policy
- Create a Dataset
- Visualize and alert on your metrics

Running Orb Agent
- Agent credentials
- Sample provisioning commands
- Configuration files
- Advanced auto-provisioning setup

Working with API Docs

Sample provisioning commands

Example

Generic

Use this command as a template by substituting in the appropriate values:

```
docker run -d --net=host
-e ORB_CLOUD_ADDRESS=<HOST>
-e ORB_CLOUD_MQTT_ID=<AGENTID>
-e ORB_CLOUD_MQTT_CHANNEL_ID=<CHANNELID>
-e ORB_CLOUD_MQTT_KEY=<AGENTKEY>
-e PKTVISOR_PCAP_IFACE_DEFAULT=mock
ns1labs/orb-agent:develop
```

localhost, mock

This command is useful for connecting to a local develop environment, perhaps running on Docker compose. Note that the "mock" interface will generate random traffic rather than observe real traffic.

```
docker run -d --net=host
-e ORB_CLOUD_ADDRESS=localhost
-e ORB_CLOUD_MQTT_ID=7fb96f61-5de1-4f56-99d6-4eb8b43f8bad
-e ORB_CLOUD_MQTT_CHANNEL_ID=3e60e85d-4414-44d9-b564-0c1874898a4d
-e ORB_CLOUD_MQTT_KEY=44e42d90-aaef-45de-9bc2-2b2581eb30b3
-e PKTVISOR_PCAP_IFACE_DEFAULT=mock
-e ORB_TLS_VERIFY=false
ns1labs/orb-agent:develop
```

orb.live, eth0

This command is similar to one you would use on the orb.live SaaS platform

Getting started
- Register a new account
- Create an Agent
- Create an Agent Group
- Create a Sink
- Create a Policy
- Create a Dataset
- Visualize and alert on your metrics

Running Orb Agent
- Agent credentials
- Sample provisioning commands
- Configuration files
- Advanced auto-provisioning setup

Working with API Docs

Community

Contribute

Orb is an open source project born at NS1 Labs. Work with us on GitHub and star the project to show your interest.

Star

Contact

We are very interested to hear about your use cases, feature requests, and contribution ideas.

- Sign up to get Orb updates
- File an issue
- Follow our public work board
- Start a discussion
- Join us on Slack
- Check out the NS1 Labs YouTube channel
- Send mail to info@pktvisor.dev

Table of contents

Contribute
Contact
Explore
Articles
Conference Presentations