

OX PowerDNS Cloud Control

Monitoring

Feb 16, 2021

Release 1.0.0

©2021 by Open-Xchange AG and PowerDNS.COM BV. All rights reserved. Open-Xchange, PowerDNS, the Open-Xchange logo and PowerDNS logo are trademarks or registered trademarks of Open-Xchange AG. All other company and/or product names may be trademarks or registered trademarks of their owners. Information contained in this document is subject to change without notice.

Contents

1 Overview	1
1.1 Cloud Control Monitoring	1
1.2 Grafana Dashboards	2
1.2.1 Dashboard: overview	2
1.2.2 Dashboard: DNSdist performance	3
1.2.3 Dashboard: Recursor performance	3
2 Standalone Deployment	4
2.1 Getting Started	4
2.1.1 Install Tools	4
2.1.2 Download Helm Chart	4
2.1.3 Download Helm Chart (OCI - experimental)	5
2.1.4 Deploying Cloud Control Monitoring	5
2.1.5 Accessing Grafana	6
2.1.6 Accessing Prometheus	6
2.2 Helm Chart configuration	7
2.2.1 grafana	7
2.3 Images	8
3 Integrated Deployment	9

1 Overview

1.1 Cloud Control Monitoring

Cloud Control Monitoring helps provide insight into Cloud Control deployments. The monitoring stack is built using the following components:

- **Grafana** - Visualisation
- **Prometheus** - Metrics gathering & storage
- **kube-state-metrics** - Service that exposes Kubernetes metrics to prometheus

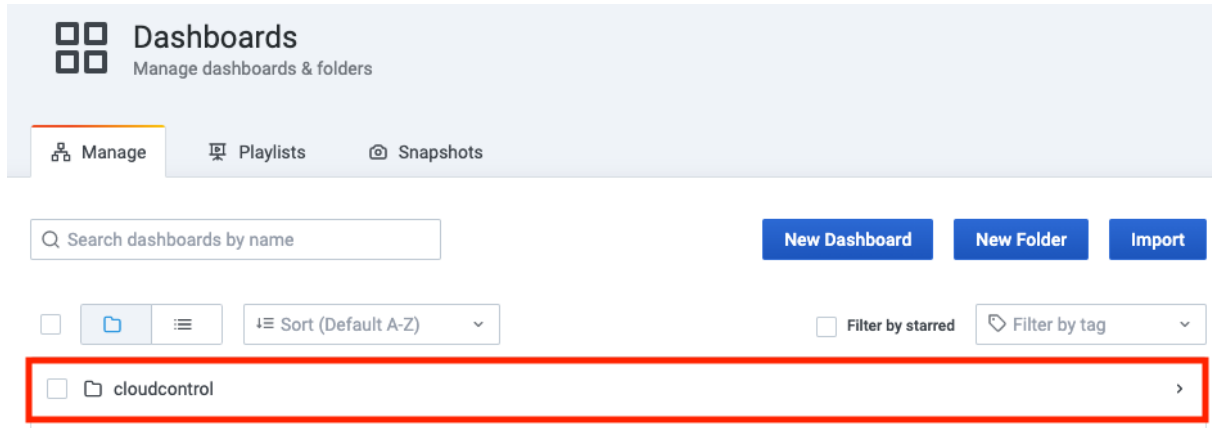
The stack can be deployed in 2 ways:

- **Standalone** - Intended for shortlived (demo & development) environments. Deploys Grafana & Prometheus, but storage of Prometheus is non-persistent and pods are not highly available.
- **Integrated** - Assumes a production-grade Grafana & Prometheus deployment pre-exist on the Kubernetes cluster. Artifacts are available to import the necessary Cloud Control dashboards into Grafana + scrapers & service discovery for Prometheus.

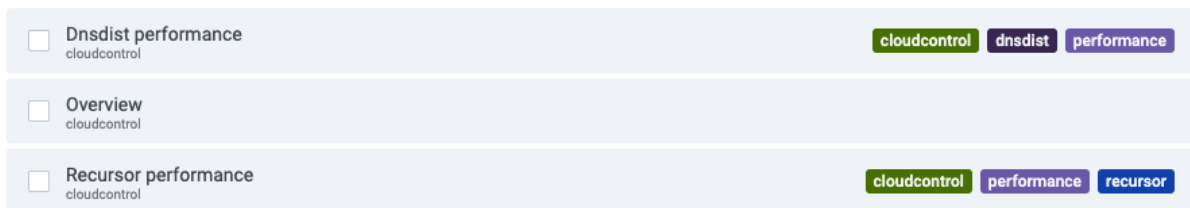
Deployment instructions for each variant is described in more detail in the following chapters.

1.2 Grafana Dashboards

Both deployment methods will lead to the same result in Grafana, a dashboards folder named 'cloudcontrol':



And within that folder there are several dashboards:

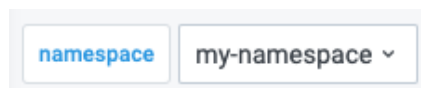


Note: The dashboards are still under development as of this release, more metrics will be added gradually.

1.2.1 Dashboard: overview

The overview dashboard exists as a landing page for Cloud Control in Grafana and shows the namespaces that were discovered to hold Cloud Control deployments + basic information about the deployed PowerDNS components within those namespaces.

You can select the Namespace you wish to view using the *namespace* dropdown top-left of the dashboard:



1.2.2 Dashboard: DNSdist performance

The DNSdist performance dashboard allows for in-depth insight into the performance of DNSdist instances deployed into a namespace. You can use the *namespace* and *instances* dropdowns to select the scope of the data presented by the dashboard. The list of *instances* depends on the selected *namespace*, so make sure to select a namespace first and then narrow your selection of instances (if you wish to analyse specific instances).

1.2.3 Dashboard: Recursor performance

The recursor performance dashboard allows for in-depth insight into the performance of recursor instances deployed into a namespace. You can use the *namespace* and *instances* dropdowns to select the scope of the data presented by the dashboard. The list of *instances* depends on the selected *namespace*, so make sure to select a namespace first and then narrow your selection of instances (if you wish to analyse specific instances).

2 Standalone Deployment

2.1 Getting Started

2.1.1 Install Tools

You will need the following software on the machine from which you want to deploy Cloud Control Monitoring:

- Kubectl (Configured for your target Kubernetes cluster)
- Helm v3 (<https://helm.sh/docs/intro/install/>)

2.1.2 Download Helm Chart

Cloud Control Monitoring Helm Charts are available on the Open-Xchange registry, located at: registry.open-xchange.com.

There are several methods for obtaining Helm Charts using Helm's CLI, in this chapter we are using a method that copies the chart locally to your filesystem prior to using it. Any Helm-supported method will work, but you will need to adjust the commands in this guide accordingly if you wish to utilise a different method.

First step will be to make Helm aware of the Cloud Control repository (replace username & password with your OX registry credentials):

```
helm repo add cloudcontrol https://registry.open-xchange.com/chartrepo/cloudcontrol \
--username=REGISTRY_USERNAME_HERE --password=REGISTRY_PASSWORD_HERE
```

Once the repository has been added you can pull the Cloud Control Monitoring Helm Charts. To pull the monitoring Helm Chart and export it to your current working directory use the following commands:

```
# The release we're working with
CCTAG=1.0.0

# Ensure repo data is up-to-date
helm repo update

# Pull the Helm Chart & unpack
helm pull cloudcontrol/monitoring -d . --version=$CCTAG --untar
```

2.1.3 Download Helm Chart (OCI - experimental)

Helm v3 includes OCI support, although it is still experimental and expected to contain breaking changes prior to official release. Cloud Control Monitoring is made available in OCI format, but recommended deployment methodology remains the traditional way of working with Helm Charts.

To work with Helm's OCI capabilities, you need to enable Helm OCI support using:

```
export HELM_EXPERIMENTAL_OCI=1
```

Cloud Control Monitoring Helm Charts are available on the Open-Xchange registry, located at: registry.open-xchange.com. To make sure Helm can interact with the registry, use the Helm registry login command:

```
helm registry login registry.open-xchange.com
```

Once authenticated you can pull & export the Cloud Control Monitoring Helm Charts. To pull the monitoring Helm Chart and export it to your current working directory use the following commands:

```
# The release we're working with
CCTAG=1.0.0

# Pull the Helm Chart from the registry
helm chart pull registry.open-xchange.com/cloudcontrol/helm-cc-monitoring:$CCTAG

# Export the Helm Chart to the current working directory
helm chart export registry.open-xchange.com/cloudcontrol/helm-cc-monitoring:$CCTAG -d .
```

2.1.4 Deploying Cloud Control Monitoring

To deploy the monitoring stack without any customization you can use the following steps:

```
# The namespace
CC_MON_NAMESPACE=ccmon
HELM_RELEASE=ccmon

helm install $HELM_RELEASE ./monitoring --namespace $CC_MON_NAMESPACE --create-namespace
```

Note: you can remove `--create-namespace` if you have an existing namespace to deploy into

The resulting namespace should contain a Service object for each component. Using the default deployment will result in each Service being of type 'ClusterIP', meaning you will need to expose them somehow. Several options for exposing these Services would be:

- **One-time use** - Setup port-forwarding using kubectl (see below example)
- **Ingress** - You'll need to create an Ingress object specific to your cluster's Ingress controller to expose the service
- **Loadbalancer / NodePort** - Configurable via additional parameters in the Helm configuration values (see *Helm Chart configuration* chapter)

2.1.5 Accessing Grafana

You can use kubectl's port-forwarding to quickly access the Grafana service:

```
# The namespace
CC_MON_NAMESPACE=cmon

kubectl --namespace=$CC_MON_NAMESPACE port-forward svc/grafana 3000:http
```

You can now visit Grafana at: <http://localhost:3000/>

For a more permanent method of access Grafana, refer to the *Helm Chart Configuration* chapter to configure the Service object.

2.1.6 Accessing Prometheus

You can use kubectl's port-forwarding to quickly access the Prometheus service:

```
# The namespace
CC_MON_NAMESPACE=cmon

kubectl --namespace=$CC_MON_NAMESPACE port-forward svc/prometheus 9090:http
```

You can now visit Grafana at: <http://localhost:9090/>

2.2 Helm Chart configuration

The parameters listed in this chapter can be used to configure the standalone deployment of Cloud Control Monitoring. You can store any overrides in a YAML file (commonly `values.yaml`) and pass it as an additional `values=values.yaml` argument to the Helm commandline.

2.2.1 grafana

- **grafanaAnnotations:** Annotations which will be applied to the Grafana service object. Particularly useful for `Type=Loadbalancer`.
- **grafanaNodePort:** If type is `NodePort` or `Loadbalancer`, force this to be the `nodeport`. By default a random `NodePort` is chosen to avoid allocation conflicts.
- **grafanaPort:** Port on which traffic will be accepted by the service. (default: `80`)
- **grafanaServiceType:** Type of service (`NodePort`, `ClusterIP`, `Loadbalancer`) (default: `ClusterIP`)

2.3 Images

Following table lists the images used by this version of Cloud Control Monitoring:

Component	Image	Tag
grafana	docker.io/grafana/grafana	7.3.6
kubestatemetrics	quay.io/coreos/kube-state-metrics	v1.9.7
prometheus	docker.io/prom/prometheus	v2.24.0

3 Integrated Deployment

Assumes a production-grade Grafana & Prometheus deployment pre-exist on the Kubernetes cluster. Deployment instructions will be added later.