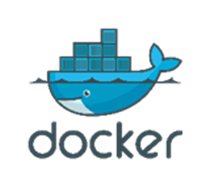
∞

Taking Cyber Security Seriously

**Expandable Defensive Cyber Operations Platform  
Configuration Guide – Master Server (CentOS)**

****



****

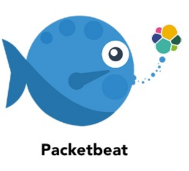












Sealing Technologies, Inc.

7134 Columbia Gateway Dr.

Suite 160

Columbia, MD 21046

|  |  |  |
| --- | --- | --- |
| Version | Author | Notes |
| 1.0 | Markus & Michael | Initial Draft. Program Reviews populated with instructions. |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[Introduction to the EDCOP 4](#_Toc493083864)

[Overview 4](#_Toc493083865)

[Basic Requirements 4](#_Toc493083866)

[Docker 4](#_Toc493083867)

[Installation 4](#_Toc493083868)

[Creating Docker Containers 6](#_Toc493083869)

[Environment Requirements 6](#_Toc493083870)

[Building the Master Containers 7](#_Toc493083871)

[Rancher 8](#_Toc493083872)

[Environment Requirements 8](#_Toc493083873)

[Installation 9](#_Toc493083874)

[Access Control 10](#_Toc493083875)

[Adding the Master Host 10](#_Toc493083876)

[Kubernetes 12](#_Toc493083877)

[Installation 12](#_Toc493083878)

[Labeling the Master Server 13](#_Toc493083879)

[Running the Master Applications 15](#_Toc493083880)

[Deleting Kubernetes 16](#_Toc493083881)

# Introduction to the EDCOP

This guide provides an overview of the Expandable Defense Cyber Operations Platform (EDCOP) concepts as well as configuration information, examples, and reference information pertaining to the tools utilized in the platform.

## Overview

The virtualized Network Security Monitor (NSM) supports security features that can protect your network against degradation as well upholding the confidentiality and integrity of the system through continuous monitoring.

The following components are covered in this configuration guide:

* Docker
* Rancher
* Kubernetes
* Elasticsearch
* Kibana
* Moloch

For more information on the core applications, see the concept of operations (CONOPS) document. After the completion of this guide, please move on to the **EDCOP – Node Configuration Guide** to continue the setup of the EDCOP environment.

## Basic Requirements

Before you begin this guide, it is assumed that you have already installed Cent OS 7, have obtained a **static IP address** for the host, and have the capability to access the internet. Basic Linux knowledge of the command line, file editing, and folder structuring is strongly recommended.

# Docker

## Installation

In order to utilize the EDCOP, setup of Docker is necessary.

1. Initially, ensure there are no older versions of Docker, or the Docker-engine by uninstalling the software alongside the dependencies.

$ sudo yum remove docker \

docker-common \

container-selinux \

docker-selinux \

docker-engine

1. Enable the EPEL repository:

$ sudo yum -y install epel-release

1. Install the required packages.

$ sudo yum install -y yum-utils device-mapper-persistent-data lvm2 git wget net-tools python-pip python-wheel

1. Add the docker repository:

$ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

1. Update the *yum* package index

$ sudo yum makecache fast

1. Install version 17.06 of Docker – be sure to install this exact version!

$ sudo yum -y install docker-ce-17.06.0.ce-1.el7.centos.x86\_64

1. Edit the etc/docker/daemon.json file. If it does not exist, simply create it.

$ sudo vi /etc/docker/daemon.json

The file should have the following contents:

{

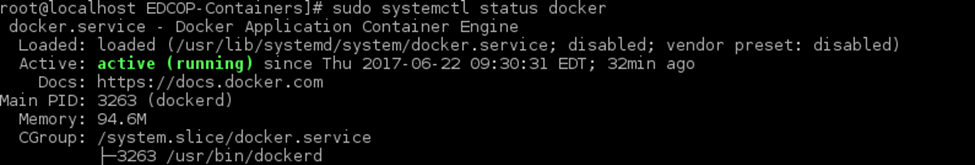
"storage-driver": "devicemapper"

}

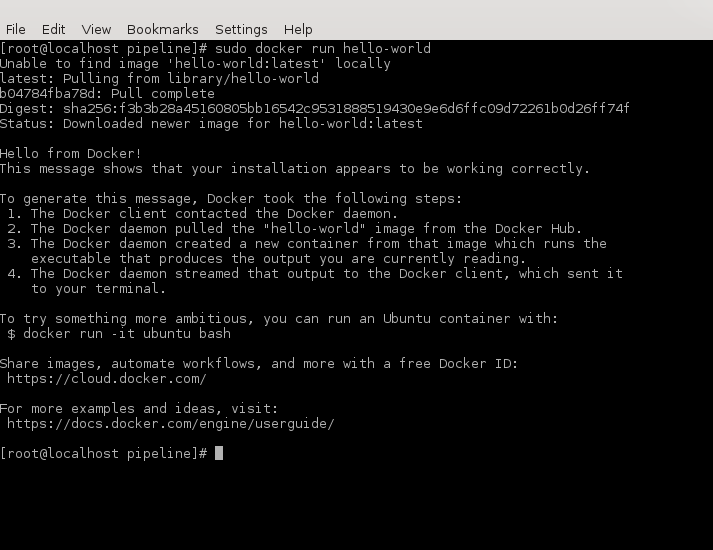
1. Start Docker

$ sudo systemctl start docker

1. Docker should now be up and running.

$ sudo systemctl status docker

1. To ensure that docker is correctly installed, run the hello-world image.

$ sudo docker run hello-world

For more information on Docker, please visit <https://doc.docker.com/engine/reference/commandline/docker>

# Creating Docker Containers

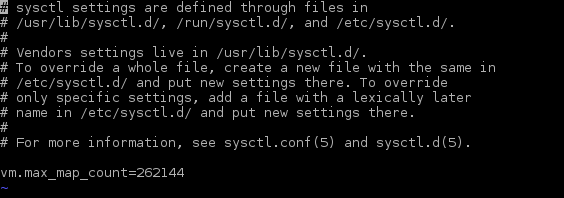
To begin setting up the master applications we need to create containers for them to run inside.

## Environment Requirements

1. Before creating your containers, you will need to increase your virtual memory map count to run Elasticsearch. The bare minimum is 262144, but feel free to increase the value even further:

$ sudo sysctl -w vm.max\_map\_count=262144

\*Note: this change will be reset anytime you restart your system. To make the changes permanent, you will need to edit the */etc/sysctl.conf* file, add the following line:

$ vm.max\_map\_count=262144

1. You will also need to create a directory for Elasticsearch store its data into. This directory will be crucial to our system, as pieces of our other instances will be formatted and stored here. Additionally, you will need to give this directory permissions.

$ sudo mkdir /esdata /data /data/moloch /data/moloch/raw /data/moloch/logs

$ sudo chmod 757 /esdata sudo chmod a+rwx /data/moloch/raw /data/moloch/logs

## Building the Master Containers

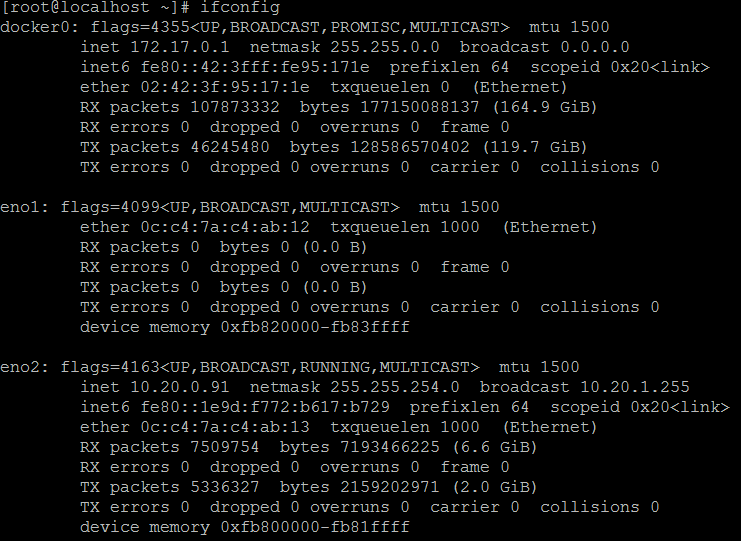
1. Now that we’ve made the preparations for running our containerized platform, you can clone the required files from our official GitHub page into an empty directory:

$ git clone https://github.com/sealingtech/edcop-cluster <$DIRECTORY>

1. Once the files finish copying, change directories into the newly created directory. Afterwards, change directories into the **“containers/master”** directory , and then give the *buildcontainers.sh* script permissions to run as an executable:

$ sudo chmod +x buildcontainers.sh

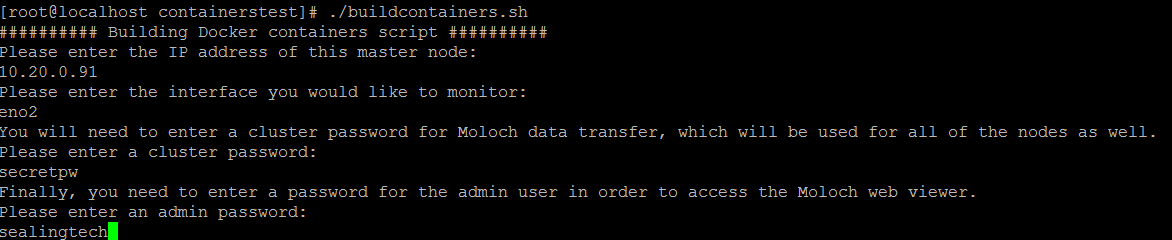
1. Now you’re going to need to know your host’s IP address and come up with two passwords for Moloch to use; one for accessing the web interface and one for data transfer. Lastly, you’ll need to figure out which interface you’re going to monitor. You can see a list of interfaces by using the ***ifconfig*** command:

$ sudo ifconfig

1. Once you have all of the information you’re going to need, run the script using the following command from within the aforementioned directory:

$ ./buildcontainers.sh

***\*Note: This will create container images for all the Master EDCOP applications that run within Docker. You may also use this script to make changes to the configuration files and then rebuild the images if changes are needed.***

1. Follow the on-screen instructions and then the script will build your master node containers using the information you have provided.

# Rancher

Instead of installing Kubernetes directly onto the hardware, we are going to use Rancher to automatically deploy Kubernetes clusters for us. Rancher comes with a native networking overlay that allows the use of many different nodes/hosts on the same virtualized network without the need for installing Kubernetes on each host by hand. This means we can spin up as many Rancher hosts as we want within a couple of minutes by connecting to the Rancher server and waiting for the automated deployment to finish!

## Environment Requirements

Before beginning the installation of Rancher, you must have a **supported version** of Docker installed and are required to open ports on the host to allow traffic to pass through. To discover which versions of Docker are supported, please visit: <http://rancher.com/docs/rancher/v1.6/en/hosts/#supported-docker-versions>. Once you have obtained a supported version, **do not** yum update the entire system, as it will install the latest version of docker and break your environment!

1. Before starting Rancher, expose the following ports and reload your firewall:

$ sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent

$ sudo firewall-cmd --zone=public --add-port=8080/udp --permanent

$ sudo firewall-cmd --zone=public --add-port=8005/tcp –permanent

$ sudo firewall-cmd --zone=public --add-port=8005/udp --permanent

$ sudo firewall-cmd --reload

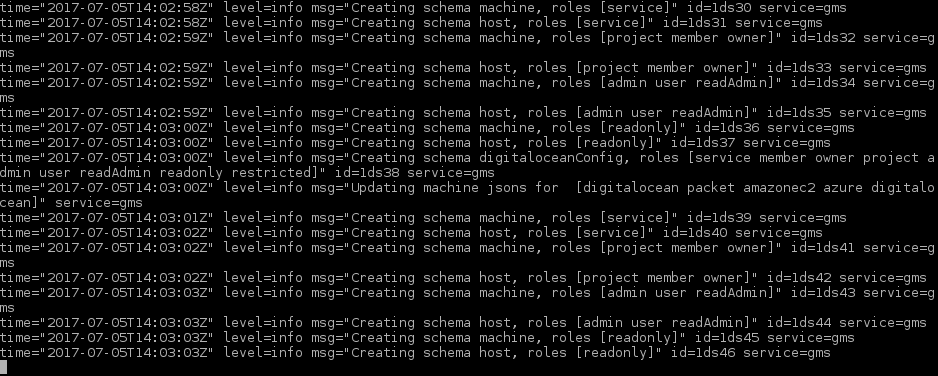
## Installation

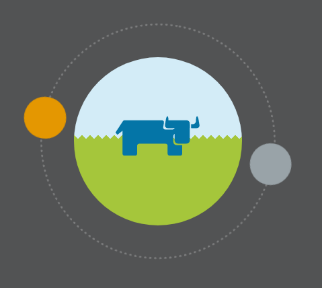
Rancher is installed within Docker containers and utilizes Docker to run all of its services, including Kubernetes! For more information, please visit Rancher’s official quick start guide: <http://rancher.com/docs/rancher/v1.6/en/quick-start-guide/>

1. To begin Rancher, replace the $MASTERIP with your node’s IP address, run the following, and then wait about 3 minutes for the set up to finish:

$ sudo docker run -d --restart=unless-stopped -p 8080:8080 -e CATTLE\_AGENT\_IP="$MASTERIP" rancher/server:stable

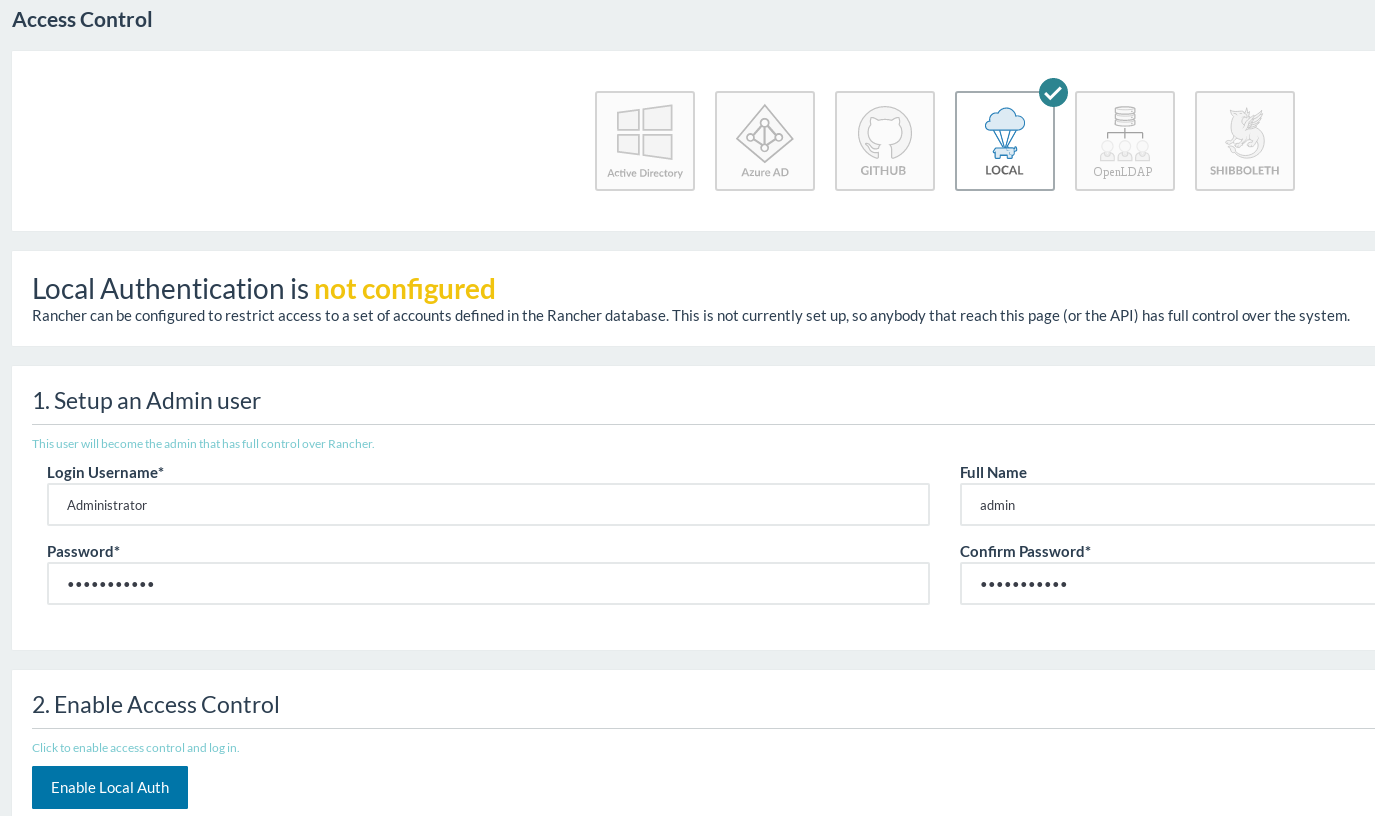
1. Check the logs of the Rancher container and wait until it’s ready to accept web interface connections (CTRL + C to quit viewing logs):

$ sudo docker logs -f <$CONTAINER-NAME>

1. You can now access Rancher’s web interface by opening your favorite web browser and navigating to your host’s IP address via port 8080: http://<$IP-ADDRESS>:8080

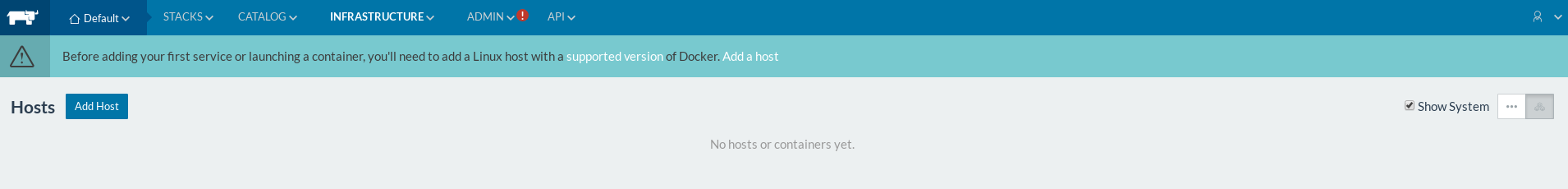
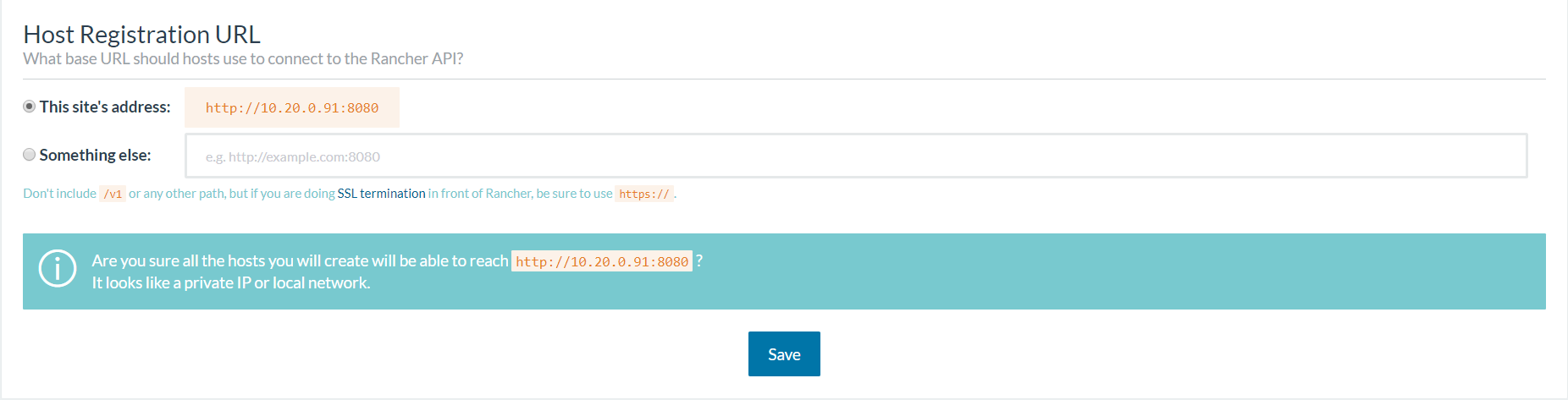
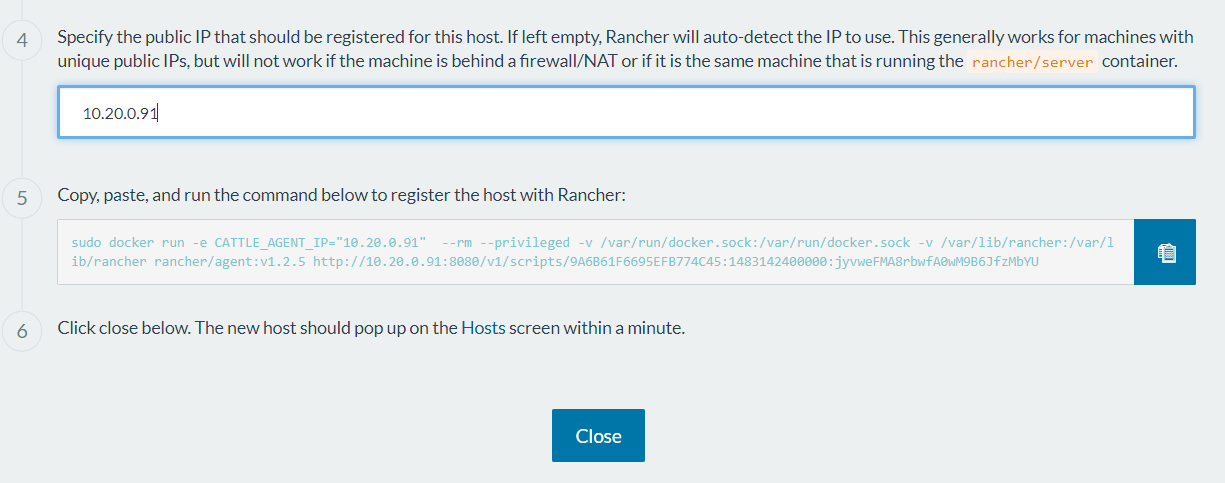
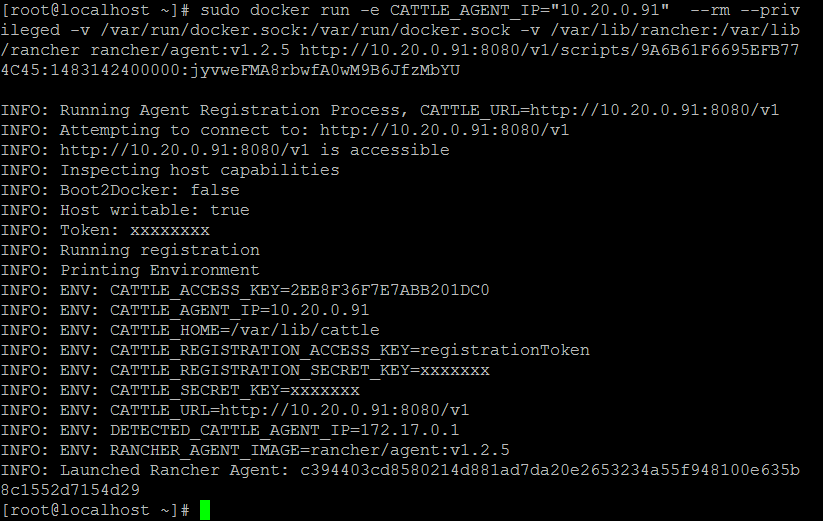
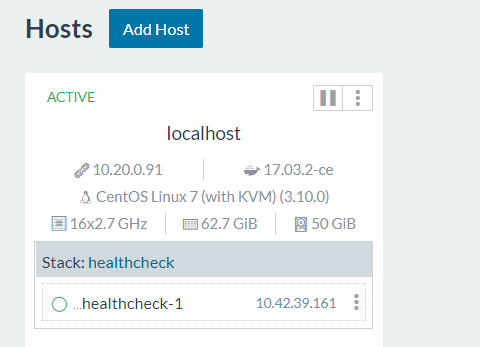
## Access Control

Although this setting is not absolutely required, it is strongly recommended to prevent unauthorized access to the cluster by anyone with the host’s IP address. Without access control, anybody who visits the unsecured webpage will have full control of your cluster!

1. Open the Rancher web UI and then navigate to the *Access Control* page within the *Admin* tab. Click on Local access control, and then create an administrator that has control of the cluster:

## Adding the Master Host

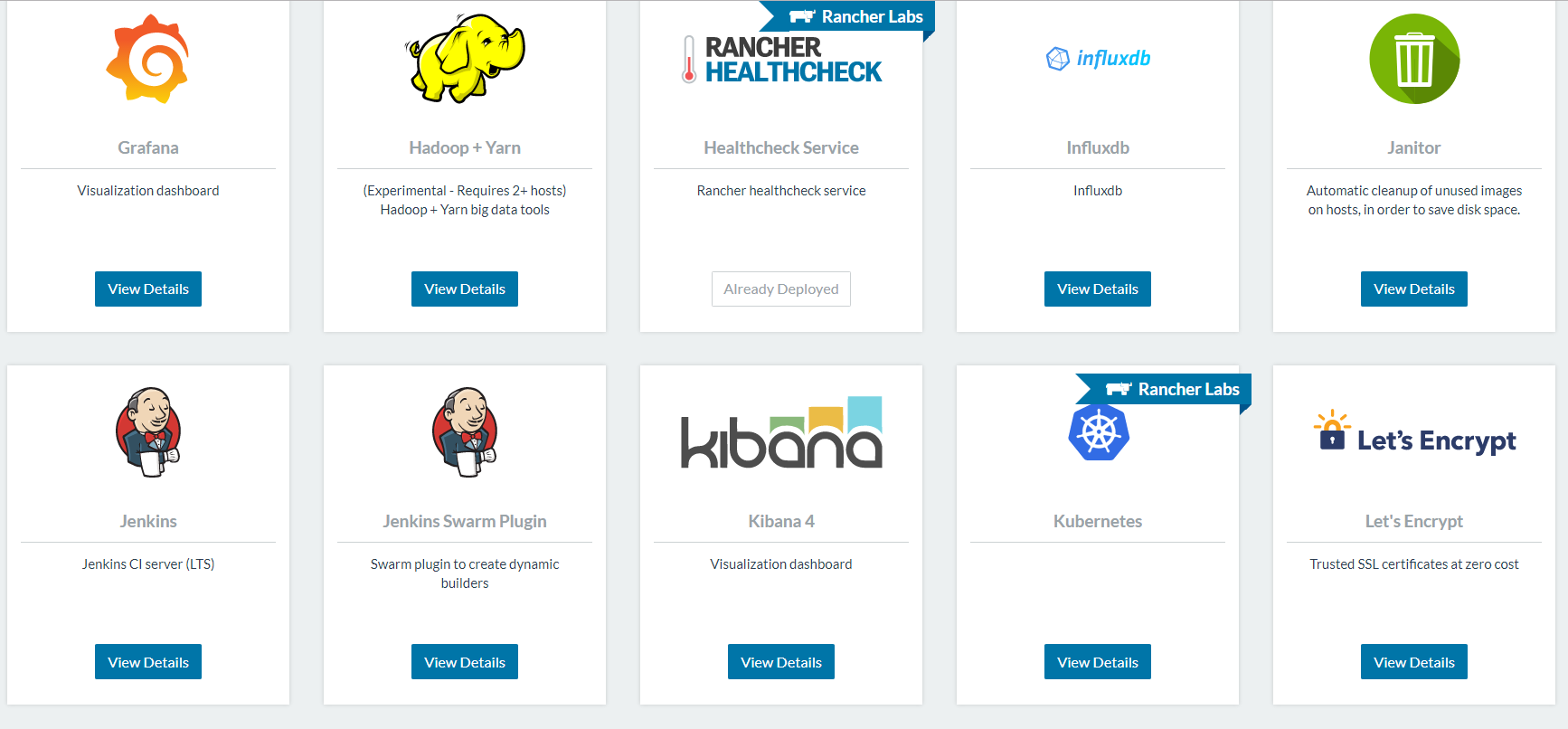
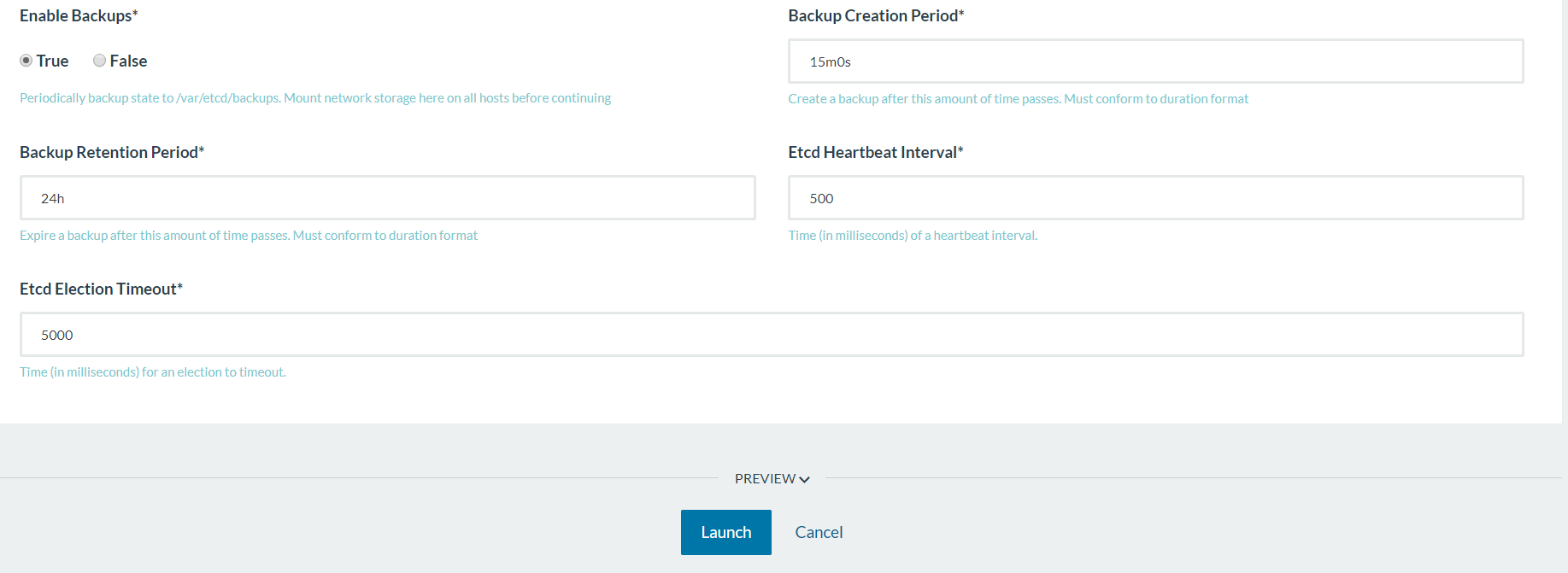
In order to run containers/services that Rancher can manage, your server will need to be registered as a Rancher Agent.

1. From your internal network, navigate to Rancher’s web interface at http://<$IP-ADDRESS>:8080 and navigate to the *Hosts* page from within the *Infrastructure* tab:
2. Click on *Add Host* and then make sure you are using “This site’s address” for the base URL. Confirm the IP address and then click save:
3. Add the master host running Rancher as a *Rancher Agent* that can run containers recognizable by Rancher and orchestrated by Kubernetes. Input the master host’s IP address and then copy the generated command:
4. Switch back to the command line on the master host, and paste the command:
5. Wait for Rancher to start up all of its agent services and then check the web interface to make sure the host has been added:
6. You can add other physical node hosts running a supported version of Docker the same way, but this will be covered in the ***EDCOP – Node Configuration Guide***. Before adding nodes, we need to set up Kubernetes and deploy our master applications.

# Kubernetes

Luckily for us, Kubernetes doesn’t need any special environment requirements and can be deployed by Rancher automatically!

## Installation

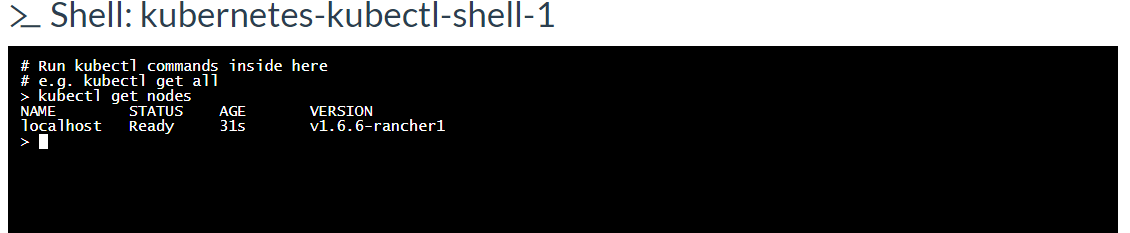
1. Open the Rancher web interface, then navigate to the *Catalog* tab and select all catalogs. Afterwards, scroll down the list until you see Kubernetes like so:
2. Click on view details and leave all the settings as default. Click Launch to deploy Kubernetes to your master server.
3. Screen ClippingKubernetes will now deploy a master server on the single host you’ve added. Once it’s done installing, your Rancher web UI will replace the Stacks and Catalog tabs with one Kubernetes tab. From there, you can access the Kubernetes dashboard and kubectl shell within your browser.

For more information on Kubernetes within Rancher, please visit <http://rancher.com/docs/rancher/v1.6/en/kubernetes/>

## Labeling the Master Server

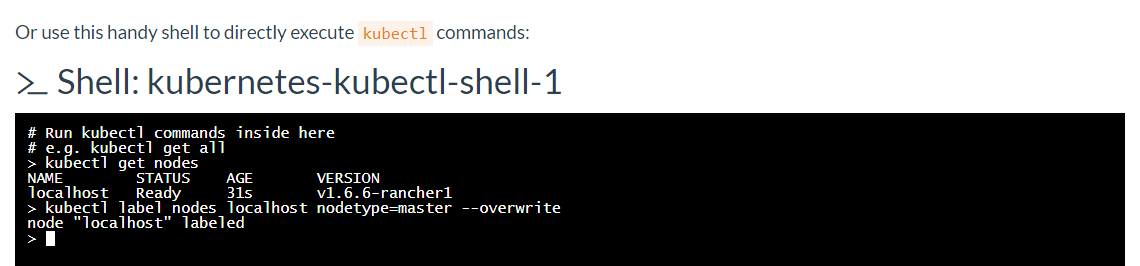
In order to prevent node applications from running on our master server, we need to label the master accordingly.

1. Navigate to the *CLI* page within the Kubernetes tab in order to use the shell:
2. Your node name should be the same as the name of the host, but to make sure begin by getting the information of the current nodes by typing:

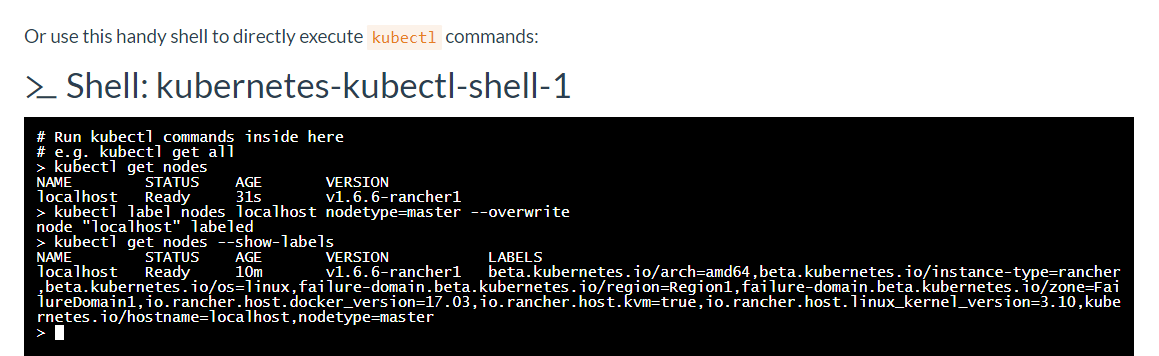
$ kubectl get nodes

*\*Note: In a production environment, your host names* ***must*** *be unique, do not leave them as the default!*

1. Now that we know our master node’s name, we can label it like so:

$ kubectl label nodes <$NODENAME> nodetype=master --overwrite

1. To make sure our label was applied correctly, you can get the node information again, but add the “show-labels” flag to see all the labels that belong to each node:

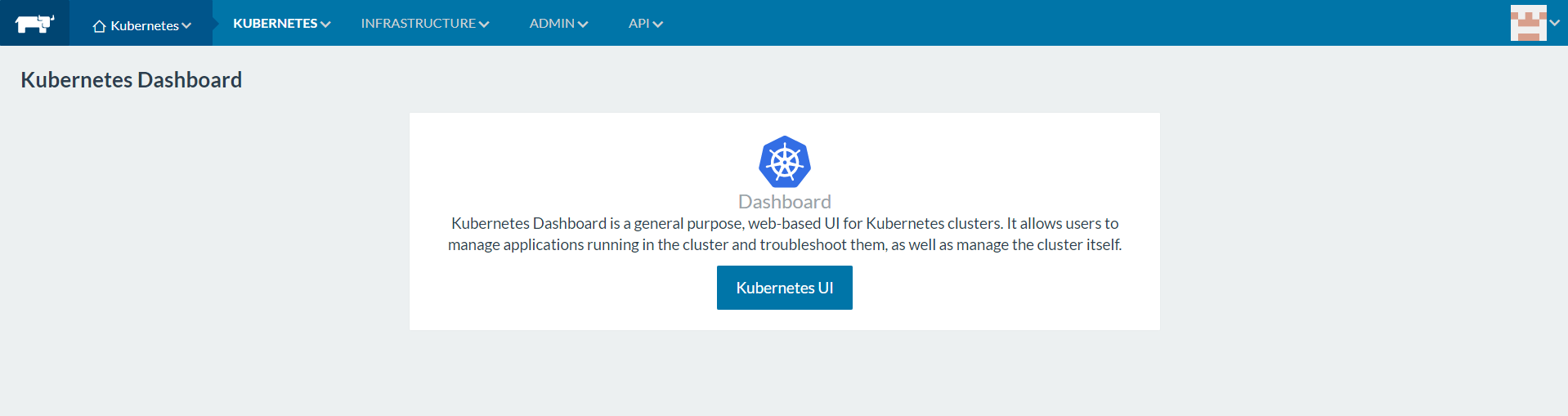
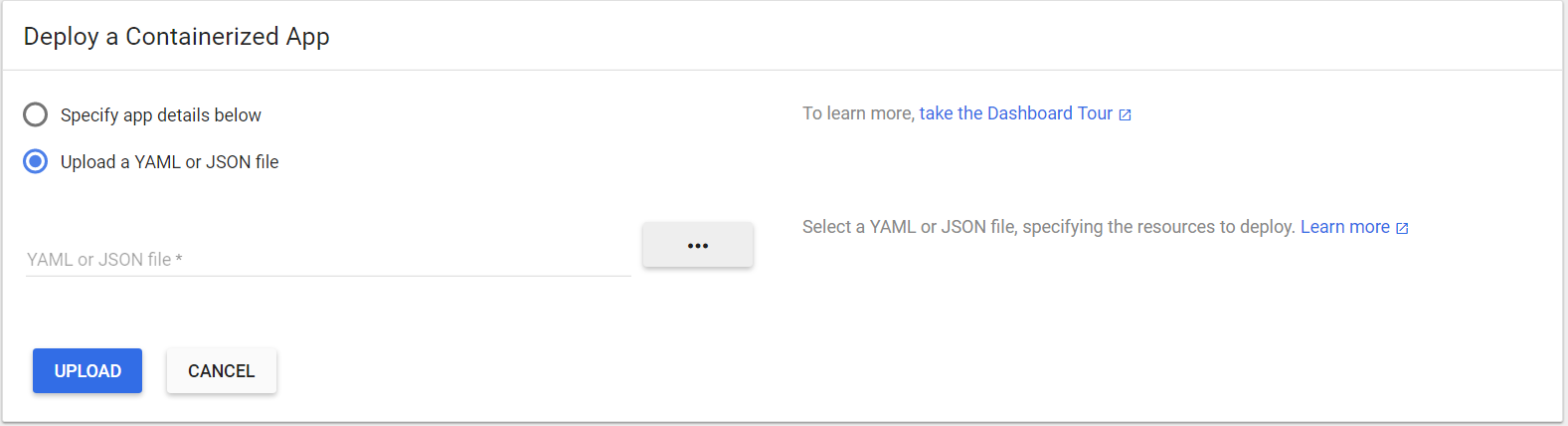
$ kubectl get nodes --show-labels

## Running the Master Applications

Now that we have our environment set up, we can run our master applications within the Kubernetes web UI. You can also run them through the kubectl shell as well, but uploading preconfigured YAML files is easier.

1. If you’re accessing the Rancher + Kubernetes webpages through an external host, the website will search your local filesystem for the YAML files. If you’re using the same host Rancher is installed on, the YAML files will be located within the directory you cloned earlier inside the subdirectory named **kubernetes.** You can always visit our GitHub and clone or download the YAML files if you need them again:

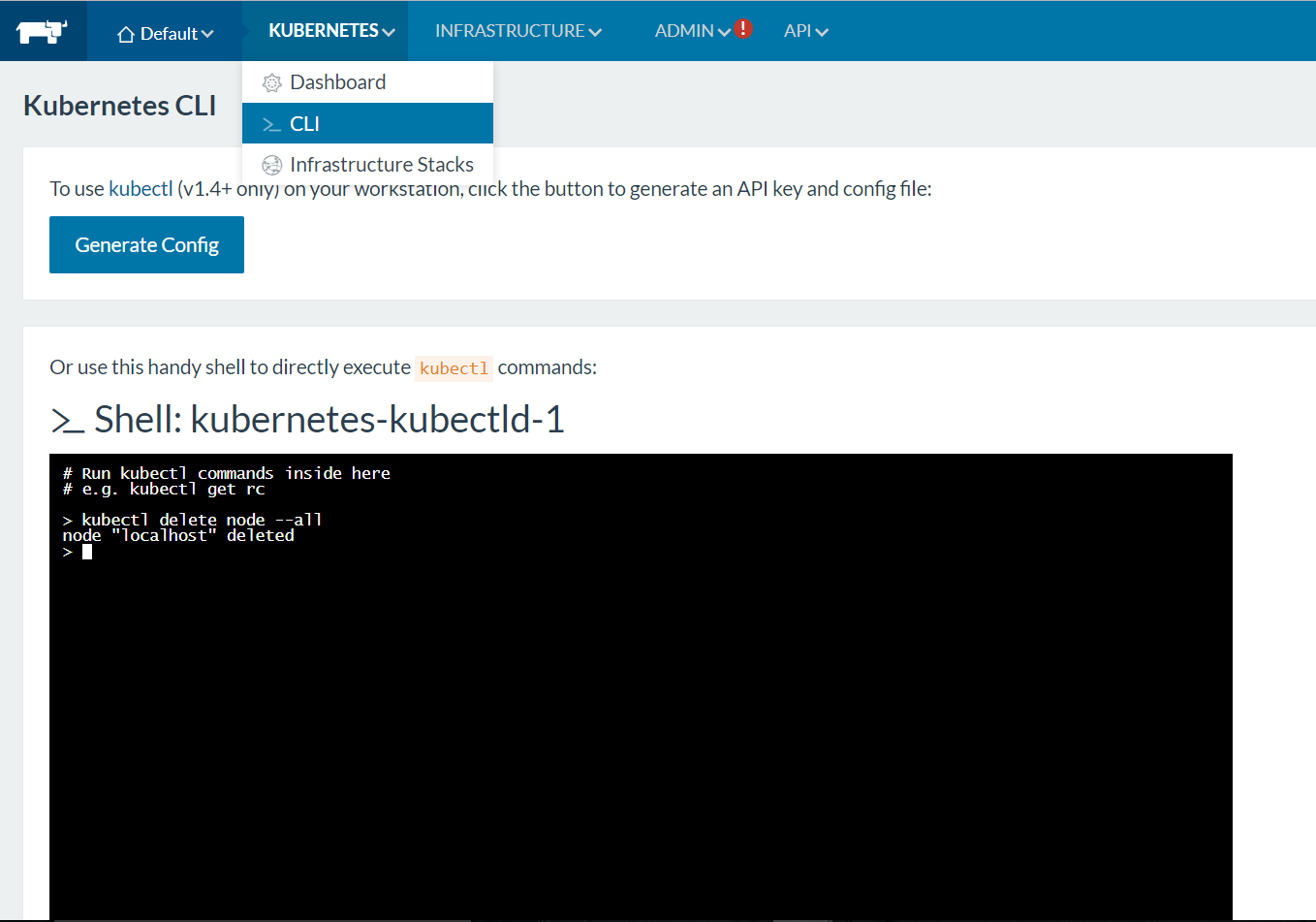
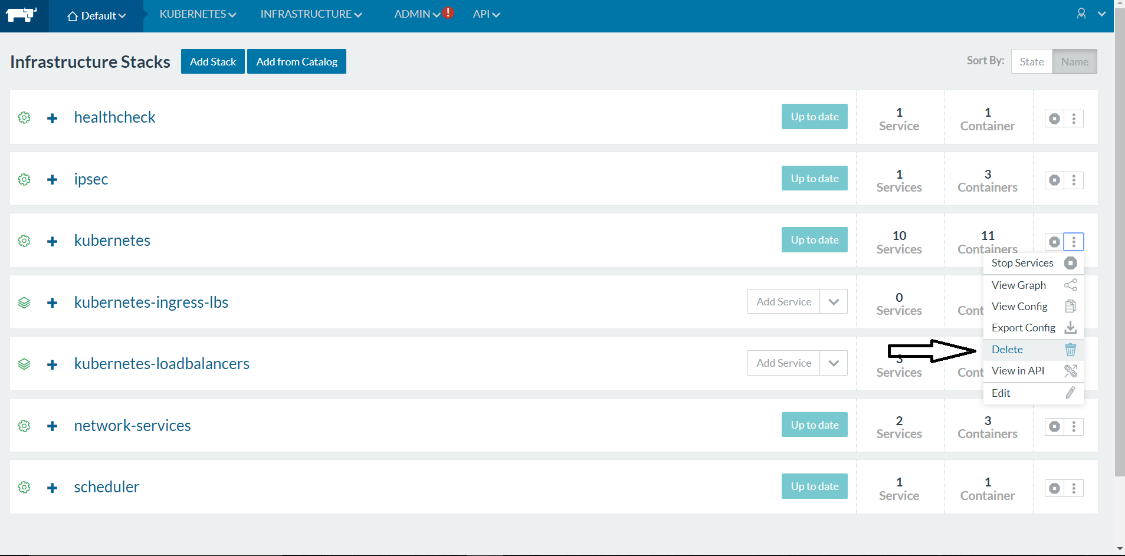
$ git clone https://github.com/sealingtech/edcop-cluster <$DIRECTORY>

1. Open the Kubernetes web dashboard through Rancher’s Kubernetes tab:
2. Click on create in the upper right hand corner.
3. Select “Upload a YAML or JSON file” and then click on the three dots to search for your files.
4. Please upload the application files in the following order, and be sure to upload the service YAML file **before** its corresponding application file:
5. Elasticsearch-Master + service file
6. Kibana + service file
7. Moloch-Master

*\*Note: After starting these applications, DO NOT start any of the node apps because they will fail without the proper container images and node hosts. Please refer to the* ***EDCOP – Node Configuration Guide*** *for instructions on how to start the node applications.*

## Deleting Kubernetes

If you have chosen to delete the Kubernetes orchestration, it is vital to clean up your pods and persistent data if you wish to continue using your environment and hosts.

1. Before deleting the Kubernetes stack, delete the nodes that are connected to the Kubernetes stack.
2. Navigate to the Infrastructure stacks and delete Kubernetes. Automatically, deleting the main Kubernetes stack will delete the ingress, as well as the load-balancers.
3. Upon deletion of the infrastructure stacks, persistent data still remains local onto all of the hosts you have added to Kubernetes. The named volume of etcd needs to be removed, as well as the backups from **all** of the hosts that were a part of this environment.

$ docker volume rm etcd

$ rm -rf /var/etcd/backups/\*

Failure to follow these steps may damage your environment can result in having an unstable Kubernetes system stack next time it is deployed. For more information, please visit: <http://rancher.com/docs/rancher/v1.5/en/kubernetes/deleting/>