

;One-click Chainlink deployment



Leo Vigna

Jun 10, 2020 · 6 min read



kubernetes



Kubeapps

Quickly deploy a Chainlink node using the KubeApps user interface.

Chainlink

Chainlink connects blockchain smart contracts with external data such as prices, identities, or API requests by the use of Chainlink nodes. Nodes watch the Ethereum blockchain for oracle requests and get compensated in LINK for updating the on-chain data. One core use has been the development of decentralized price feeds which you can browse at <https://feeds.link>

Before you read this

In this guide we will be deploying a Chainlink node to a Kubernetes cluster with the click of a button. If you only wish to run a standalone Chainlink node using Docker, we'd recommend the official [documentation](#).

While many existing official operators use Kubernetes for container orchestration, our deployment will differ by using KubeApps on top of our cluster to enable one-click deployment of our Chainlink node. In the long-term, having KubeApps running can make

it easy to upgrade to newer Chainlink versions and also help us deploy add-ons such as an external Chainlink Gas Updater or a Chainlink GraphQL API.

Requirements / Tools

Alright, so we may have lied a teeny weensy bit: this guide will help you deploy a Chainlink node with the click of a button, but **only** once you have KubeApps running. However, once you do, you will also be able to upgrade to future versions and install add-ons with the **simple click of a button**.

Here are some of the components we will be using to deploy our Chainlink node. If you're missing any, keep reading as we will provide suggestions on getting each of these setup. If you have these components setup already, you can skip to the "KubeApps Chainlink Deployment" section on how to deploy the Chainlink node using the GUI.

- Ethereum Node
- PostgreSQL Database
- Kubernetes (K8s) Cluster
- Helm configured with K8s cluster
- KubeApps Deployment on K8s Cluster

Ethereum Node

Your Chainlink node will need an Ethereum Node to watch and fulfill oracle requests on the blockchain. If looking to create a simple deployment we recommend a service such as infura.io or fiaws.io which both offer free tiers giving you access to JSON-RPC requests. If looking to maximize security, running your own node is always the better solution. You can even combine an in-house Ethereum node with a Node as a Service (NaaS) thanks to <https://github.com/OracleFinder/ChainlinkEthFailover>

PostgreSQL Server

The Chainlink Node uses an external PostgreSQL database to keep track of transactions, requests, and other important data required for oracle request execution. If you already have a PostgreSQL server, simply create a new database named "chainlink". If you do not have a PostgreSQL server, you can either use a service like AWS RDS to setup your database, or use KubeApps to deploy the PostgreSQL Chart (see later section on KubeApps).

Kubernetes (K8s) Cluster

K8s enables container orchestration. Container orchestration is the dynamic management of a cluster of containers. Once you have a cluster up and running, you'll be able to deploy multiple independent docker containers. While you could deploy a PostgreSQL and Chainlink docker container directly using the K8s API, Helm and KubeApps will greatly simplify this process by storing our configuration and adding a UI Layer.

If you do not have a K8s cluster you can use a cloud service such as AWS EKS or if looking for a light-weight solution for your own server, you can checkout MicroK8s <https://microk8s.io/>

Helm

Once you have your K8s cluster running, we will install Helm. Helm uses “charts” which store configuration on your software's K8s deployment architecture. Charts can then be shared through repositories.

Use the Helm [Quickstart guide](#) to install Helm v3 on your K8s cluster.

You can find the Vulcan Link charts at <https://github.com/vulcanlink/charts> (in development). While you could use the Helm CLI to deploy these charts, [KubeApps](#) will let you deploy these with a more intuitive frontend.

KubeApps

KubeApps is an opensource UI front-end developed by Bitnami designed to help you manage your K8s deployments. Once KubeApps is deployed on your K8s cluster, you will be able to deploy Helm charts with the click of a button. Follow the [Quickstart guide](#) to install KubeApps on your K8s cluster using Helm v3.

KubeApps Chainlink Deployment

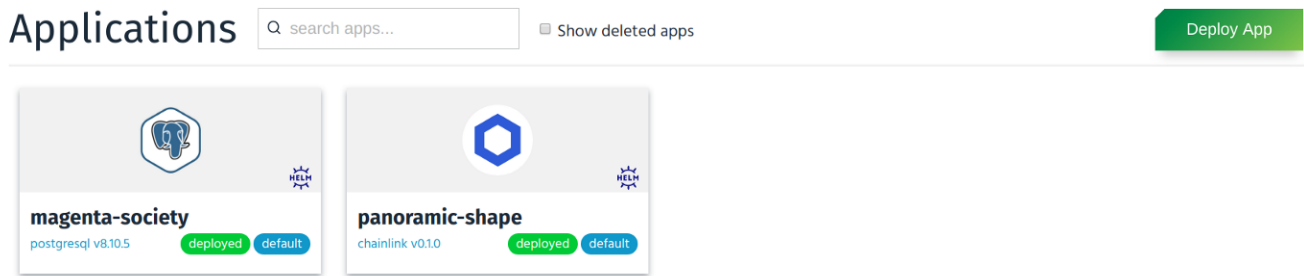
Using KubeApps

Now that KubeApps is running, this is where the magic happens. If you followed the KubeApps Getting Started guide you should be able to access KubeApps at <http://localhost:8080>.

If you can't, make sure you are port forwarding the kubeapps service.

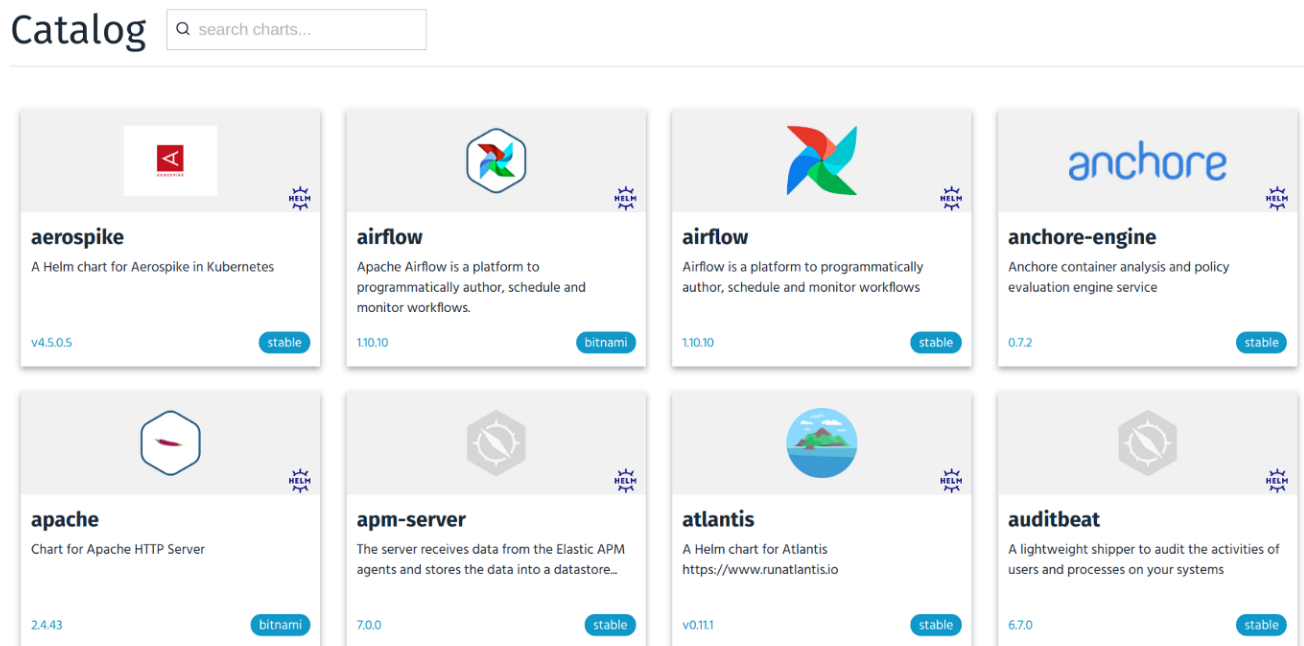
```
kubectl port-forward -n kubeapps svc/kubeapps 8080:80
```

The “Applications” page will show you Helm charts currently deployed on your cluster.



KubeApps deployed charts

The “Catalog” page lists possible charts you can choose to deploy on your cluster, a kind of App Store for K8s deployments.



KubeApps chart catalog

If you wish to deploy a PostgreSQL server on your K8s cluster, you can do so by searching PostgreSQL and deploying the respective chart. Keep reading as the steps for this will be similar as to how we deploy the Chainlink node.

At the top right, you can switch between different namespaces on your cluster. In general, simply use the “default” namespace for your deployments.

Adding the Vulcan Link Chart Repo

The Chainlink node isn’t part of the Catalog as the “@stable” repository does not contain a Chainlink chart. However, Vulcan Link has created a helm chart for Chainlink, and

deployed it to our own independent Helm repository hosted on GitHub.

To add the Vulcan Link repo, first change the namespace to “All Namespaces” to add the repo to all namespaces. Go to “Configuration > App Repositories”.

Repo	Namespace	URL	Actions
bitnami	kubeapps	https://charts.bitnami.com/bitnami	Delete Edit Refresh
incubator	kubeapps	https://kubernetes-charts-incubator.storage.googleapis.com	Delete Edit Refresh
stable	kubeapps	https://kubernetes-charts.storage.googleapis.com	Delete Edit Refresh
svc-cat	kubeapps	https://svc-catalog-charts.storage.googleapis.com	Delete Edit Refresh
vulcanlink	kubeapps	https://vulcanlink.github.io/charts/	Delete Edit Refresh

[Add App Repository](#) [Refresh All](#)

Add an Helm Chart Repository to KubeApps

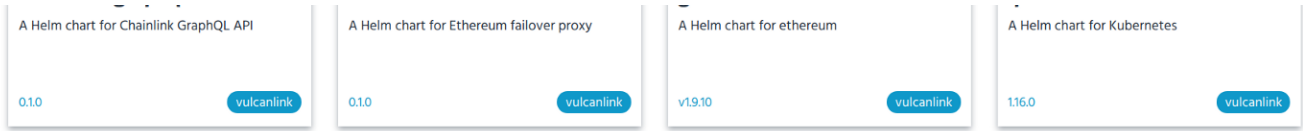
Click “Add App Repository” and use these values:

```
name: vulcanlink
url: https://vulcanlink.github.io/charts/
```

You should now be able to view the Vulcan Link charts on the Catalog page. To test this, simply type “vulcanlink” in the Catalog search bar.

Catalog

bitcoind A Helm chart for Kubernetes 1.16.0 vulcanlink	chainlink A Helm chart for Chainlink 0.8.3 vulcanlink	chainlink-api A Helm chart for Chainlink GraphQL API 0.1.0 vulcanlink	chainlink-gas-updater A Helm chart for Chainlink Gas Updater v0.1.3 vulcanlink
chainlink-graphql	eth-failover	geth	ipfs



Deploying a Chainlink Node

Switch the namespace back to “default”, select the “chainlink” chart, and click “Deploy”.

Kubeapps Applications Catalog Service Instances (alpha) NAMESPACE: default Configuration Logout

vulcanlink/chainlink
0.8.3 - vulcanlink
A Helm chart for Chainlink Deploy

Chainlink Helm Chart

Helm chart deploying a <http://chain.link> oracle node.

Chart Versions
0.1.0 - Jun 9, 2020

You can now use the form to enter your environment variables such as the URLs to the Ethereum Node and PostgreSQL database. You can also add additional environment variables by editing the config key under the Values section.

vulcanlink/chainlink

Name

Version

Form ? Values (YAML) Changes

Chainlink Node Details

Login Email	<input type="text"/>
Login Password	<input type="text"/>
Wallet Password	<input type="text"/>
Postgres database uri	<input type="text" value="postgresql://postgres@localhost:5432/chainlink"/>
Database timeout tolerance	<input type="text" value="500ms"/>
Link Token Contract Address	<input type="text" value="0x514910771af9ca656af840dff83e8264ecf986ca"/>

Ethereum Node URL

ws://localhost:8546

After clicking submit, your Chainlink deployment should pop up in the “Applications” section.

Ready

1 Pod

panoramic-shape
A Helm chart for Chainlink

App Version: 0.8.3
Chart Version: 0.1.0

Up to date

Access URLs
The current application does not expose a public URL.

Notes

```
1. Get the application URL by running these commands:
export POD_NAME=$(kubectl get pods --namespace default -l
"app.kubernetes.io/name=chainlink,app.kubernetes.io/instance=panoramic-shape" -o jsonpath="{.items[0].metadata.name}")
echo "Visit http://127.0.0.1:8080 to use your application"
kubectl --namespace default port-forward $POD_NAME 8080:6688
```

Deployments

NAME	DESIRED	UP-TO-DATE	AVAILABLE
panoramic-shape-chainlink	1	1	1

Services

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
panoramic-shape-chainlink	ClusterIP	10.152.183.157	None	6688/TCP

Of course, CLI commands should also show you that the Chainlink node has been correctly deployed.

```
kubectl get deployments #Using K8s API
helm ls #Using Helm
```

To get access to the Chainlink UI, follow the commands from “Notes”:

```
1. Get the application URL by running these commands:
export POD_NAME=$(kubectl get pods --namespace default -l
"app.kubernetes.io/name=chainlink,app.kubernetes.io/instance=panoramic-shape" -o jsonpath="{.items[0].metadata.name}")
echo "Visit http://127.0.0.1:8080 to use your application"
kubectl --namespace default port-forward $POD_NAME 8080:6688
```

You might have to change the 8080 port to 8081 if you are already serving KubeApps on 8080

You should now be able to visit the Chainlink Node dashboard at <http://localhost:8080>

The END

That's it! Your Chainlink node is deployed! All there is left to do now is setting up an on-chain Oracle contract and funding your node to start [Fulfilling Requests](#).

About Vulcan Link

We are a Paris-based Chainlink node operator working on actively maintaining 30+ reliable data feeds and developing decentralized applications that leverage smart contracts with external data. We believe in building trust through transparency by contributing to opensource projects. If you'd like us to add other data feeds to [feeds.link](#), feel free to reach out through our links below!

Find us at [vulcan.link](#)

Follow us on Twitter [@vulcanlink](#) for updates on new projects like this one.

If you'd like to contribute, join us on [Telegram](#) and [Discord](#)