Exercise 14a

Create a Kubernetes Cluster in DigitalOcean and Deploy an app

Prior Knowledge
Unix Command Line Shell
YAML

Learning Objectives
Introduction to Kubernetes

Software Requirements
Browser
kubectl
k9s

Overview

In this exercise we are going to sign up to Digital Ocean to get some free credit, then instantiate a Kubernetes cluster in DO, then install an app onto the kubernetes cluster. Finally we will do some monitoring.

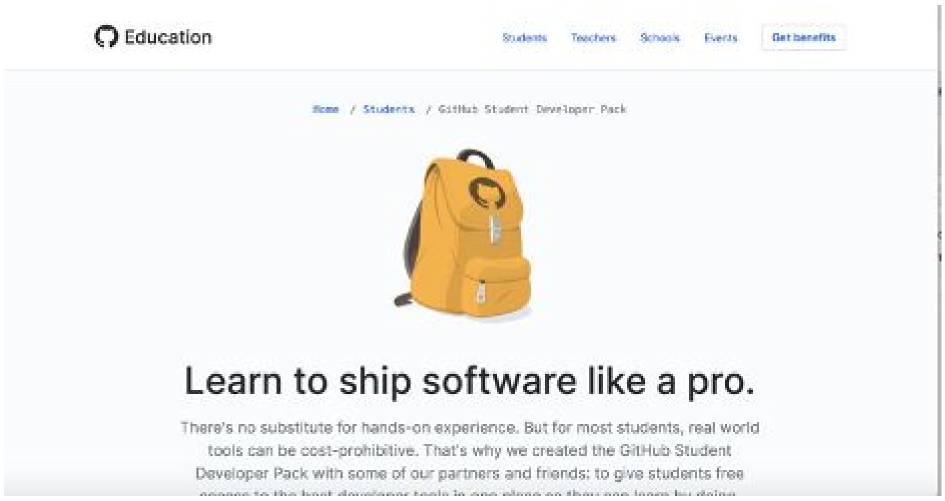
There is a follow up lab that then installs cassandra onto the cluster.

Although I've given instructions for using the Ubuntu VM, you can install kubectl/k9s on your own machines and do this from there as well.

PART A: SIGN UP WITH GITHUB EDUCATION AND DIGITAL OCEAN THEN START A K8S CLUSTER

If you already have a DigitalOcean account and credit, skip this step. Even if you already have an account, this exercise should cost less than $1 assuming you kill off the kubernetes cluster and all other volumes and load balancers when you are done.

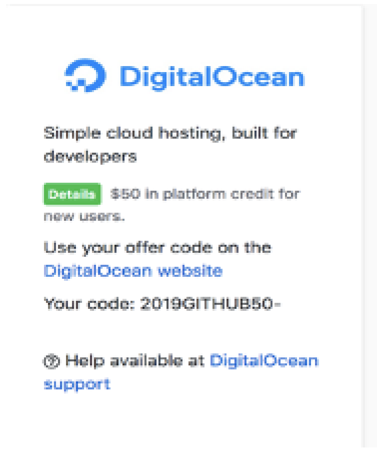
2. First sign up with Github Education (if you haven't already).
   https://education.github.com/
   This gives you $50 credit for DigitalOcean and credit for many other systems as well.



Learn to ship software like a pro.

There's no substitute for hands-on experience. But for most students, real world tools can be cost-prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free

   a. After signing in using your GitHub education account, you can get the promo here:
      https://education.github.com/pack/offers#digitalocean



Simple cloud hosting, built for developers

$50 in platform credit for new users.

Use your offer code on the DigitalOcean website

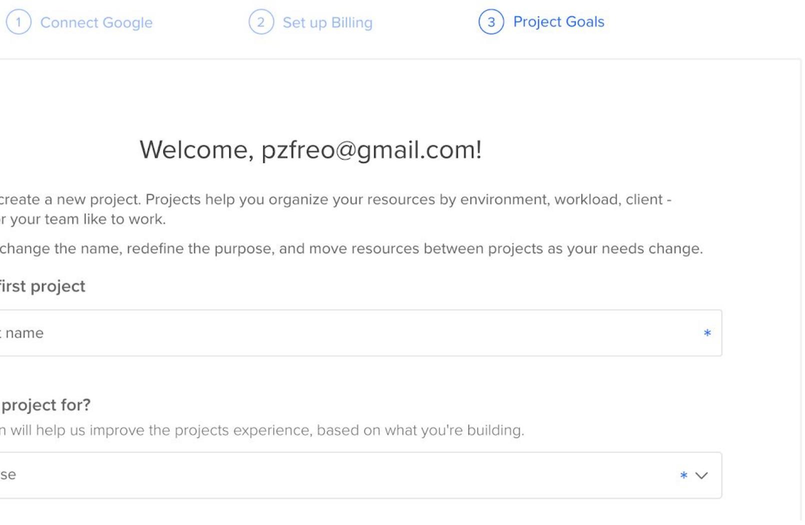
Your code: 2019GITHUB50-

Help available at DigitalOcean support

   b. Sign up and get a free Digital Ocean account.

c. Enter your billing details

d. You should see something like:



Welcome, pzfreo@gmail.com!

To begin, let's create a new project. Projects help you organize your resources by environment, workload, client - however you or your team like to work.

Note: You can change the name, redefine the purpose, and move resources between projects as your needs change.

Create your first project

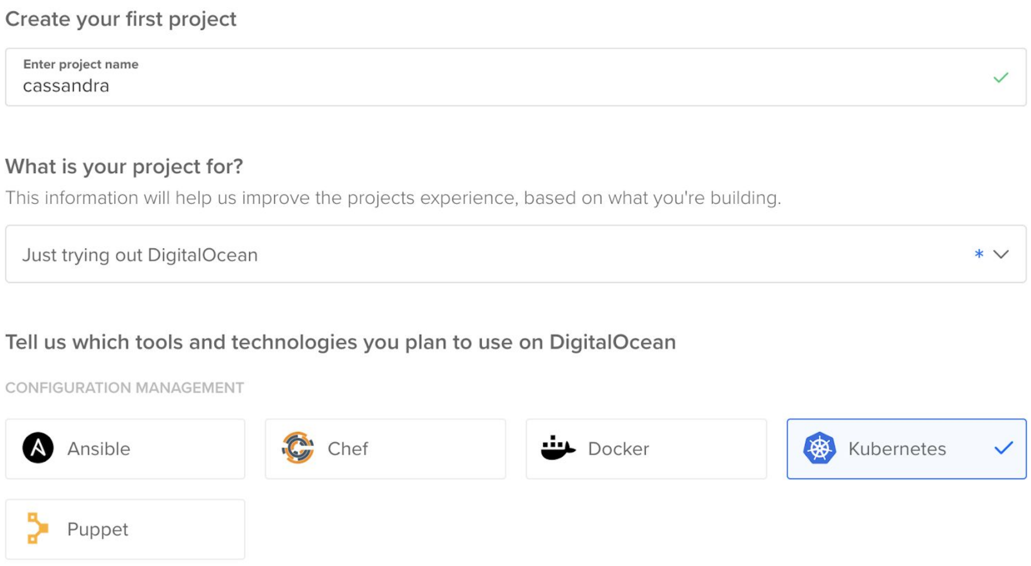
Enter project name *

What is your project for?

This information will help us improve the projects experience, based on what you're building.

Select purpose *

e. Give your project the name "cassandra":



Create your first project

Enter project name: cassandra ✓

What is your project for?

This information will help us improve the projects experience, based on what you're building.

Just trying out DigitalOcean *

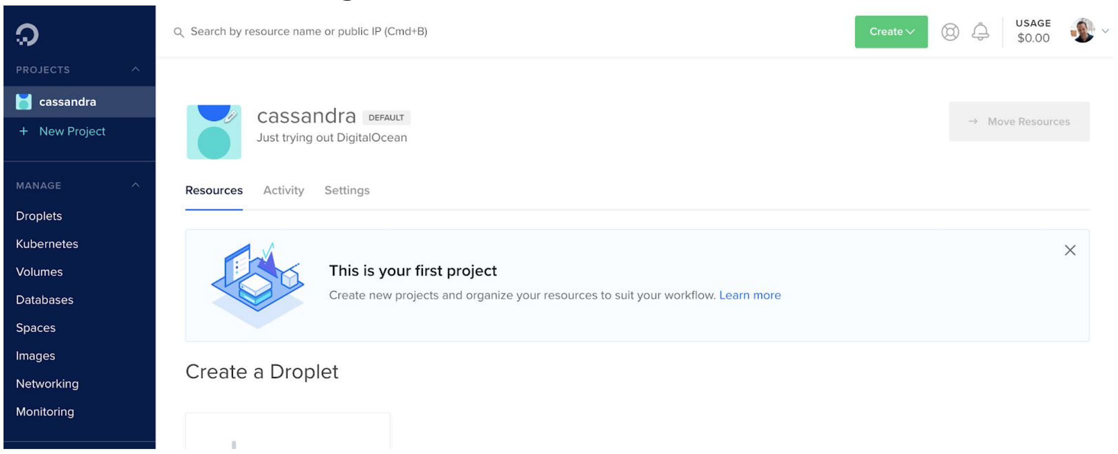
Tell us which tools and technologies you plan to use on DigitalOcean

CONFIGURATION MANAGEMENT

f. Select "Just trying out DigitalOcean" and tick Kubernetes

g. Click "Start" at the bottom of the page

h. You should see something like:



i. Go to Billing and enter your Github Education Promo code.

j. You are now ready to create the Kubernetes cluster.

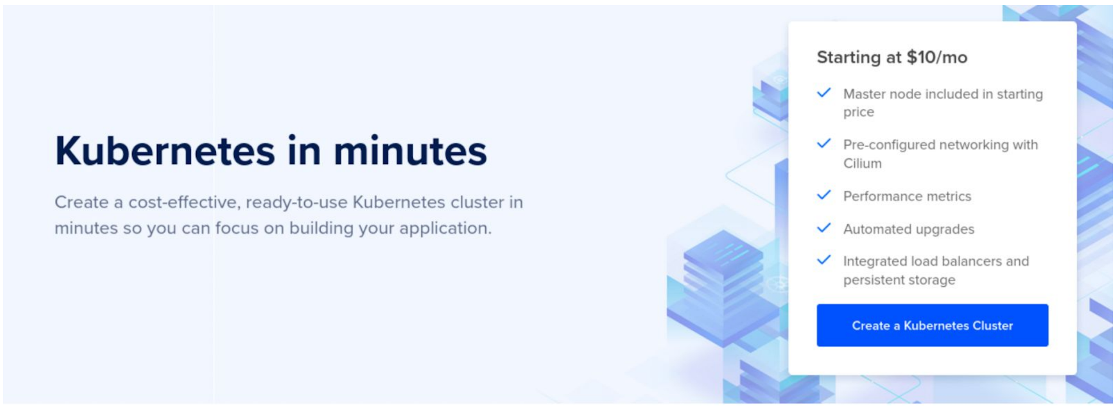
3. Before we create the Kubernetes cluster, we'd like to update the kubernetes CLI tool. By default the one in the Ubuntu package repo is out of date. We can fix that by doing the following commands (taken from https://kubernetes.io/docs/tasks/tools/install-kubectl/)

```
sudo apt-get update
sudo apt-get install -y apt-transport-https gnupg2
curl -s  https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a
/etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubectl
```

Your version should now be 1.18.x

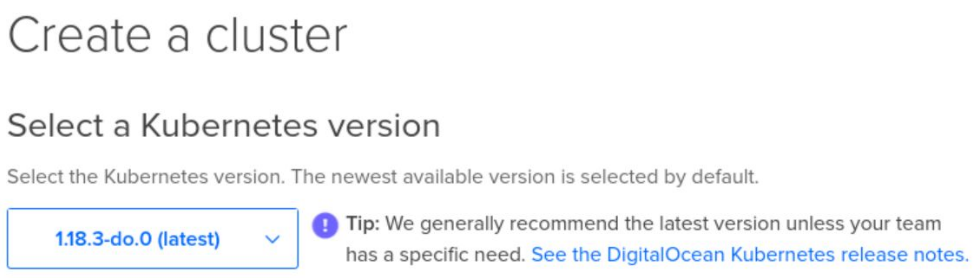
4. Go back to your Firefox / DigitalOcean window.

5. Click on Kubernetes in the left hand side. You should see:


Kubernetes in minutes
Create a cost-effective, ready-to-use Kubernetes cluster in minutes so you can focus on building your application.
Starting at $10/mo
Master node included in starting price
Pre-configured networking with Cilium
Performance metrics
Automated upgrades
Integrated load balancers and persistent storage
Create a Kubernetes Cluster

Now click **Create a Kubernetes Cluster**

Choose Kubernetes version 1.18.x


Create a cluster

Select a Kubernetes version

Select the Kubernetes version. The newest available version is selected by default.

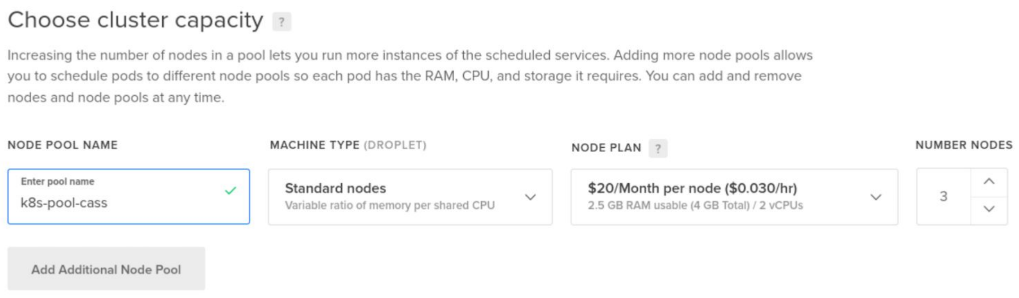
1.18.3-do.0 (latest)

Tip: We generally recommend the latest version unless your team has a specific need. See the DigitalOcean Kubernetes release notes.

(The kubectl client version and server version should be within one major revision of each other. e.g. 1.17 and 1.18 are compatible but 1.16 and 1.18 might not be).

6. Choose your nearest datacentre (e.g. London)

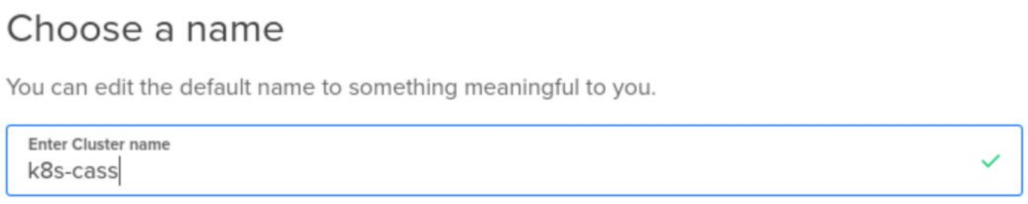
7. Choose the following:
   3 nodes
   Standard Nodes
   $20/month per node (2.5Gb RAM / 2 vcpus)


Choose cluster capacity

Increasing the number of nodes in a pool lets you run more instances of the scheduled services. Adding more node pools allows you to schedule pods to different node pools so each pod has the RAM, CPU, and storage it requires. You can add and remove nodes and node pools at any time.

NODE POOL NAME	MACHINE TYPE (DROPLET)	NODE PLAN	NUMBER NODES
Enter pool name k8s-pool-cass	Standard nodes Variable ratio of memory per shared CPU	$20/Month per node ($0.030/hr) 2.5 GB RAM usable (4 GB Total) / 2 vCPUs	3

8. Change the name to k8s-cass

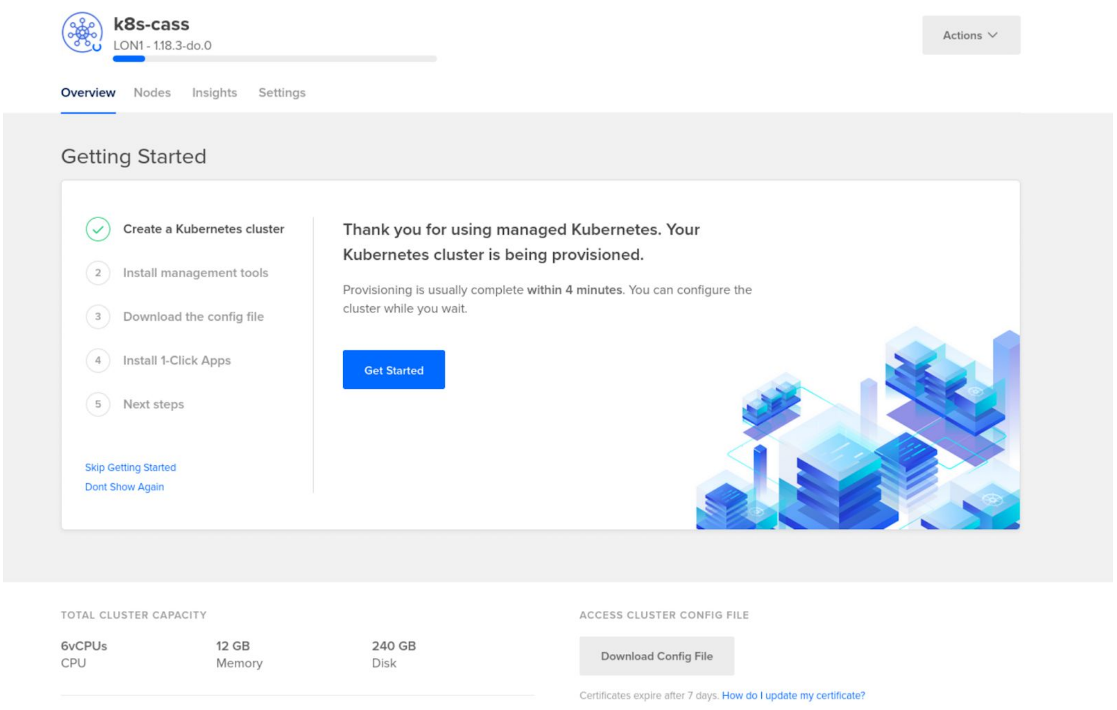

Choose a name

You can edit the default name to something meaningful to you.

Enter Cluster name
k8s-cass

9. Click **Create Cluster**

10. You should see:



k8s-cass
LON1 - 1.18.3-do.0

Getting Started

Thank you for using managed Kubernetes. Your Kubernetes cluster is being provisioned.

Provisioning is usually complete **within 4 minutes**. You can configure the cluster while you wait.

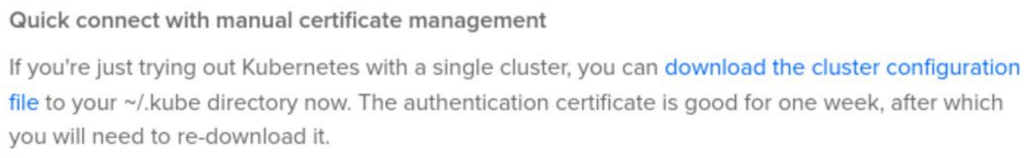
TOTAL CLUSTER CAPACITY		
6vCPUs CPU	12 GB Memory	240 GB Disk

ACCESS CLUSTER CONFIG FILE

Certificates expire after 7 days. How do I update my certificate?

11. There is a nice “checklist” of actions you can do with your cluster. Click on #2. We already have the management tools downloaded (at least kubectl), so we can **Continue**

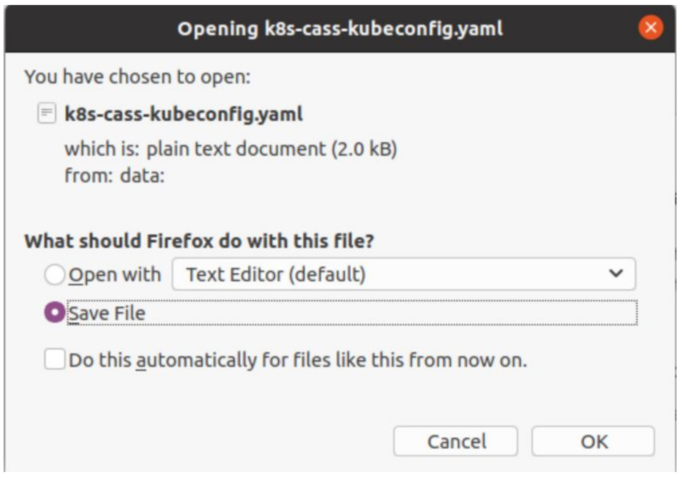
12. If you are going to use DO Kubernetes a lot, I suggest you read the section on using their `doctl` CLI tool. However, since I am more interested in you learning about kubernetes right now, I’d like you to follow the “manual” approach:

    Quick connect with manual certificate management

    If you're just trying out Kubernetes with a single cluster, you can download the cluster configuration file to your ~/.kube directory now. The authentication certificate is good for one week, after which you will need to re-download it.

    Click on “download the cluster configuration file”
    Save File

    Opening k8s-cass-kubeconfig.yaml

    You have chosen to open:

    k8s-cass-kubeconfig.yaml

    which is: plain text document (2.0 kB)
    from: data:

    What should Firefox do with this file?

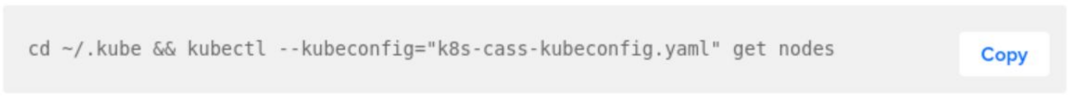
    ○ Open with Text Editor (default)
    [✓ Save File]

    ☐ Do this automatically for files like this from now on.

13. Open a terminal window and type:

```
mkdir ~/.kube
mv ~/Downloads/k8s-cass-kubeconfig.yaml ~/.kube/
```

Then you will see the command shown in the Web UI and execute that:



```
cd ~/.kube && kubectl --kubeconfig="k8s-cass-kubeconfig.yaml" get nodes
```

```
cd ~/.kube && kubectl
--kubeconfig="k8s-cass-kubeconfig.yaml" get nodes
```

You should see something like:

```
NAME                  STATUS   ROLES    AGE   VERSION
k8s-pool-cass-3o8i7   Ready    <none>   35m   v1.18.3
k8s-pool-cass-3o8ic   Ready    <none>   34m   v1.18.3
k8s-pool-cass-3o8iu   Ready    <none>   34m   v1.18.3
```

14. We want to use this config file all the time (without needing to do --kubeconfig="k8s-cass-kubeconfig.yaml" on every command):

    ```
    export KUBECONFIG=~/.kube/k8s-cass-kubeconfig.yaml
    ```

    (There are also other things we can do, but this works fine)

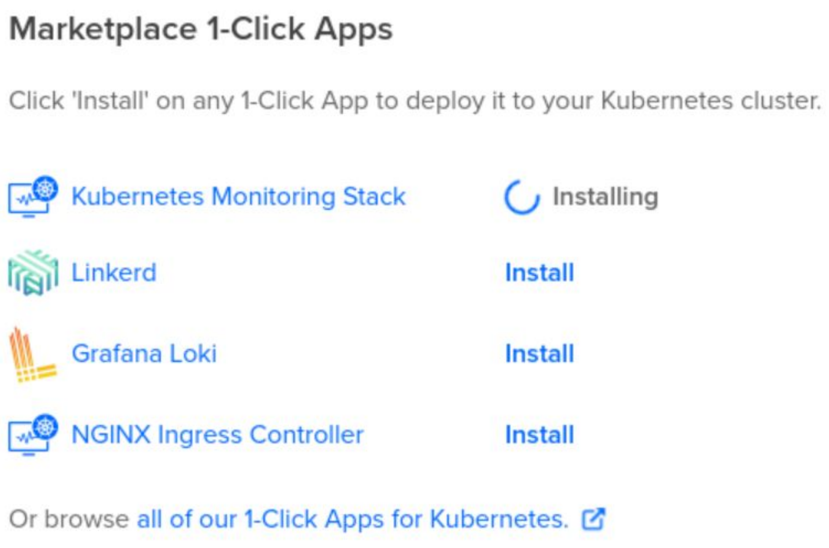
15. Check it works:

    ```
    kubectl get all
    ```

    You should see something like:

    ```
    NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
    service/kubernetes   ClusterIP   10.245.0.1   <none>        443/TCP   45m
    ```

16. Back in the Web UI, go to part 3 of the Getting Started, and install the **Kubernetes Monitoring Stack**



Marketplace 1-Click Apps

Click 'Install' on any 1-Click App to deploy it to your Kubernetes cluster.

Kubernetes Monitoring Stack	Installing
Linkerd	Install
Grafana Loki	Install
NGINX Ingress Controller	Install

Or browse all of our 1-Click Apps for Kubernetes.

PART B: INSTALL AN APP INTO K8S

17. Let's deploy a sample app:

    This app https://github.com/paulbouwer/hello-kubernetes is a great starting place to check out Kubernetes:

```
cd ~
git clone https://github.com/paulbouwer/hello-kubernetes.git
cd ~/hello-kubernetes
```

18. Now let's apply (install) this app into kubernetes:

```
kubectl apply -f yaml/hello-kubernetes.yaml
```

20. The install will be quick, but it might take a while to allocate an external address:

```
kubectl get all
```

```
NAME                                      READY   STATUS              RESTARTS   AGE
pod/hello-kubernetes-594f6f475f-4rksn    0/1     ContainerCreating   0          5s
pod/hello-kubernetes-594f6f475f-h25gz    0/1     ContainerCreating   0          5s
pod/hello-kubernetes-594f6f475f-sjd6h    0/1     ContainerCreating   0          5s

NAME                       TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)
AGE
service/hello-kubernetes   LoadBalancer   10.245.249.202   <pending>     80:30816/TCP   5s
service/kubernetes         ClusterIP      10.245.0.1       <none>        443/TCP
125m

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-kubernetes   0/3     3            0           5s

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-kubernetes-594f6f475f   3         3         0       5s
```

We are going to wait until everything is running (maybe a few minutes). When it's ready it should look like this:

```
NAME                                      READY   STATUS    RESTARTS   AGE
pod/hello-kubernetes-594f6f475f-4rksn    1/1     Running   0          5m2s
pod/hello-kubernetes-594f6f475f-h25gz    1/1     Running   0          5m2s
pod/hello-kubernetes-594f6f475f-sjd6h    1/1     Running   0          5m2s

NAME                       TYPE           CLUSTER-IP       EXTERNAL-IP     PORT(S)
AGE
service/hello-kubernetes   LoadBalancer   10.245.249.202   188.166.139.3   80:30816/TCP
service/kubernetes         ClusterIP      10.245.0.1       <none>          443/TCP

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-kubernetes   3/3     3            3           5m2s

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-kubernetes-594f6f475f   3         3         3       5m2s
```

While you are waiting, you can look at the YAML:

```yaml
apiVersion: v1
kind: Service
metadata:
 name: hello-kubernetes
spec:
 type: LoadBalancer
 ports:
 - port: 80
   targetPort: 8080
 selector:
   app: hello-kubernetes
---
apiVersion: apps/v1
kind: Deployment
metadata:
 name: hello-kubernetes
spec:
 replicas: 3
 selector:
   matchLabels:
     app: hello-kubernetes
 template:
   metadata:
     labels:
       app: hello-kubernetes
   spec:
     containers:
     - name: hello-kubernetes
       image: paulbouwer/hello-kubernetes:1.8
       ports:
       - containerPort: 8080
```

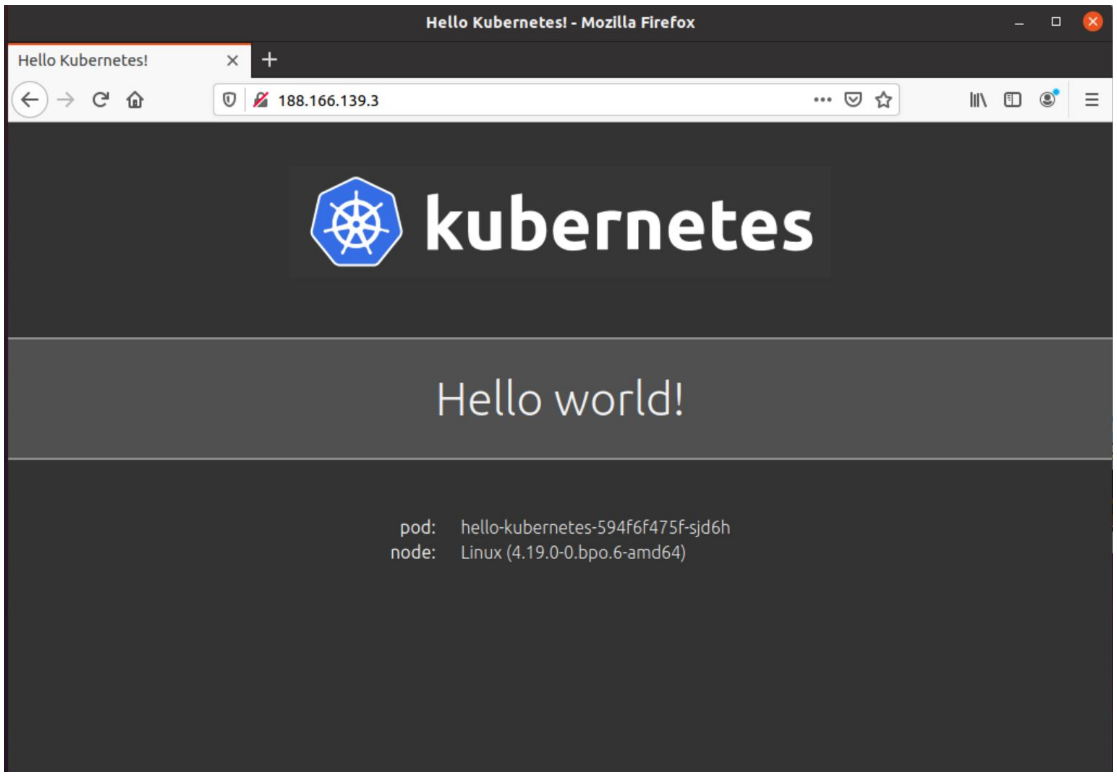
This basically defines a pod with 3 replicas containing a single container instance. There is then a load-balancer that balances load across the three replicas.

21. Now it should be running, get the external IP address:

```
kubectl get service hello-kubernetes

NAME               TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE
hello-kubernetes   LoadBalancer   10.245.249.202  188.166.139.3   80:30816/TCP   11m
```

22. Go to the external IP address in your browser:



Hello world!

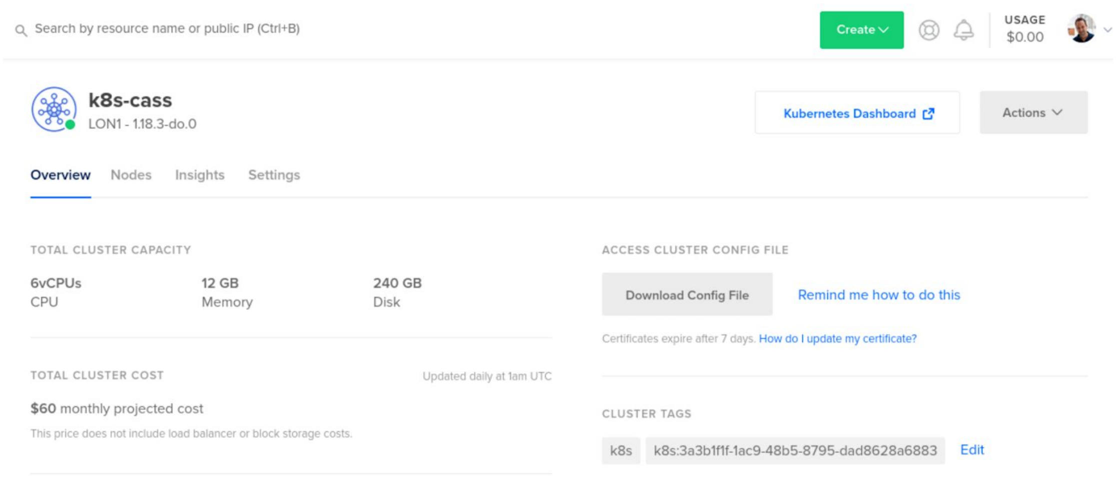
pod: hello-kubernetes-594f6f475f-sjd6h
node: Linux (4.19.0-0.bpo.6-amd64)

23. Keep reloading and you should see the pod details change.

24. Congrats - you've deployed a k8s app.

PART C: MONITORING

25. We can go and monitor the system from the DigitalOcean web ui. Navigate to the cluster info page:



k8s-cass
LON1 - 1.18.3-do.0

TOTAL CLUSTER CAPACITY

6vCPUs CPU | 12 GB Memory | 240 GB Disk

TOTAL CLUSTER COST — Updated daily at 1am UTC

$60 monthly projected cost

This price does not include load balancer or block storage costs.

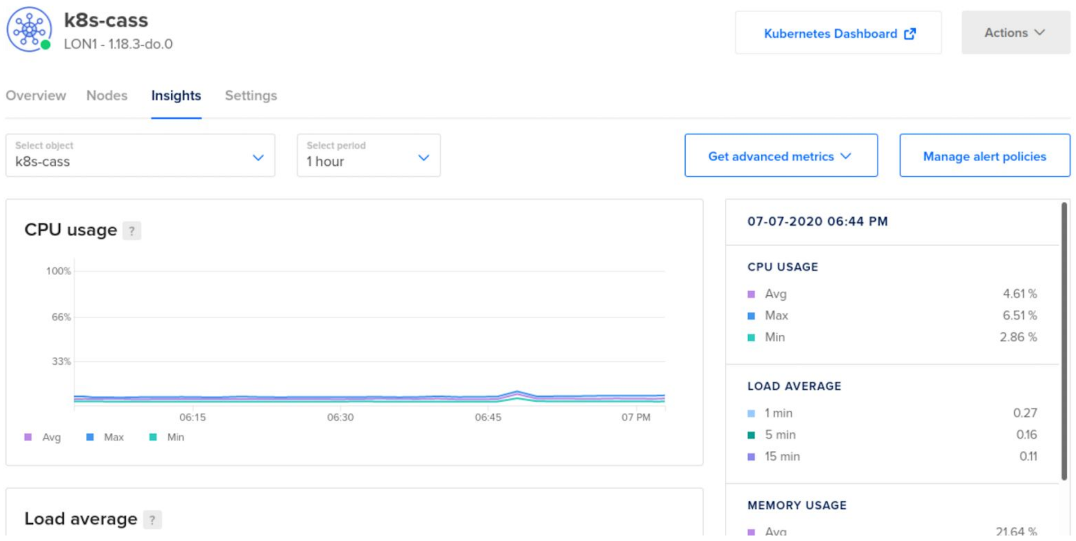
ACCESS CLUSTER CONFIG FILE

Certificates expire after 7 days. How do I update my certificate?

CLUSTER TAGS

k8s k8s:3a3b1f1f-1ac9-48b5-8795-dad8628a6883

26. Click on **Insights**



k8s-cass
LON1 - 1.18.3-do.0

Kubernetes Dashboard | Actions

Overview | Nodes | **Insights** | Settings

Select object: k8s-cass | Select period: 1 hour | Get advanced metrics | Manage alert policies

CPU usage

07-07-2020 06:44 PM

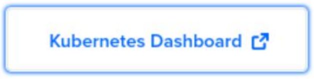
CPU USAGE	
Avg	4.61 %
Max	6.51 %
Min	2.86 %

LOAD AVERAGE	
1 min	0.27
5 min	0.16
15 min	0.11

MEMORY USAGE	
Avg	21.64 %

Load average

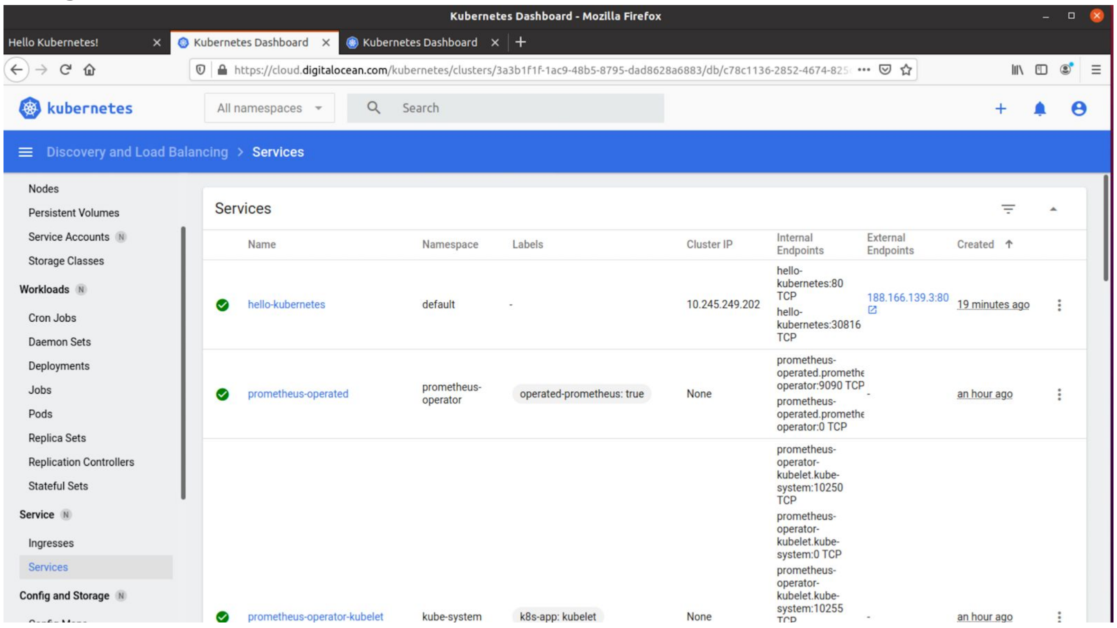
27. You can see the system monitoring.



Kubernetes Dashboard

28. Click on

29. Navigate to look at Services



Kubernetes Dashboard - Mozilla Firefox

Hello Kubernetes! | Kubernetes Dashboard | Kubernetes Dashboard

https://cloud.digitalocean.com/kubernetes/clusters/3a3b1f1f-1ac9-48b5-8795-dad8628a6883/db/c78c1136-2852-4674-825

kubernetes | All namespaces | Search

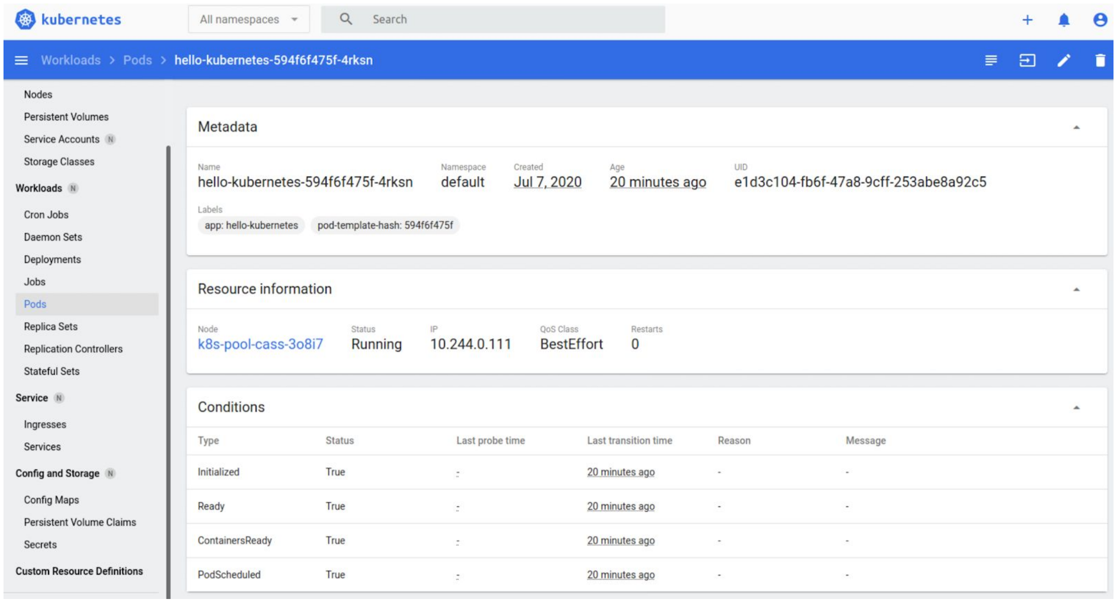
Discovery and Load Balancing > Services

Services

Name	Namespace	Labels	Cluster IP	Internal Endpoints	External Endpoints	Created
hello-kubernetes	default	-	10.245.249.202	hello-kubernetes:80 TCP hello-kubernetes:30816 TCP	188.166.139.3:80	19 minutes ago
prometheus-operated	prometheus-operator	operated-prometheus: true	None	prometheus-operated.promethe operator:9090 TCP prometheus-operated.promethe operator:0 TCP	-	an hour ago
prometheus-operator-kubelet	kube-system	k8s-app: kubelet	None	prometheus-operator-kubelet.kube-system:10250 TCP prometheus-operator-kubelet.kube-system:0 TCP prometheus-operator-kubelet.kube-system:10255	-	an hour ago

30. You can see a nice link to the external webpage of your app.

31. Browse pods and go look at a pod:



Metadata

Name	Namespace	Created	Age	UID
hello-kubernetes-594f6f475f-4rksn	default	Jul 7, 2020	20 minutes ago	e1d3c104-fb6f-47a8-9cff-253abe8a92c5

Labels
app: hello-kubernetes
pod-template-hash: 594f6f475f

Resource information

Node	Status	IP	QoS Class	Restarts
k8s-pool-cass-3o8i7	Running	10.244.0.111	BestEffort	0

Conditions

Type	Status	Last probe time	Last transition time	Reason	Message
Initialized	True	-	20 minutes ago	-	-
Ready	True	-	20 minutes ago	-	-
ContainersReady	True	-	20 minutes ago	-	-
PodScheduled	True	-	20 minutes ago	-	-

32. If you are a command-line person instead, let’s try a more CLI-ish approach:

```
sudo snap install k9s
```

```
Warning: /snap/bin was not found in your $PATH. If you've not
restarted your session since you
         installed snapd, try doing that. Please see
https://forum.snapcraft.io/t/9469 for
         more details.

k9s 0.7.12 from Fernand Galiana (derailed) installed
```

Ignore the warning.

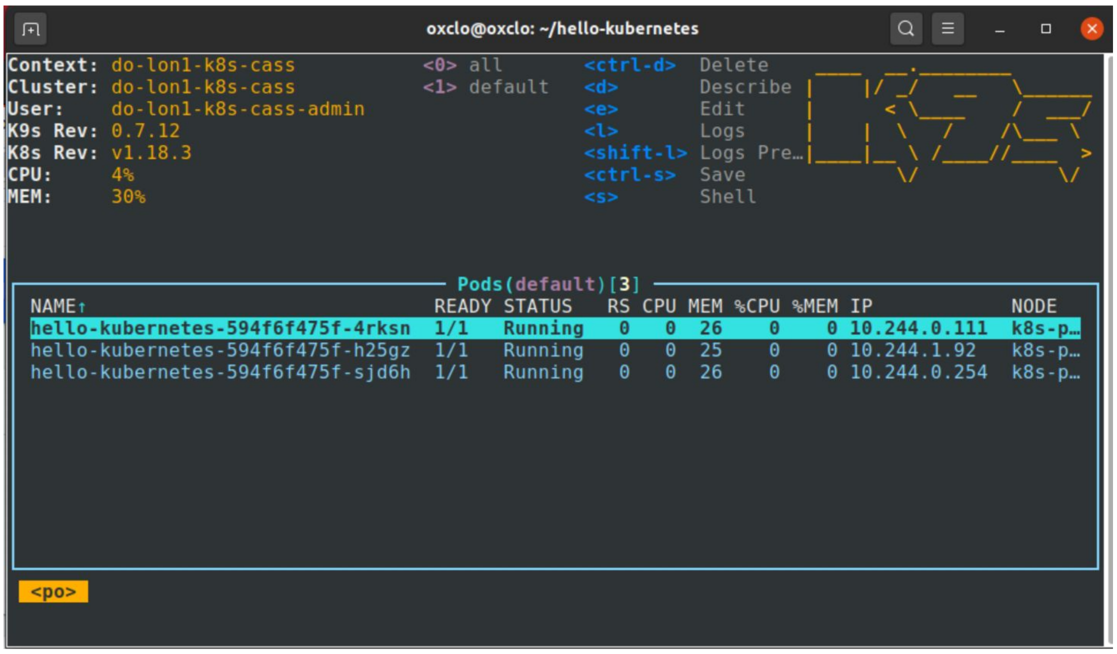
33. For some obscure reason we need to create a directory for the . k9s config file:

```
mkdir ~/.k9s
```

34. Now start k9s:

```
k9s
```

You should see:



```
oxclo@oxclo: ~/hello-kubernetes

Context:  do-lon1-k8s-cass          <0> all       <ctrl-d>  Delete
Cluster:  do-lon1-k8s-cass          <1> default   <d>       Describe
User:     do-lon1-k8s-cass-admin                  <e>       Edit
K9s Rev:  0.7.12                                  <l>       Logs
K8s Rev:  v1.18.3                                 <shift-l> Logs Pre…
CPU:      4%                                      <ctrl-s>  Save
MEM:      30%                                     <s>       Shell

                                Pods(default)[3]
  NAME↑                              READY STATUS   RS CPU MEM %CPU %MEM IP           NODE
  hello-kubernetes-594f6f475f-4rksn  1/1   Running  0   0  26    0    0 10.244.0.111 k8s-p…
  hello-kubernetes-594f6f475f-h25gz  1/1   Running  0   0  25    0    0 10.244.1.92  k8s-p…
  hello-kubernetes-594f6f475f-sjd6h  1/1   Running  0   0  26    0    0 10.244.0.254 k8s-p…

<po>
```

35. This is an awesome tool. Hit enter twice to see the pod logs. Have a look at the docs here: https://k9scli.io/

36. You can "drill" into pods and containers just by hitting Enter. You get back to the main screen with Esc. You can leave k9s by using Ctrl-C.

37. Do you remember that we installed the Kubernetes 1-click monitoring. Let's take a look at that.

38. We need to be able to access the pod containing Grafana:

    Find the pod name with:
    ```
    kubectl -n prometheus-operator get pods | grep \
    prometheus-operator-grafana
    ```

    You should see something like:
    ```
    prometheus-operator-grafana-cf6954699-xgklc          2/2     Running    0         113m
    ```

    Copy that name into this:

    ```
    kubectl port-forward prometheus-operator-grafana-cf6954699-xgklc  \
    -n prometheus-operator 8080:3000
    ```

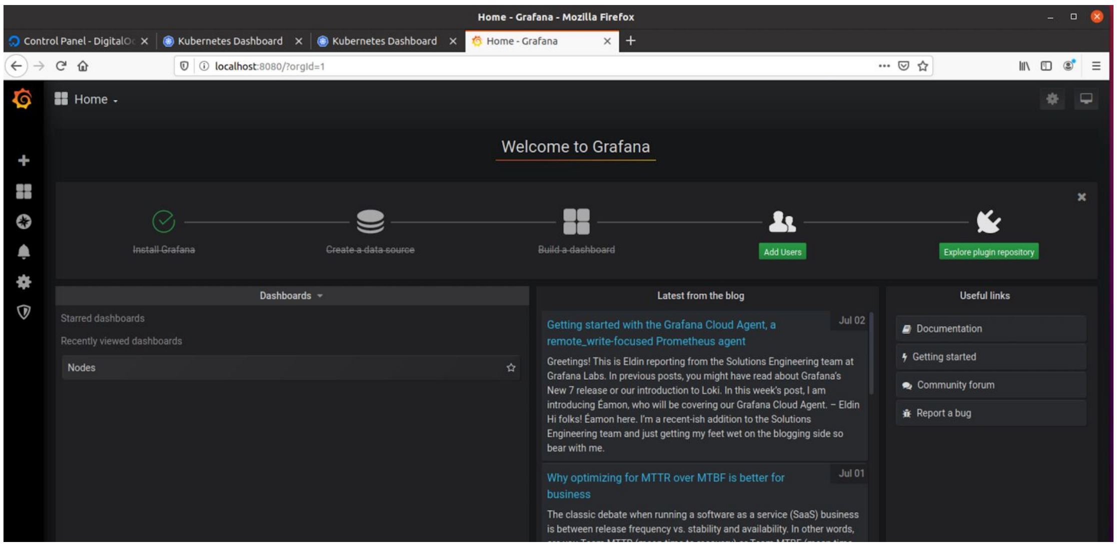
    Changing the name to match yours.

    You should see:
    ```
    Forwarding from 127.0.0.1:8080 -> 3000
    Forwarding from [::1]:8080 -> 3000
    ```

40. Now browse http://localhost:8080

    The username you need is **admin**
    And the password is **prom-operator**

    Obviously in a prod system you'd need to change these!



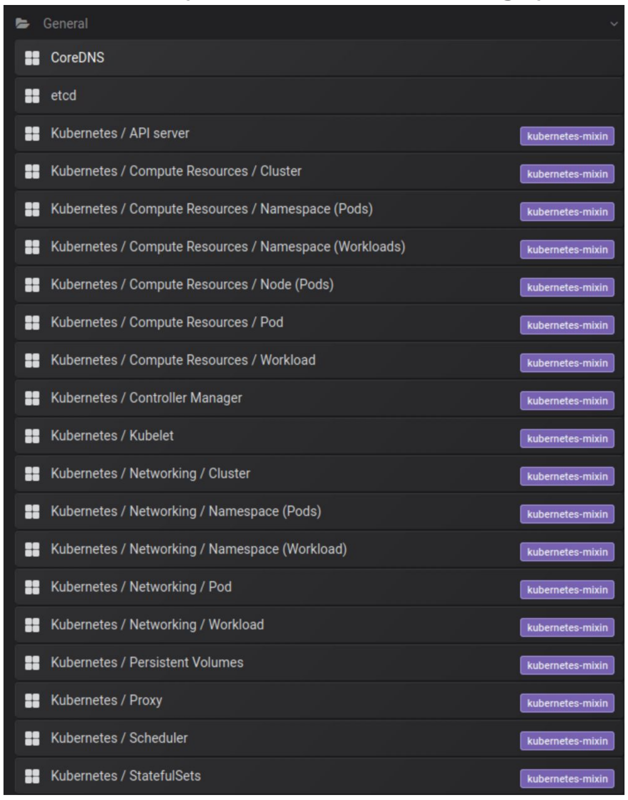
Getting started with the Grafana Cloud Agent, a remote_write-focused Prometheus agent — Jul 02

Greetings! This is Eldin reporting from the Solutions Engineering team at Grafana Labs. In previous posts, you might have read about Grafana's New 7 release or our introduction to Loki. In this week's post, I am introducing Éamon, who will be covering our Grafana Cloud Agent. – Eldin Hi folks! Éamon here. I'm a recent-ish addition to the Solutions Engineering team and just getting my feet wet on the blogging side so bear with me.

Why optimizing for MTTR over MTBF is better for business — Jul 01

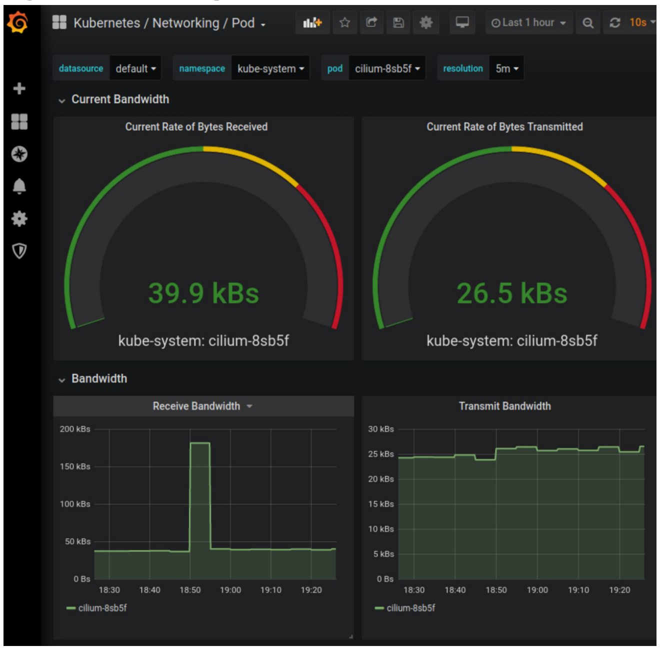
The classic debate when running a software as a service (SaaS) business is between release frequency vs. stability and availability. In other words,

41. Under **Home** you will see lots of things you can look at:



CoreDNS	
etcd	
Kubernetes / API server	kubernetes-mixin
Kubernetes / Compute Resources / Cluster	kubernetes-mixin
Kubernetes / Compute Resources / Namespace (Pods)	kubernetes-mixin
Kubernetes / Compute Resources / Namespace (Workloads)	kubernetes-mixin
Kubernetes / Compute Resources / Node (Pods)	kubernetes-mixin
Kubernetes / Compute Resources / Pod	kubernetes-mixin
Kubernetes / Compute Resources / Workload	kubernetes-mixin
Kubernetes / Controller Manager	kubernetes-mixin
Kubernetes / Kubelet	kubernetes-mixin
Kubernetes / Networking / Cluster	kubernetes-mixin
Kubernetes / Networking / Namespace (Pods)	kubernetes-mixin
Kubernetes / Networking / Namespace (Workload)	kubernetes-mixin
Kubernetes / Networking / Pod	kubernetes-mixin
Kubernetes / Networking / Workload	kubernetes-mixin
Kubernetes / Persistent Volumes	kubernetes-mixin
Kubernetes / Proxy	kubernetes-mixin
Kubernetes / Scheduler	kubernetes-mixin
Kubernetes / StatefulSets	kubernetes-mixin

42. e.g. **Networking / Pod**

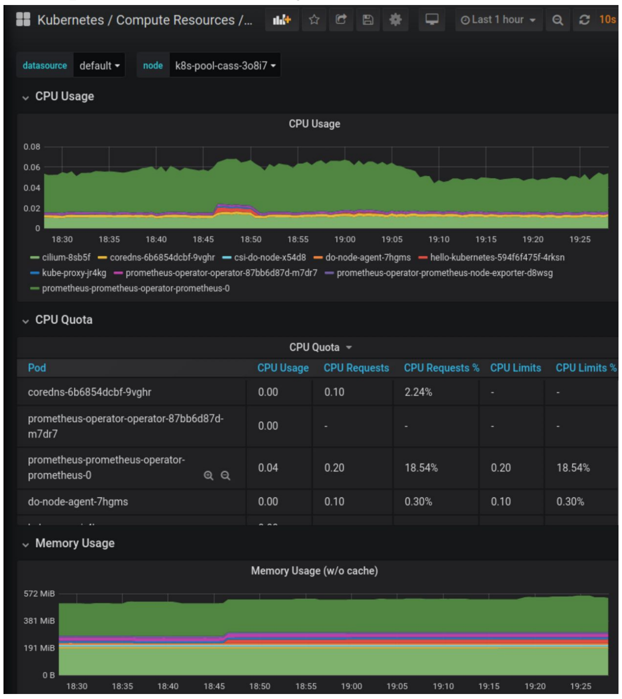


Kubernetes / Networking / Pod

datasource: default · namespace: kube-system · pod: cilium-8sb5f · resolution: 5m

Last 1 hour
10s
Current Bandwidth
Current Rate of Bytes Received
39.9 kBs
kube-system: cilium-8sb5f
Current Rate of Bytes Transmitted
26.5 kBs
kube-system: cilium-8sb5f
Bandwidth
Receive Bandwidth
Transmit Bandwidth
cilium-8sb5f

43. Compute resources by Node:



Kubernetes / Compute Resources /...

datasource: default · node: k8s-pool-cass-3o8i7

Last 1 hour
10s
CPU Usage
CPU Usage
cilium-8sb5f
coredns-6b6854dcbf-9vghr
csi-do-node-x54d8
do-node-agent-7hgms
hello-kubernetes-594f6f475f-4rksn
kube-proxy-jr4kg
prometheus-operator-operator-87bb6d87d-m7dr7
prometheus-operator-prometheus-node-exporter-d8wsg
prometheus-prometheus-operator-prometheus-0
CPU Quota
Memory Usage
Memory Usage (w/o cache)

CPU Quota

Pod	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits %
coredns-6b6854dcbf-9vghr	0.00	0.10	2.24%	-	-
prometheus-operator-operator-87bb6d87d-m7dr7	0.00	-	-	-	-
prometheus-prometheus-operator-prometheus-0	0.04	0.20	18.54%	0.20	18.54%
do-node-agent-7hgms	0.00	0.10	0.30%	0.10	0.30%

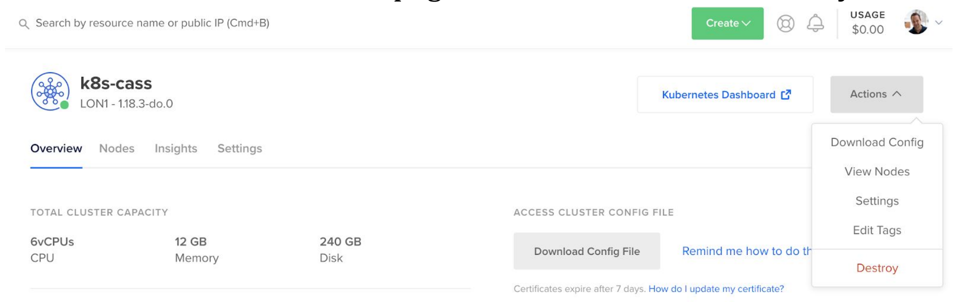
44. And lots more - have a good look.

45. That is the end of the lab. You have two choices now. Either you can delete the Kubernetes cluster (and stop spending that credit), or you can continue with the next exercise where we install cassandra into the cluster. If you want to delete the cluster, follow the next steps.

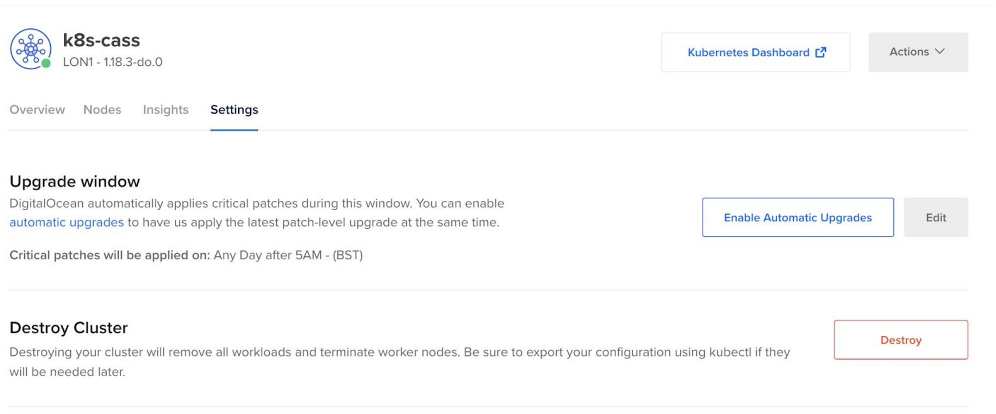
46. If you want to install Cassandra, go to **Exercise 14b**

DESTROY THE CLUSTER

47. Go to the kubernetes cluster page and find **Actions -> Destroy**



48. You will see:



Upgrade window

DigitalOcean automatically applies critical patches during this window. You can enable automatic upgrades to have us apply the latest patch-level upgrade at the same time.

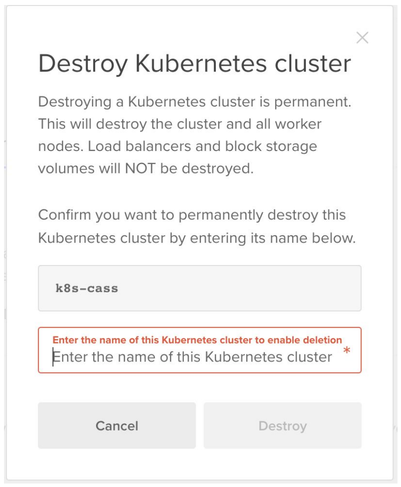
Critical patches will be applied on: Any Day after 5AM - (BST)

Destroy Cluster

Destroying your cluster will remove all workloads and terminate worker nodes. Be sure to export your configuration using kubectl if they will be needed later.

Click on **Destroy**

49. You will need to type the name of the cluster: **k8s-cass**



Destroy Kubernetes cluster

Destroying a Kubernetes cluster is permanent. This will destroy the cluster and all worker nodes. Load balancers and block storage volumes will NOT be destroyed.

Confirm you want to permanently destroy this Kubernetes cluster by entering its name below.

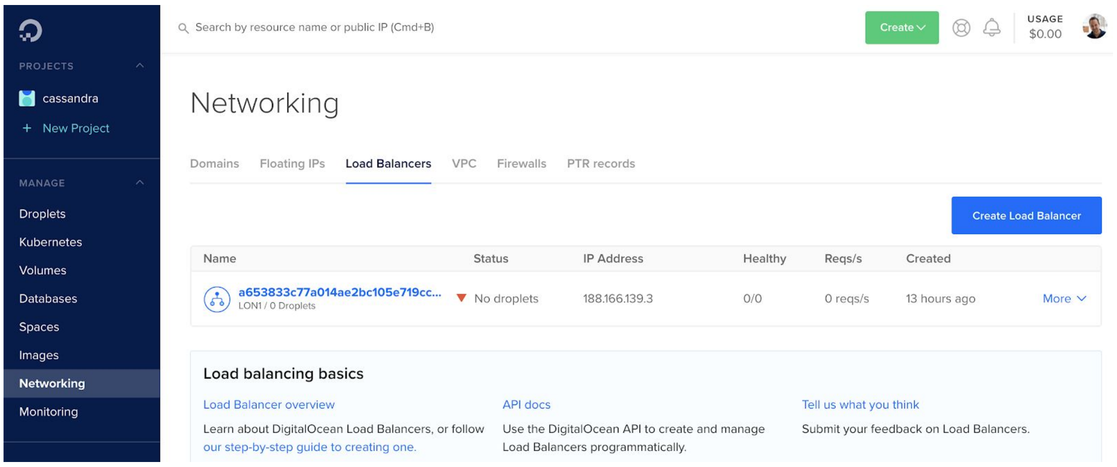
k8s-cass

Enter the name of this Kubernetes cluster to enable deletion

Enter the name of this Kubernetes cluster *

50. Then click **Destroy**

51. DigitalOcean will also have created a load-balancer to handle the incoming traffic for your service. Go to **Networking -> Load Balancers**



Networking

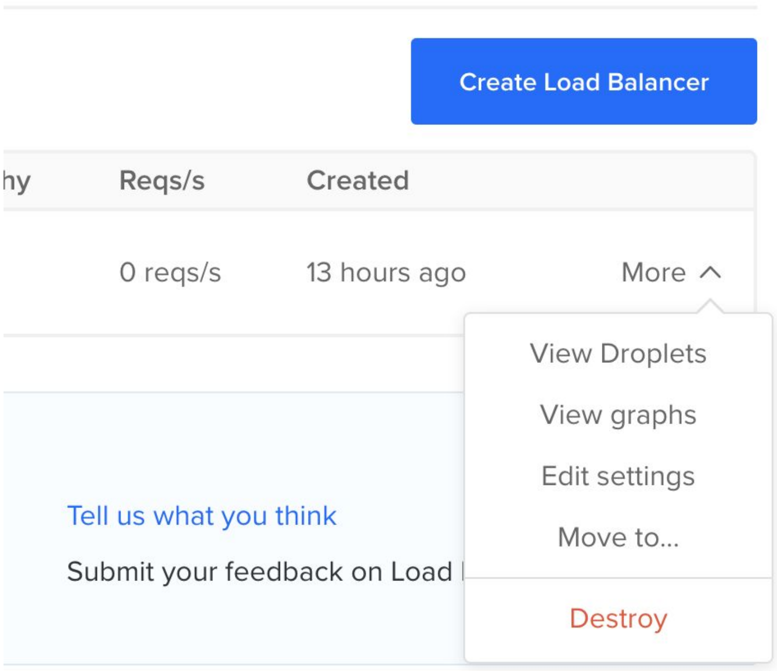
Name	Status	IP Address	Healthy	Reqs/s	Created	
a653833c77a014ae2bc105e719cc... LON1 / 0 Droplets	No droplets	188.166.139.3	0/0	0 reqs/s	13 hours ago	More

Load balancing basics

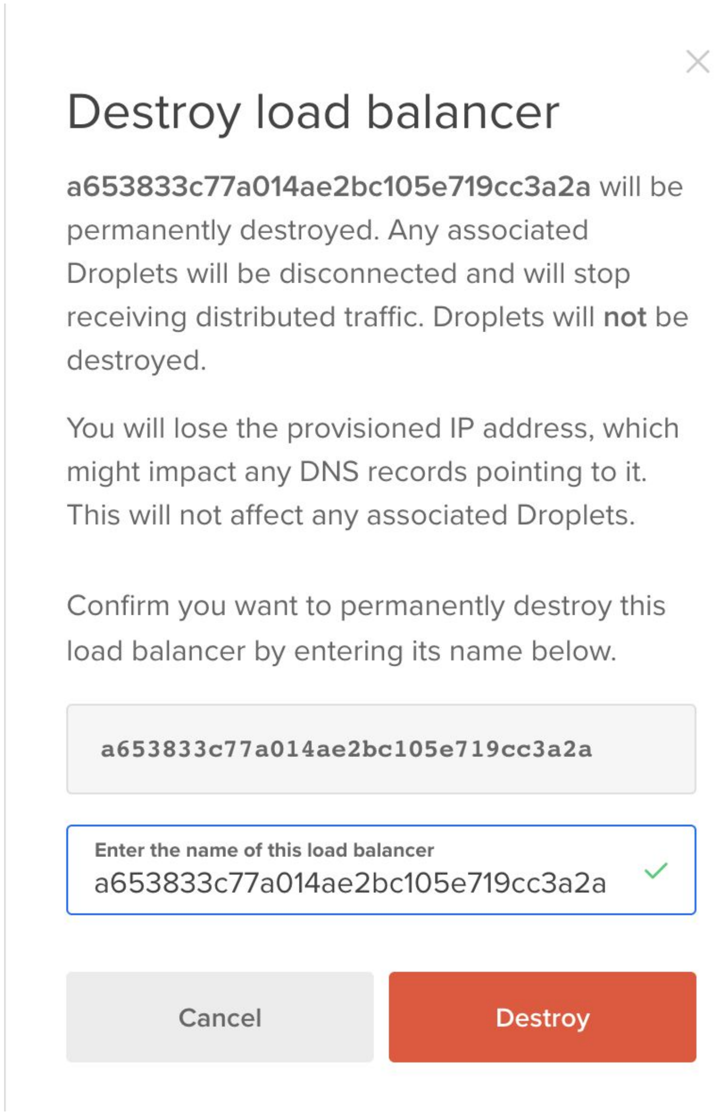
Load Balancer overview
Learn about DigitalOcean Load Balancers, or follow our step-by-step guide to creating one.

API docs
Use the DigitalOcean API to create and manage Load Balancers programmatically.

Tell us what you think
Submit your feedback on Load Balancers.



52. Click on **Destroy** and once again enter the name (copy and paste!)


Destroy load balancer

a653833c77a014ae2bc105e719cc3a2a will be permanently destroyed. Any associated Droplets will be disconnected and will stop receiving distributed traffic. Droplets will **not** be destroyed.

You will lose the provisioned IP address, which might impact any DNS records pointing to it. This will not affect any associated Droplets.

Confirm you want to permanently destroy this load balancer by entering its name below.

`a653833c77a014ae2bc105e719cc3a2a`

Enter the name of this load balancer
a653833c77a014ae2bc105e719cc3a2a ✓

53. This lab is done! Congratulations.