Week 6

Moving your app to k8s

Cassandra Cloud-Native Workshop Series

Building Cloud-Native apps with Cassandra Expertise

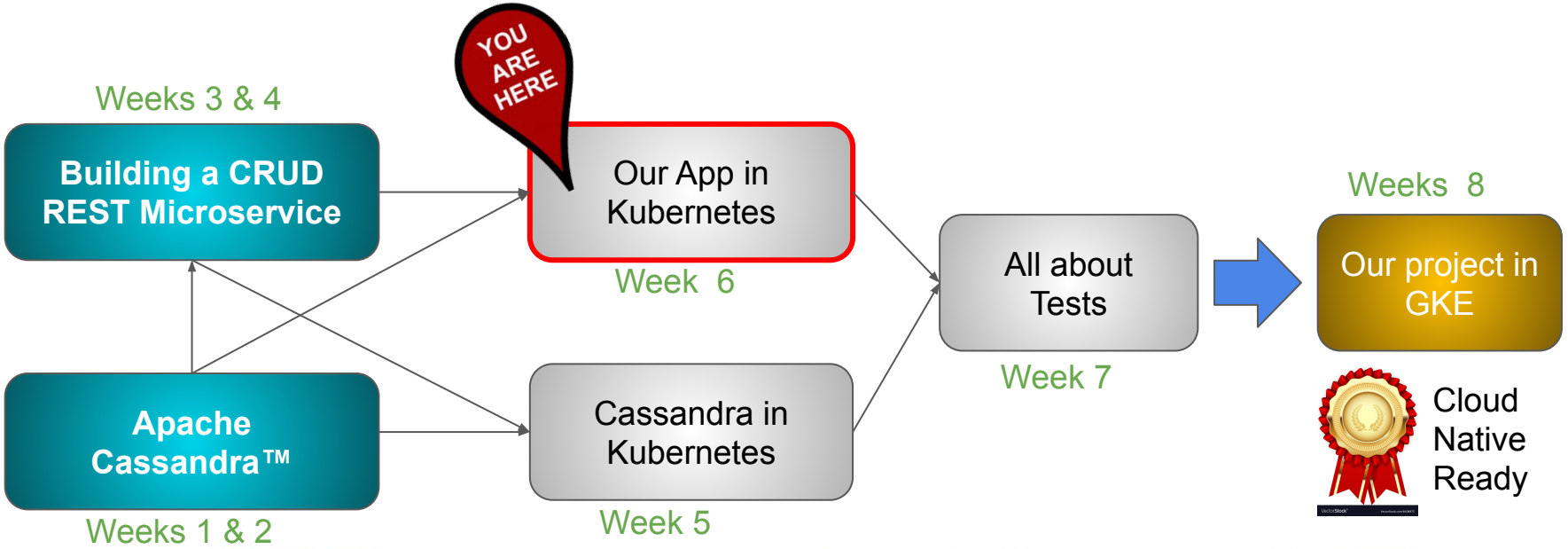
DataStax Developer Special Unit !

David Jones-Gilardi	Eric Zietlow	Erick Ramirez	Cédrick Lunven	Bettina Swynnerton	Jack Fryer	Aleksandr Volochnev

MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Workshops Series = Not only Cassandra



Developer Workshop Series **Week 6**

- ➔ Housekeeping
- ➔ Docker Review
- ➔ Kubernetes Review
- ➔ Setting Up Containers
- ➔ Getting the App Running In k8
- ➔ Resources

MATERIALS ➔ bit.ly/CassandraWorkshopMaterials

Livestreams

YouTube

Twitch

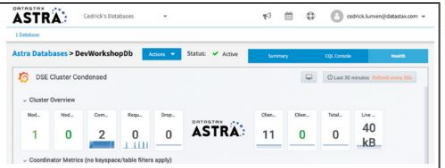
Live Questions

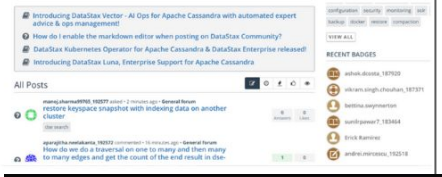
YouTube

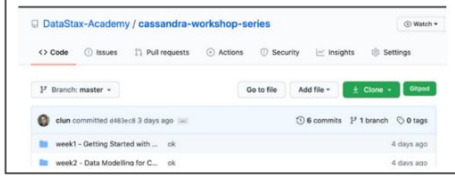
Discord

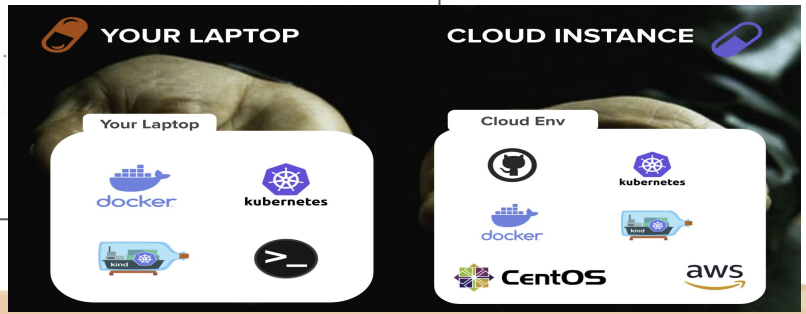
Materials & Help

DATASTAX COMMUNITY









Astra.datastax.com

cassandra

GitHub

MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra


menti.com
96 57 69
Mentimeter
What is the most desirable feature of your ideal workplace?
Flexible time
Social factor
Big paycheck
menti.com
Available on the iPhone
App Store
GET IT ON
Google play

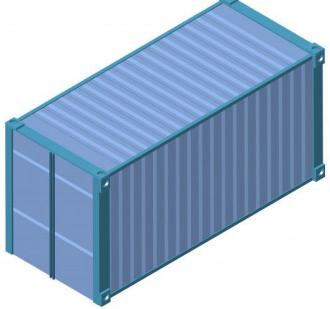
Developer Workshop Series **Week 6**

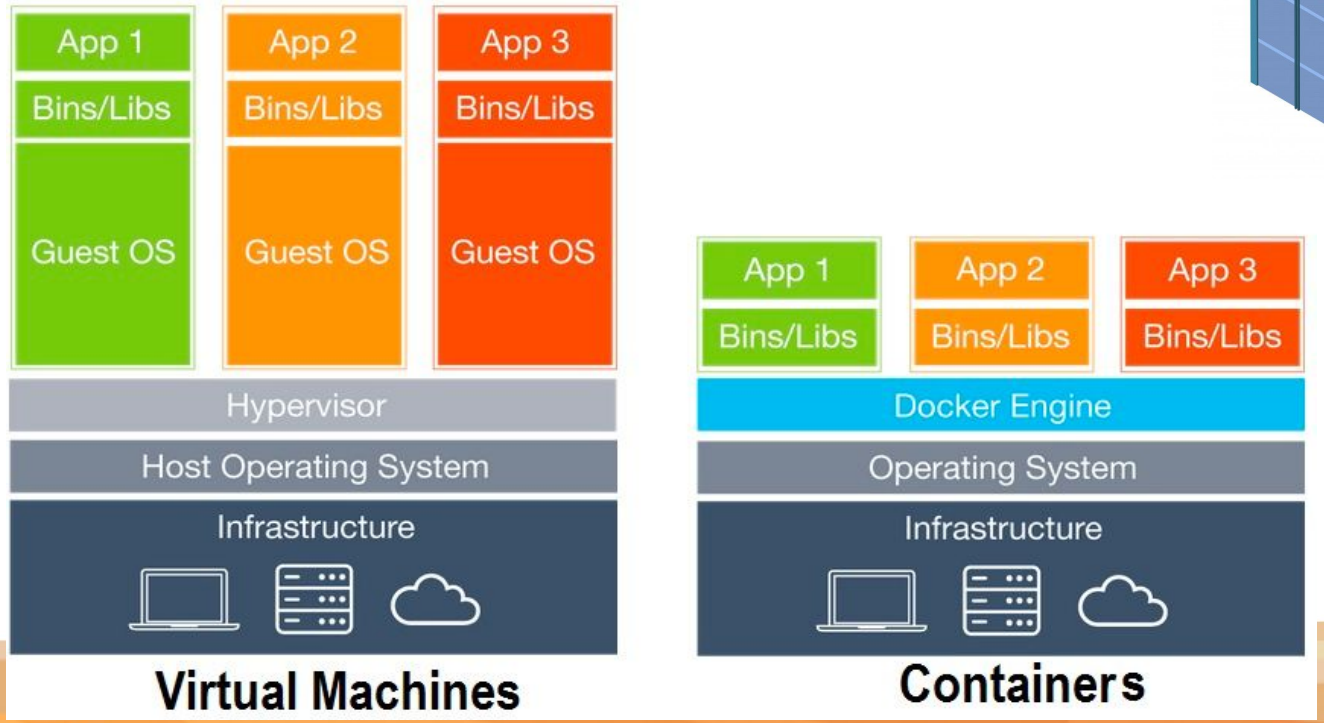
- ➔ Housekeeping
- ➔ Docker Review
- ➔ Kubernetes Review
- ➔ Setting Up Containers
- ➔ Getting the App Running In k8
- ➔ Resources

MATERIALS ➔ bit.ly/CassandraWorkshopMaterials

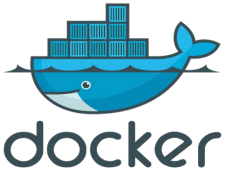
DATASTAX + Apache Cassandra

Back to Containers





Docker **BUILD** images and **RUN** containers



Image

- Image is like a **Class** or a **Template.**
- Image is a deliverable item, a package which you can **build**, **push** to a registry or **pull** from it.
- Image includes all dependencies: OS, libraries, code etc.
- We use Image in order to create a Container.

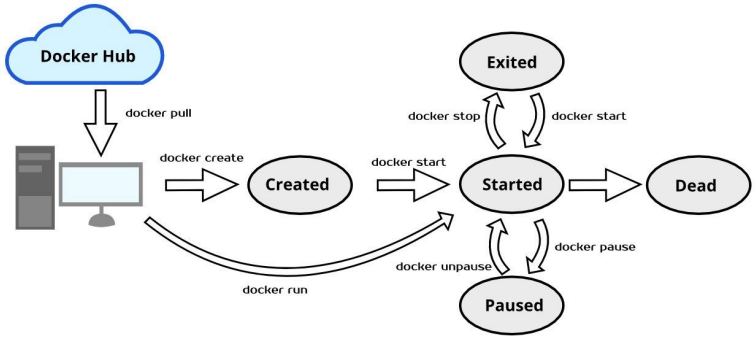
Container

- Container is like an “**Object** of that class” or “implemented Template”.
- We create a container based on an image.
- We run a container as a program, execute a command inside the container and stop it.

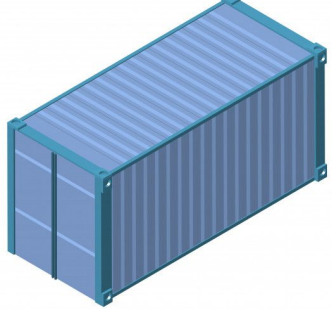
Docker main commands

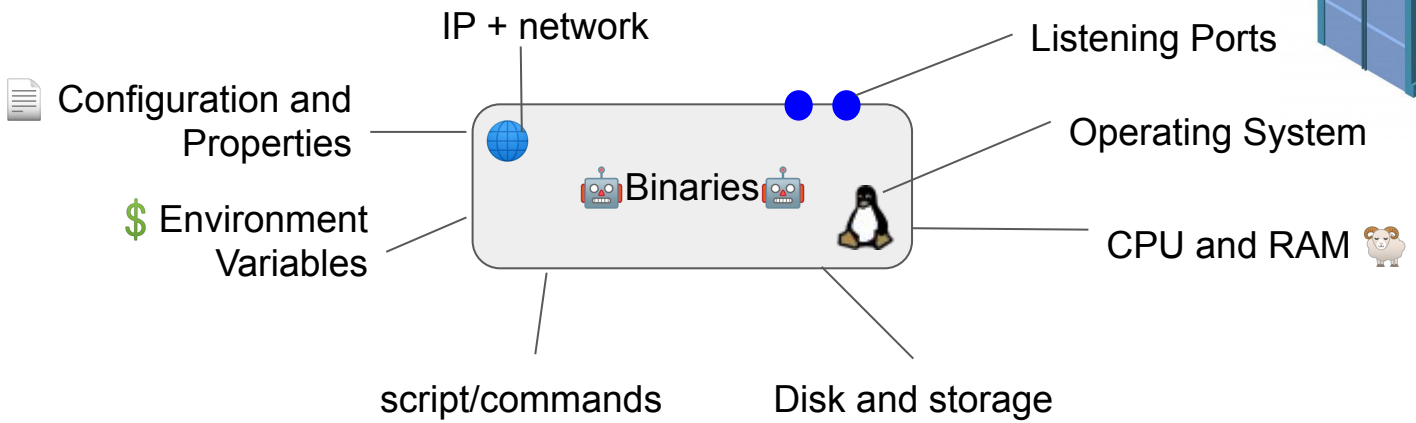
- **docker build** ← build an image
- **docker run** ← run a container from image
- **docker ps** ← watch the containers running
- **docker stop** ← stop the container
- **docker tag** ← tag the image
- **docker push** ← push the image to a repository
- **docker pull** ← pull the image to your computer

```
FROM python:3.7-alpine
WORKDIR /code
ENV FLASK_APP app.py
ENV FLASK_RUN_HOST 0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
EXPOSE 5000
COPY . .
CMD ["flask", "run"]
```

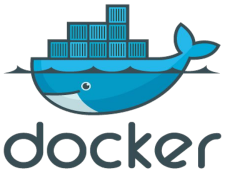


Your Applications in Containers





Cassandra in a Docker



Running Cassandra in Docker

- Define a proper network
- **Env variables** can be defined to override keys in `cassandra.yaml`
- Export ports **7000, 9042, …**
- Define volumes to stores data
  - **/var/lib/cassandra**

```
$ docker run

  --name some-cassandra -d \

  -e CASSANDRA_BROADCAST_ADDRESS=10.42.42.42 \

  -p 7000:7000,9042:9042

  -v /my/own/datadir:/var/lib/cassandra \

  cassandra:tag
```

Developer Workshop Series **Week 6**

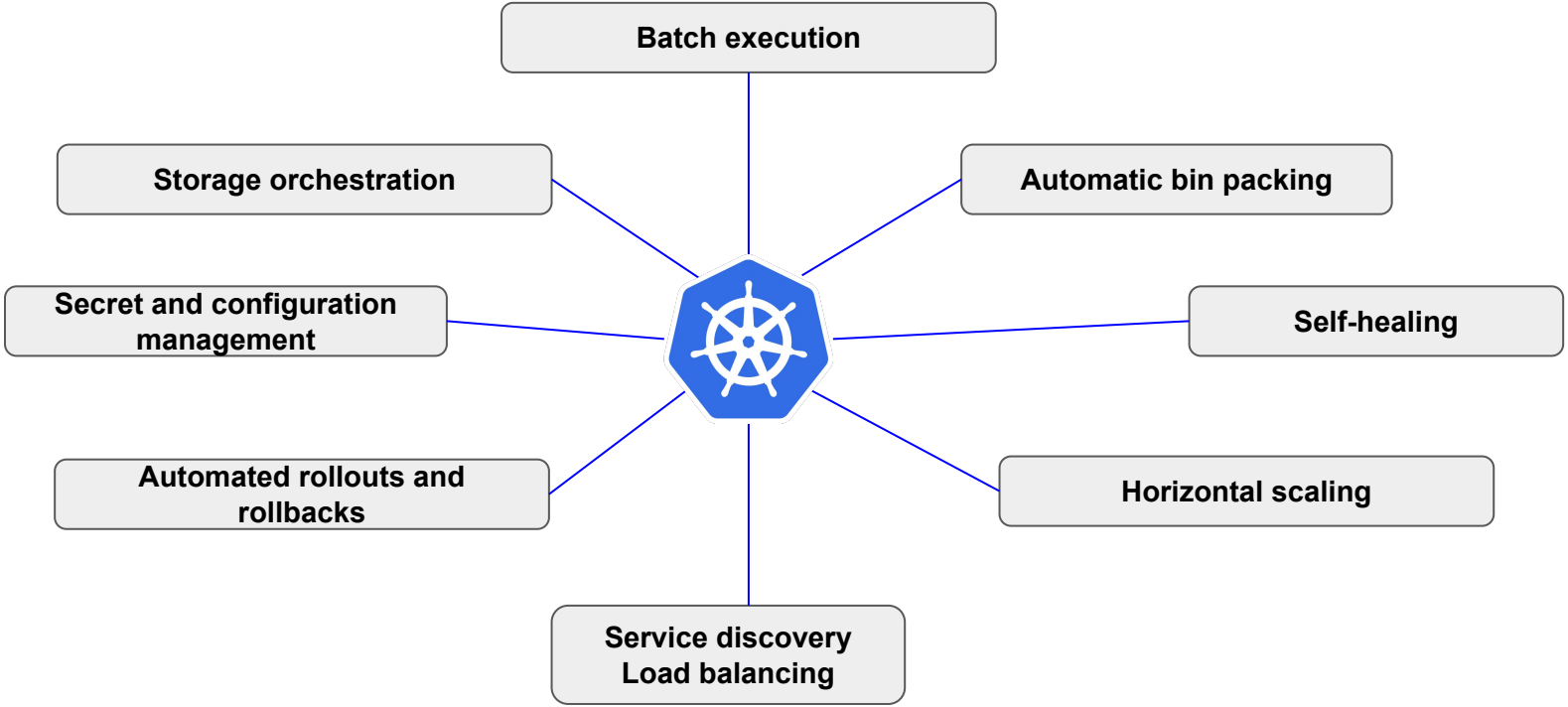
- ➔ Housekeeping
- ➔ Docker Review
- ➔ Kubernetes Review
- ➔ Setting Up Containers
- ➔ Getting the App Running In k8
- ➔ Resources

MATERIALS ➔ bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

kubernetes

"Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications."

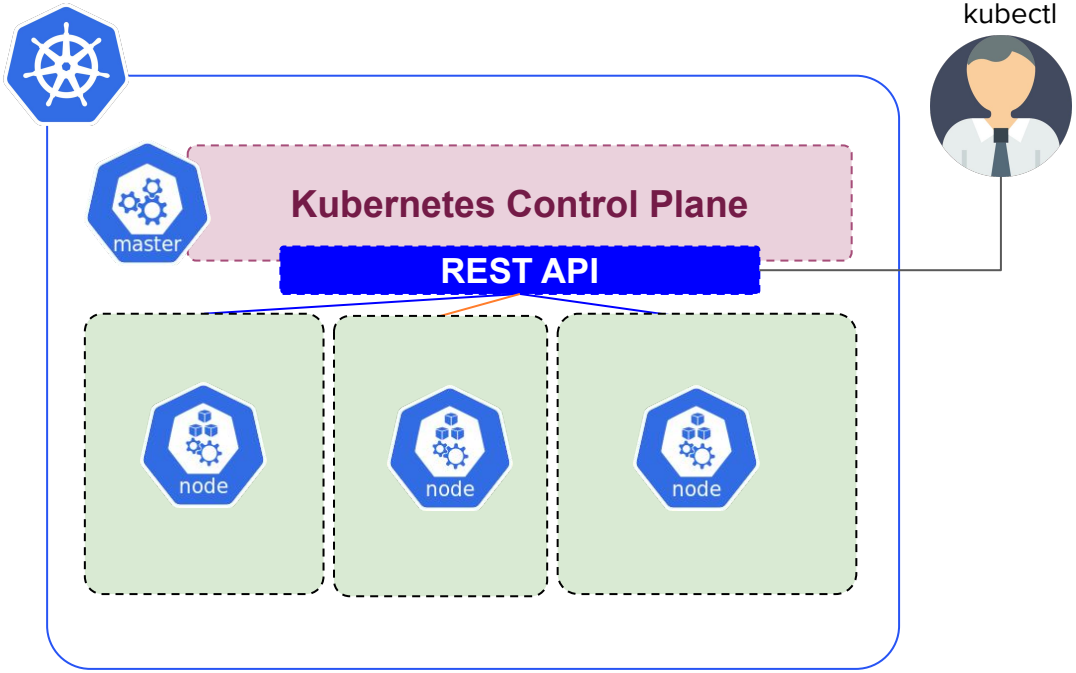

Batch execution
Storage orchestration
Automatic bin packing
Secret and configuration management
Self-healing
Automated rollouts and rollbacks
Horizontal scaling
Service discovery
Load balancing

K8s Infrastructure

Cluster:
Kubernetes cluster.

Master:
Kubernetes Control Plane.

Node:
Worker machine in Kubernetes cluster.

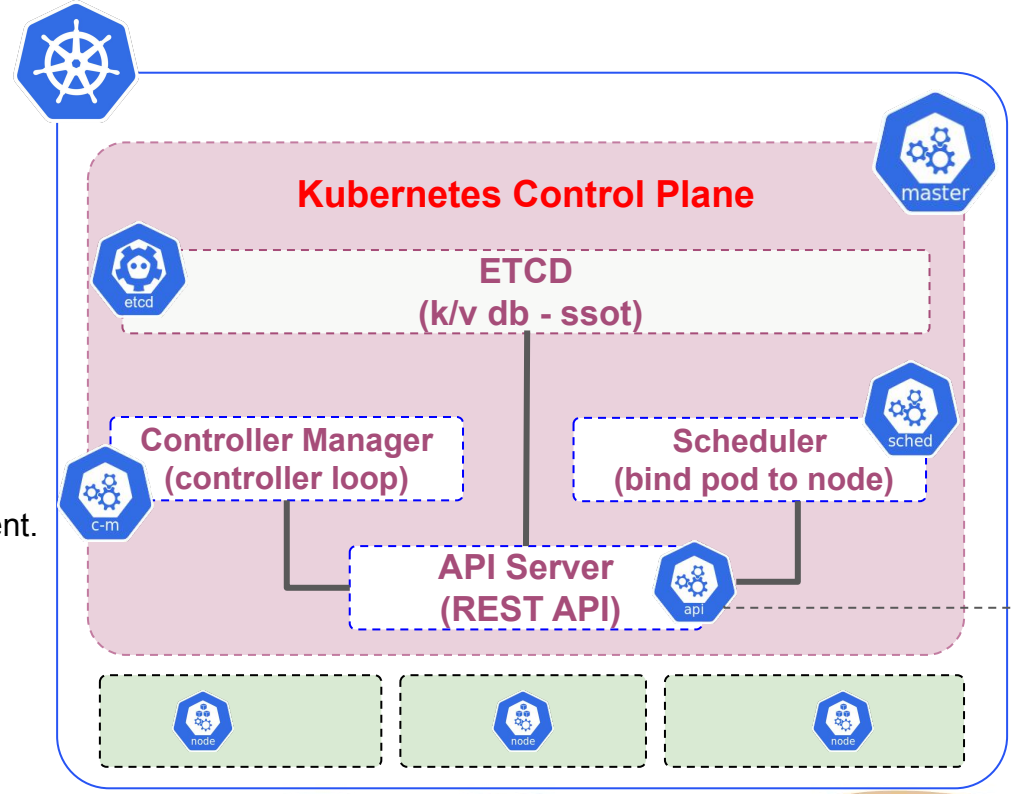


K8s API Server
Kubernetes API.

Controller Manager
Kubernetes controller manager.

Scheduler
In charge of ensuring Pods placement.

ETCD
Kubernetes' backing store.



MATERIALS → bit.ly/CassandraWorkshopMaterials

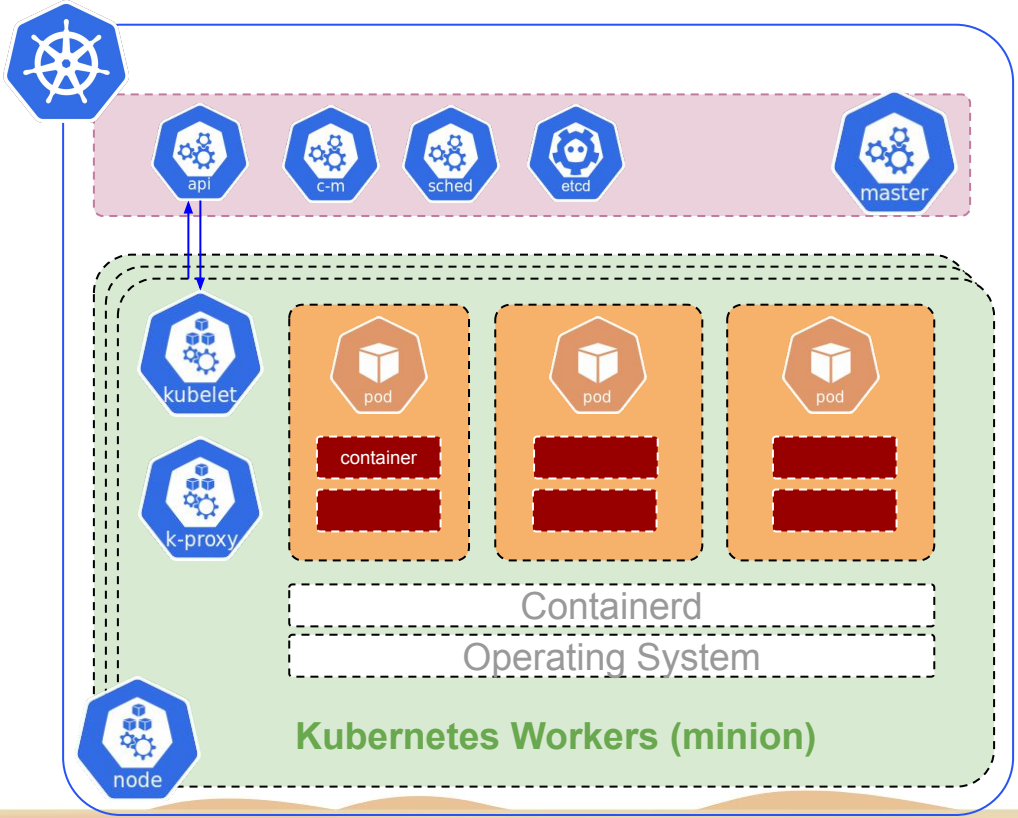
DATASTAX + Apache Cassandra

Kubelet:
The kubelet is the primary “node agent” that runs on each node.

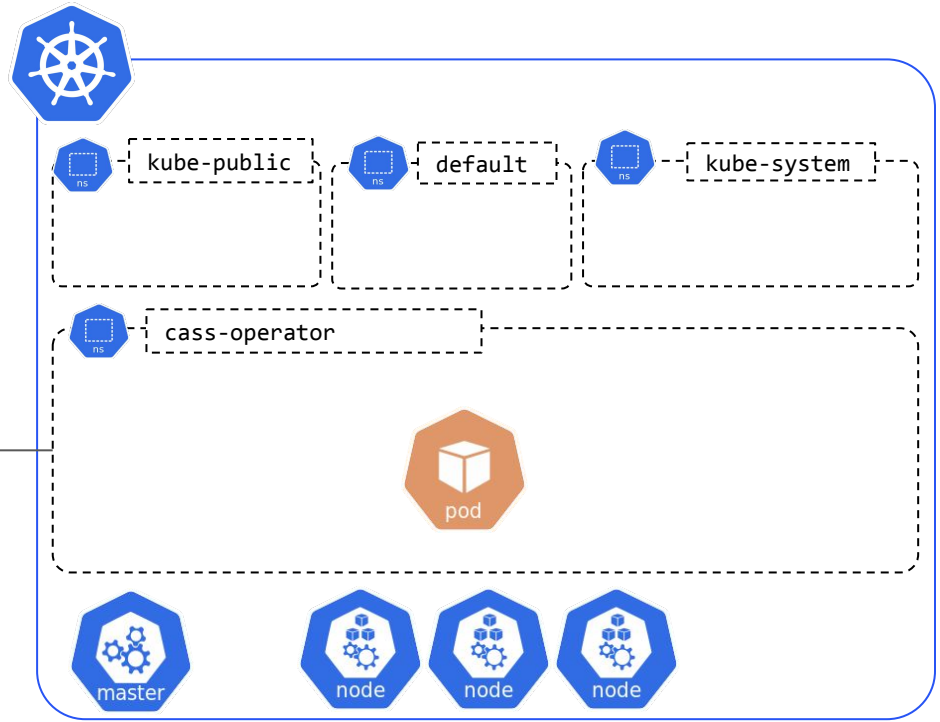
Kube-proxy
The Kubernetes network proxy runs on each node. This reflects services as defined in the Kubernetes API on each node.

POD
Collection of containers that can run on a host.

This resource is created by clients and scheduled onto hosts.



Namespace: Namespace provides a scope for Names. Use of multiple namespaces is optional.



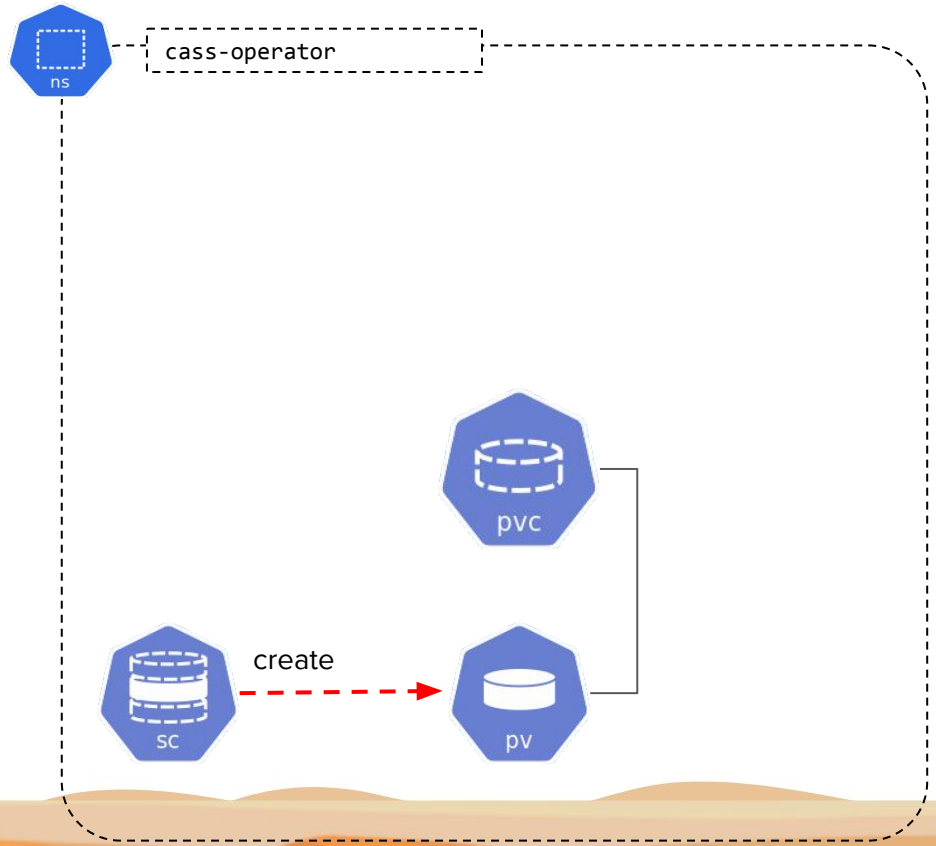
We create resources in namespaces span across node.

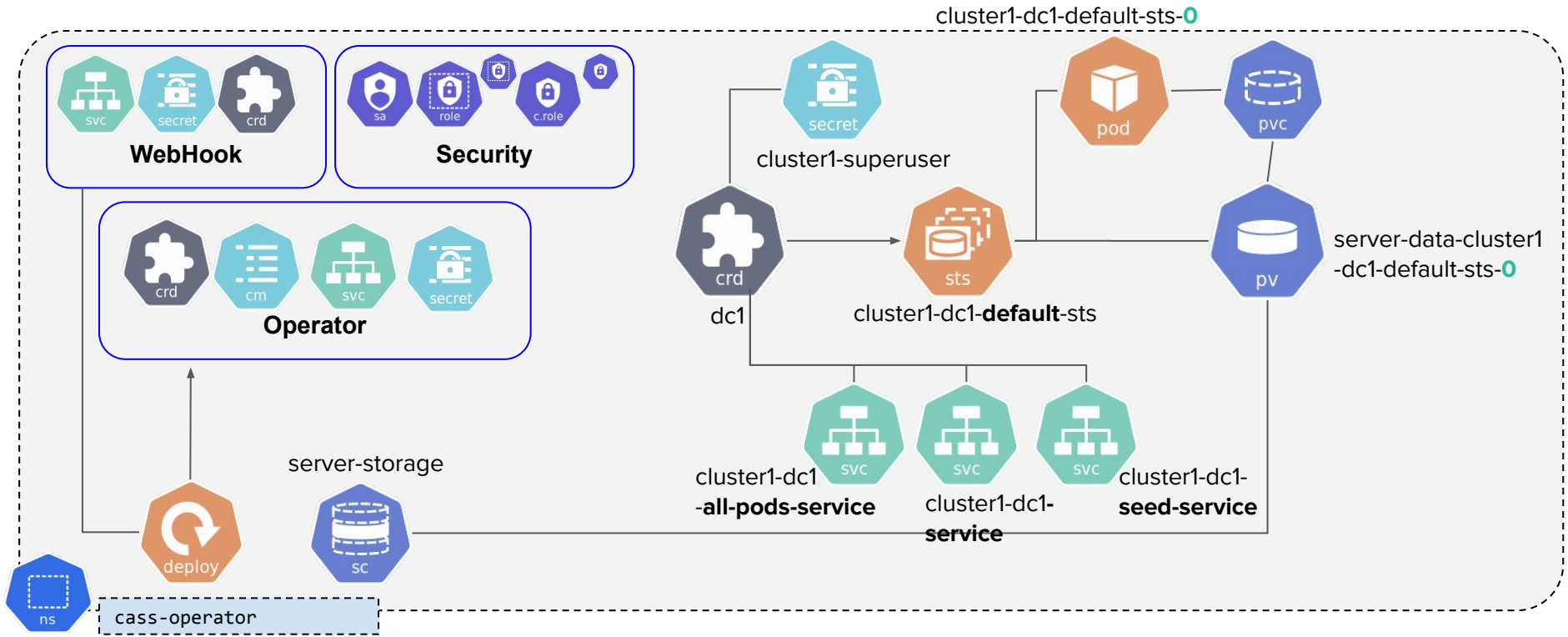
K8s Primitives : Storage

PersistentVolume: is a storage resource provisioned by an administrator.

PersistentVolumeClaim: PersistentVolumeClaim is a user's request for and claim to a persistent volume.

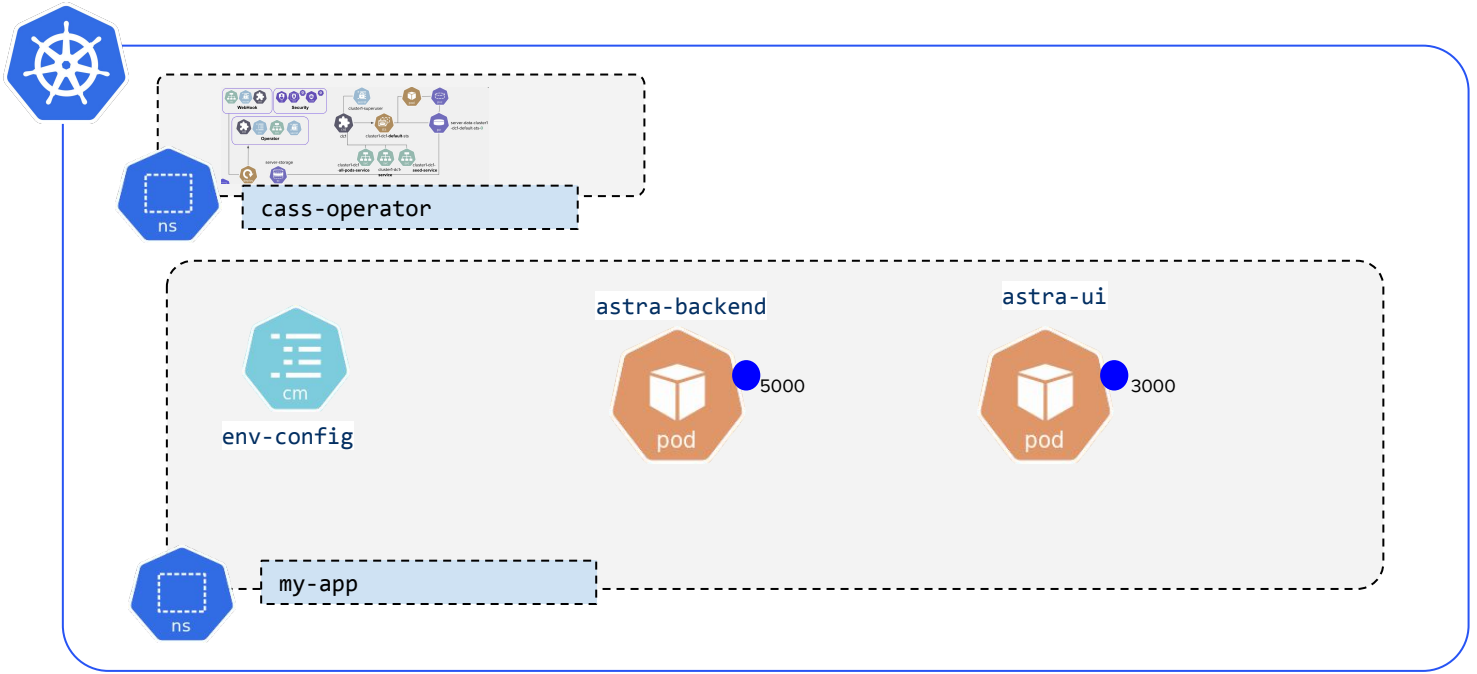
StorageClass: StorageClass describes the parameters for a class of storage for which *PersistentVolumes* can be dynamically provisioned.

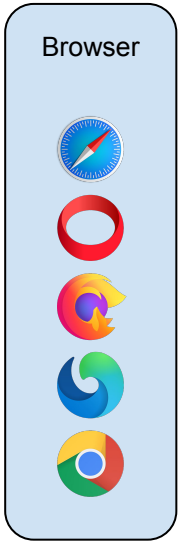




MATERIALS → bit.ly/CassandraWorkshopMaterials

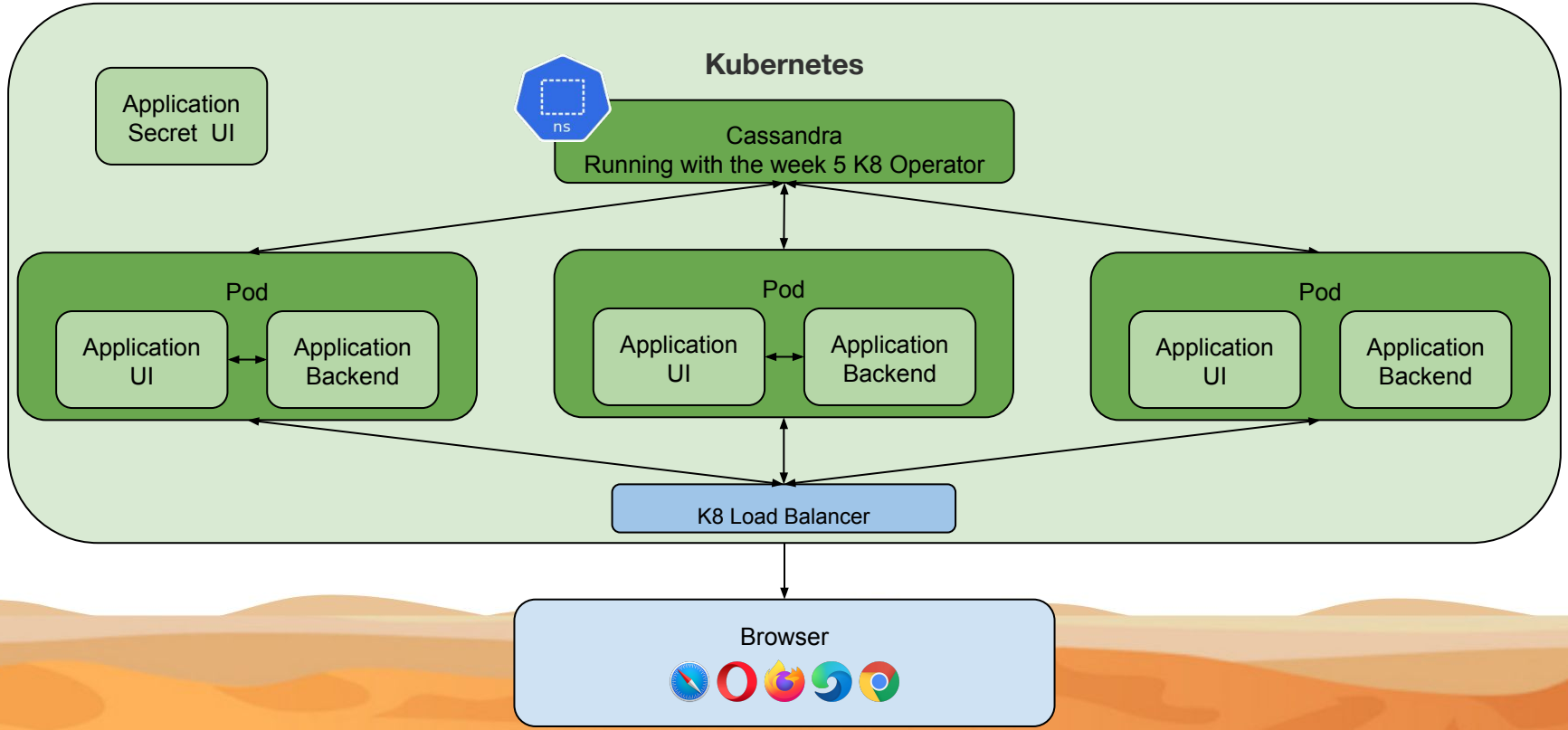
DATASTAX + Apache Cassandra



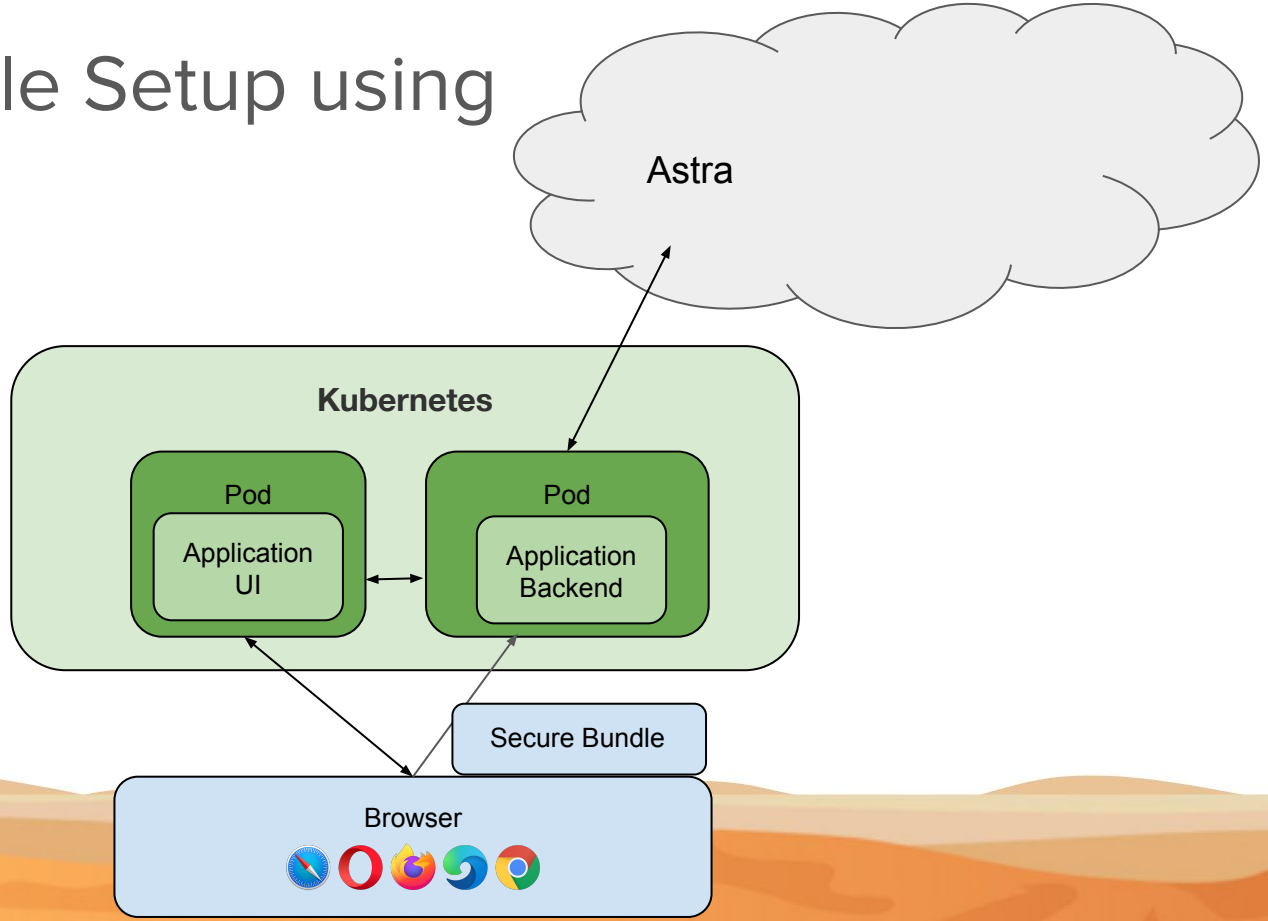


MATERIALS → bit.ly/CassandraWorkshopMaterials

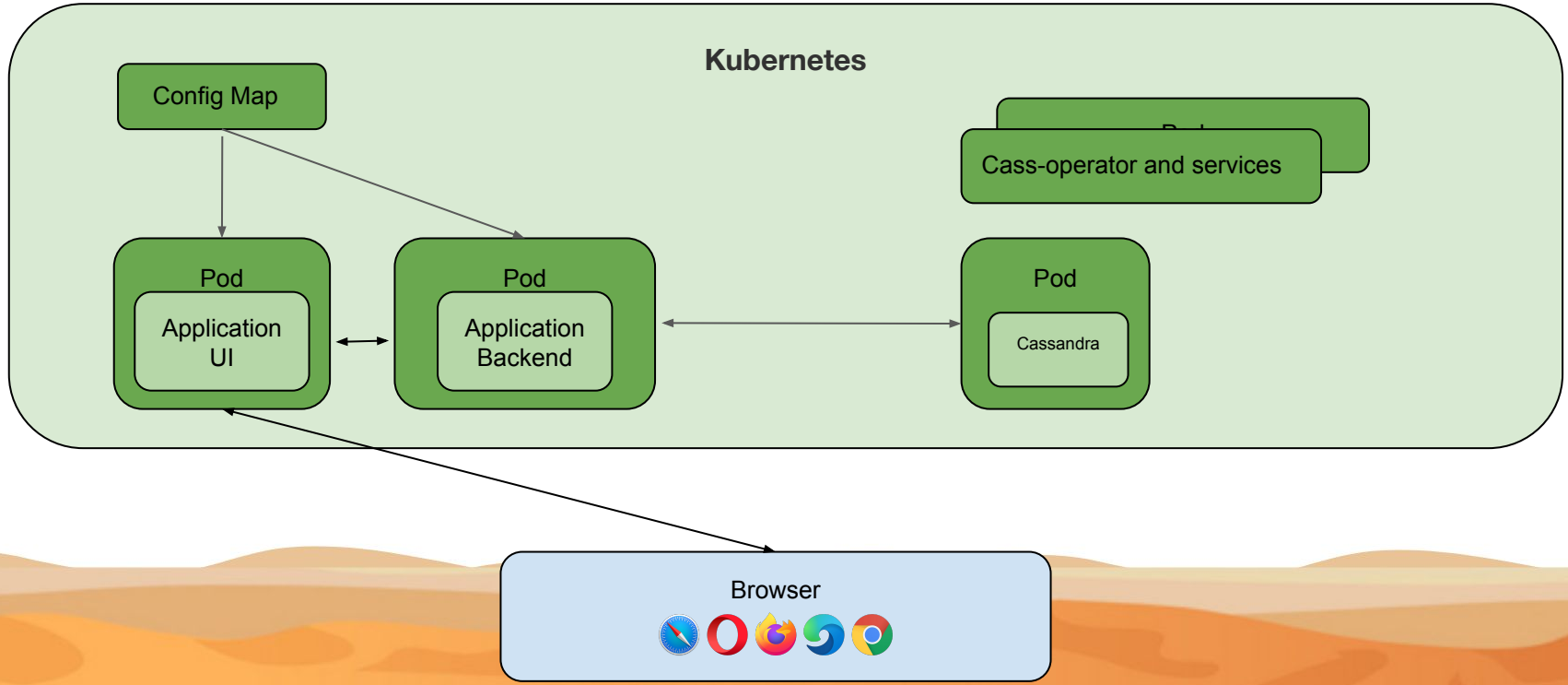
Your Full App on Kubernetes



Simple Example Setup using Astra



Example Setup with Cassandra in k8s



YOUR LAPTOP

CLOUD INSTANCE

Your Laptop

docker

kubernetes

kind

>_

Cloud Env

kubernetes

docker

kind

CentOS

aws

Developer Workshop Series **Week 6**

- ➔ Housekeeping
- ➔ Docker Review
- ➔ Kubernetes Review
- ➔ Setting Up Containers
- ➔ Getting the App Running In k8
- ➔ Resources

MATERIALS ➔ bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Running Your Application in its Current State

Launch the UI

```
$ echo “BASE_ADDRESS=http://localhost:5000/api” > .env
$ npm install
$ npm run start
```

Launch the Backend

Python

```
$ git clone
https://github.com/DataStax-Examples
/getting-started-with-astra-python.git
$ cd getting-started-with-astra-python
$ pip install Flask flask-cors
cassandra-driver
$ python getting_started_with_astra.py
```

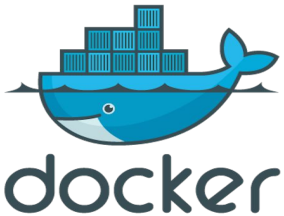
Java

```
$ git clone
https://github.com/DataStax-Examples/getti
ng-started-with-astra-java.git
$ cd getting-started-with-astra-java
$ mvn spring-boot:run
```

Can we simplify all this?

Can we simplify all this?

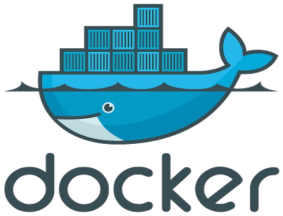
YES!!!



+

=

Less Operations Work



Dockerfiles

Python

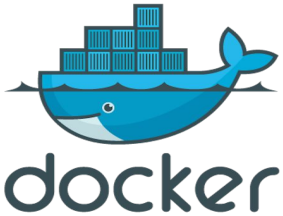
```
FROM python:alpine3.7
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
EXPOSE 5000
CMD ["python", "getting_started_with_astra.py"]
```

Java

```
FROM openjdk:11
MAINTAINER Cedrick Lunven
<cedrick.lunven@datastax.com>
VOLUME /tmp
ARG JAR_FILE
ADD ${JAR_FILE} app.jar
ENTRYPOINT
["java","-Djava.security.egd=file:/dev/./urando
m","-jar","/app.jar"]
EXPOSE 5000
```

React UI

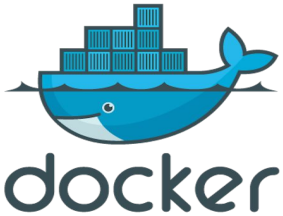
```
FROM node:12
COPY . /app
WORKDIR /app
RUN npm install
EXPOSE 3000
CMD [ "npm", "run", "start" ]
```



Dockerfiles

Python Dockerfile

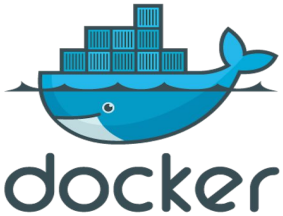
```
FROM python:alpine3.7
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
EXPOSE 5000
CMD ["python", "getting_started_with_astra.py"]
```



Dockerfiles

Setup environment template →

```
FROM python:alpine3.7
```



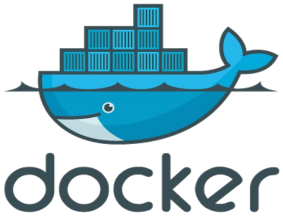
Dockerfiles

Setup environment template →

```
FROM python:alpine3.7
```

Setup application structure →

```
COPY . /app
WORKDIR /app
```



Dockerfiles

Setup environment template

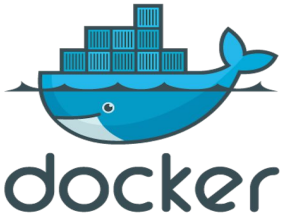
```
FROM python:alpine3.7
```

Setup application structure

```
COPY . /app
WORKDIR /app
```

Install dependencies

```
RUN pip install -r requirements.txt
```



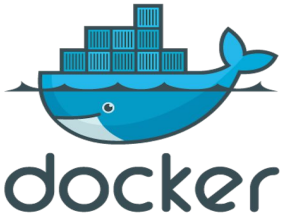
Dockerfiles

Setup environment template → `FROM python:alpine3.7`

Setup application structure →
```
COPY . /app
WORKDIR /app
```

Install dependencies → `RUN pip install -r requirements.txt`

Configure ports → `EXPOSE 5000`



Dockerfiles

Setup environment template →

```
FROM python:alpine3.7
```

Setup application structure →

```
COPY . /app
WORKDIR /app
```

Install dependencies →

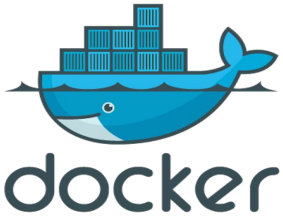
```
RUN pip install -r requirements.txt
```

Configure ports →

```
EXPOSE 5000
```

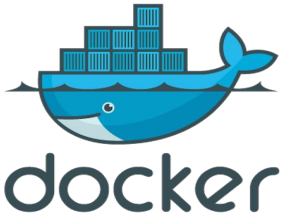
Run the application →

```
CMD ["python", "getting_started_with_astra.py"]
```



Build and Run

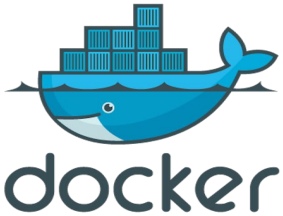
```
$ sudo docker build . --tag node-ui:my-image
$ sudo docker run --name node-ui node-ui:my-image
```



Build and Run

Build the docker image

```
sudo docker build . --tag node-ui:my-image
```



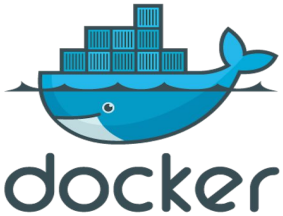
Build and Run

Build the docker image

```
sudo docker build . --tag node-ui:my-image
```

Start the container with the built image

```
sudo docker run --name node-ui node-ui:my-image
```



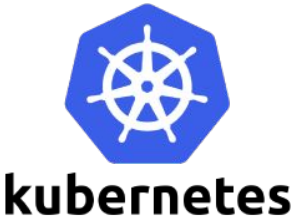
Success!!!

Developer Workshop Series **Week 6**

- Housekeeping
- Docker Review
- Kubernetes Review
- Setting Up Containers
- Getting the App Running In k8
- Resources

MATERIALS → bit.ly/CassandraWorkshopMaterials

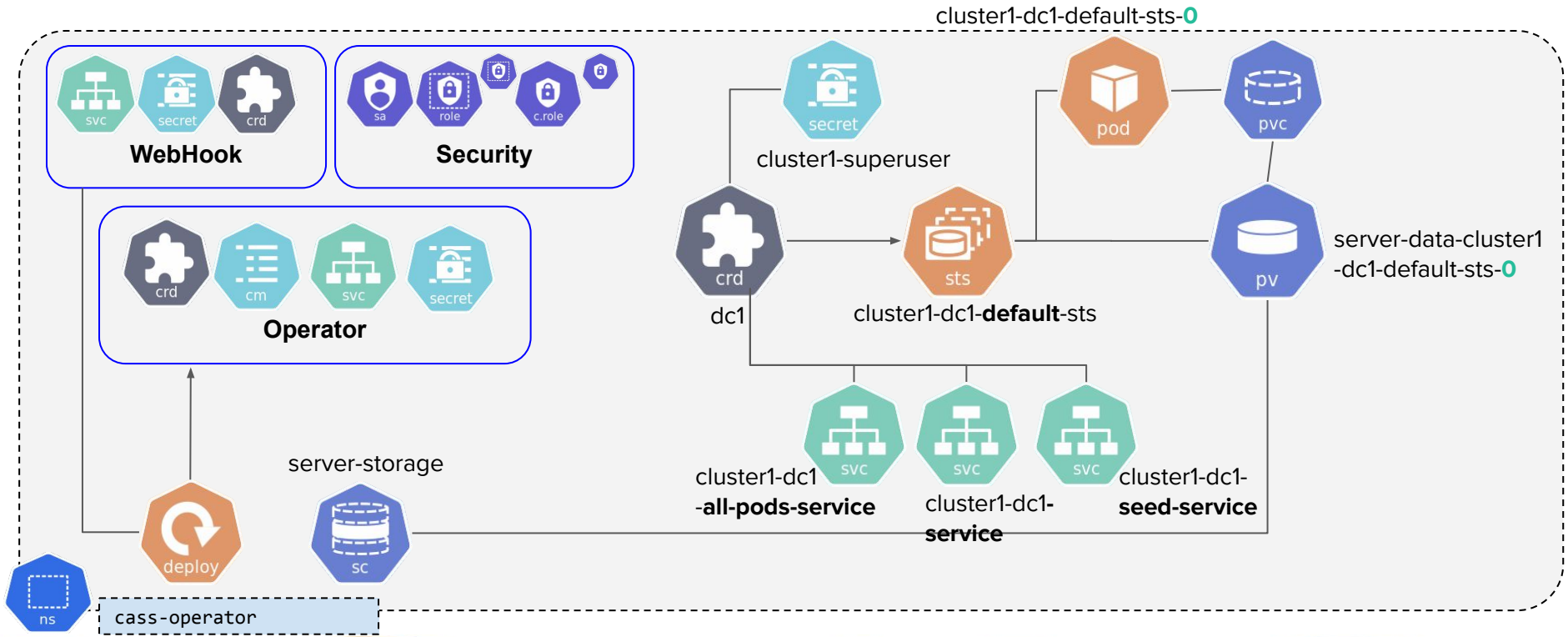
kubernetes



Getting cluster running from last week

Getting C* cluster running from last week

```
$ cd week6-App-in-k8s/kubefiles
$ kind create cluster --name kind-cassandra --config 01-kind-config.yaml
$ kubectl cluster-info --context kind-kind-cassandra
$ kubectl create ns cass-operator
$ kubectl -n cass-operator apply -f 02-storageclass-kind.yaml
$ kubectl -n cass-operator apply -f 03-install-cass-operator-v1.3.yaml
$ kubectl -n cass-operator apply -f 04-cassandra-cluster-1nodes.yaml
```



MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Building and loading docker your images

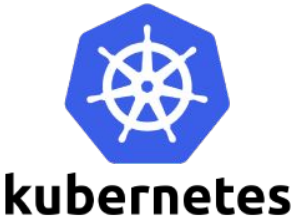
```
$ cd ../
$ docker build ./getting-started-with-astra-ui -t astra-ui:my-image
$ docker build ./getting-started-with-astra-python -t astra-backend:my-image
$ kind load docker-image astra-ui:my-image --name kind-cassandra
$ kind load docker-image astra-backend:my-image --name kind-cassandra
```

Getting your Cass-operator creds

```
$ kubectl get secret cluster1-superuser -n cass-operator -o yaml
```

Find the section for your credentials. It should look something like this

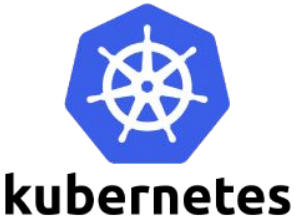
```
data:
 password: QzNkRmh0c1I2OHNJSllic19jSV9UejVURkM0WFZpblMtT19mMmpOOUxjMXg2bUhxSW9razJR
 username: Y2x1c3RlcjEtc3VwZXJ1c2Vy
```



Getting your Cass-operator creds

Replace the REPLACEME with your password string from the last step to get your password

```
$ echo REPLACEME | base64 -d && echo ""
```



Setup Schema

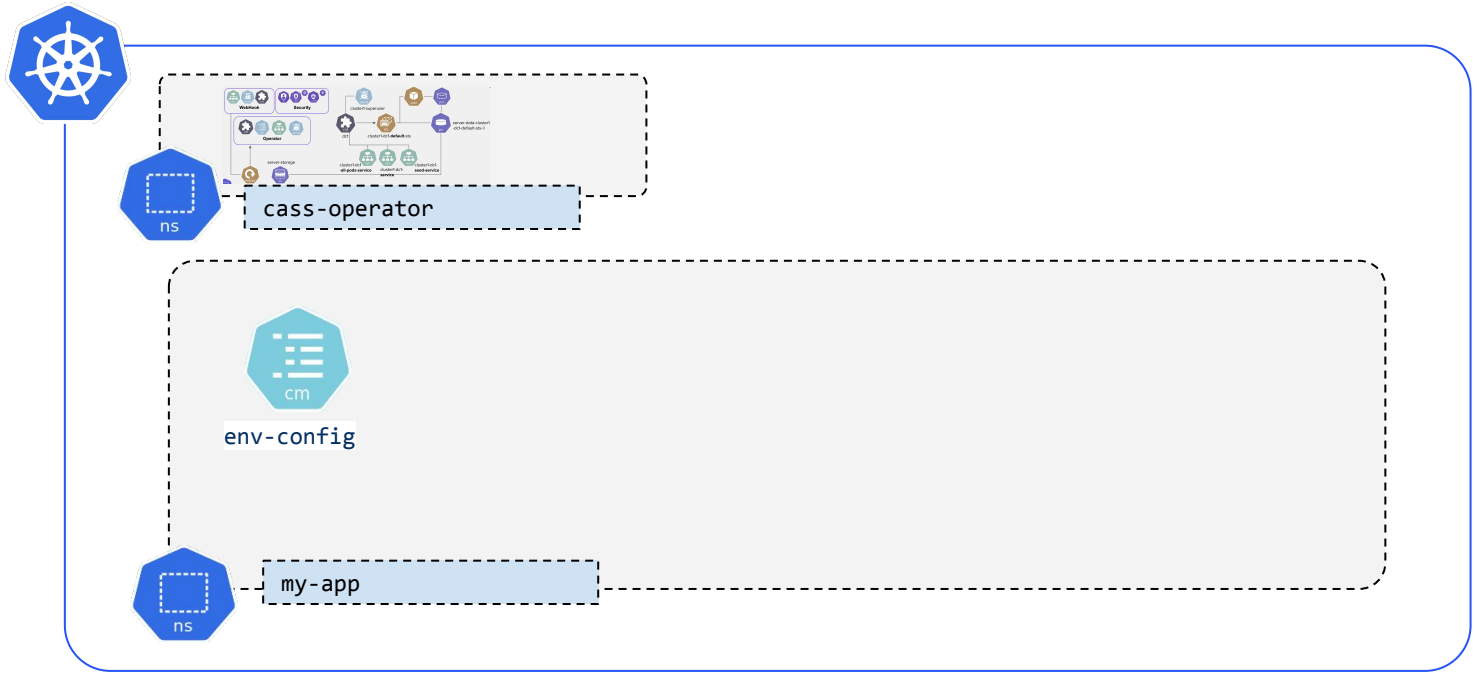
```
$ kubectl exec -it -n cass-operator cluster1-dc1-default-sts-0 /bin/bash
$ cqlsh -u cluster1-superuser -p CHANGEME
```

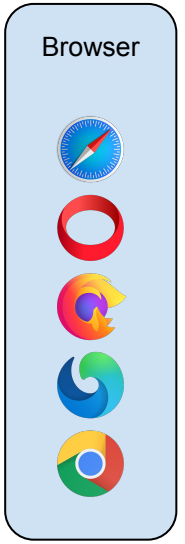
Setting up env configuration

kubernetes

configMap.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
 name: env-config
data:
 BASE_ADDRESS: 'http://localhost:5000/api'
 LOCAL_DATACENTER: 'dc1'
 USE_ASTRA: 'false'
 CONNECTION_POINTS: '10.244.32.5'
 KEYSPACE: 'killrvideo'
 USERNAME: 'cluster1-superuser'
 PASSWORD: 'C3dFhtsR68sIJYbs_cI_Tz5TFC4XVinS-O_f2jN9Lc1x6mHqIokk2Q'
```


cass-operator
ns
env-config
cm
my-app
ns


Browser

kubernetes

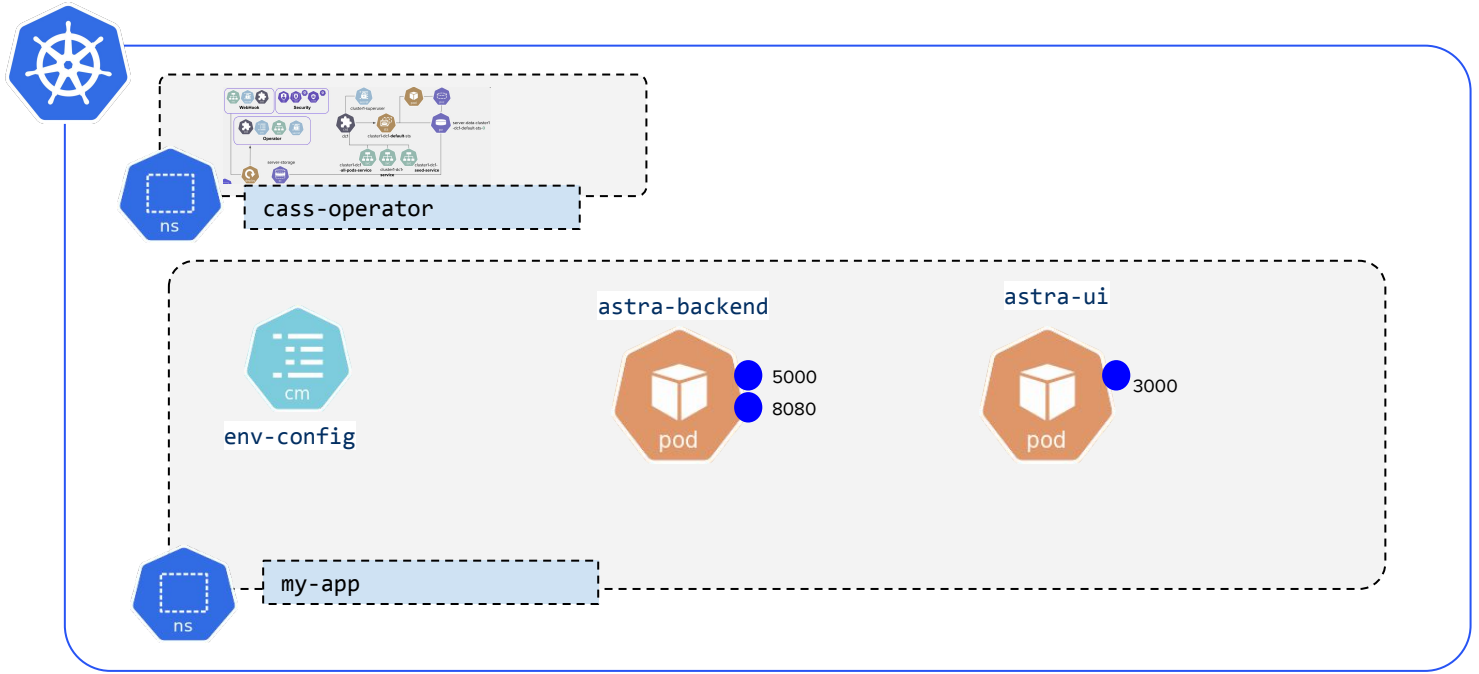
Yamls for your app

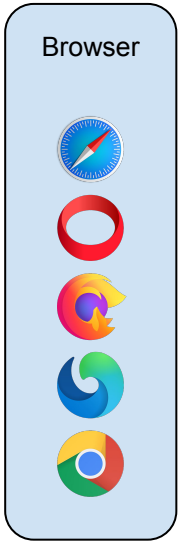
backend.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: astra-backend
spec:
  containers:
  - name: astra-backend
    image: astra-backend:my-image
    ports:
        - containerPort: 5000
        - containerPort: 8080
  envFrom:
    - configMapRef:
      name: env-config
```

ui.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: astra-ui
spec:
  containers:
  - name: astra-ui
    image: astra-ui:my-image
  envFrom:
    - configMapRef:
      name: env-config
```



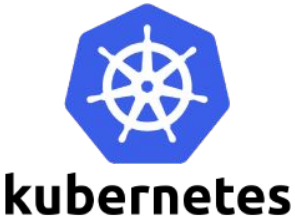


MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Starting your app

```
$ kubectl create ns my-app
$ cd kubefiles
$ kubectl -n my-app apply -f configMap.yaml
$ kubectl -n my-app apply -f ui.yaml
$ kubectl -n my-app apply -f backend.yaml
```



Forwarding your ports

```
$ kubectl -n my-app port-forward pods/astra-backend 5000:5000
$ kubectl -n my-app  port-forward pods/astra-ui 3000:3000
```

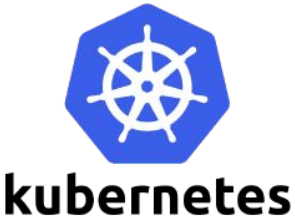
menti.com

96 57 69

Available on the iPhone App Store

GET IT ON Google play

LET’S DO IT FOR REAL!!!!



Cleaning up

```
$ kubectl delete ns cass-operator
$ kubectl delete ns my-app
$ kind delete cluster --name=kind-cassandra
```

Developer Workshop Series **Week 6**

- ➔ Housekeeping
- ➔ Docker Review
- ➔ Kubernetes Review
- ➔ Setting Up Containers
- ➔ Getting the App Running In k8
- ➔ Resources

MATERIALS ➔ bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Homework Week 6 (cf community)

1. **Practice**
    a. Finish workshop exercises if needed, following guidance on Github.
        i. https://github.com/DataStax-Academy/cassandra-workshop-series/tree/master/week6-App-in-k8s

2. **Validation form of the week**

AstraKathon *Reminder*

- 4 weeks to build real Applications

- END DATE IS NOW SET AFTER THE 8TH WEEK WORKSHOP
  - AUGUST 20TH 12pm PST.

- August 26th we will do an “ASK ME, ASK YOU ANYTHING” Special Episode with a lot of surprises and where we will announce winners.

DATASTAX®

Thank You

#ASK US ANYTHING

Discord box