ubuntu

The Canonical Distribution of Kubernetes

Easy k8s deployment and operations on Ubuntu

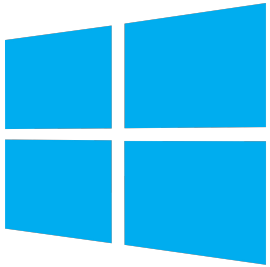
Presentation: https://goo.gl/WAvx3t

Files: https://goo.gl/ZuRhZC


kubeapi-load-bala...
easyrsa
etcd
flannel
kubernetes-worker
kubernetes-master

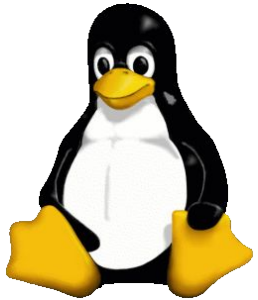
1. Installing Juju

documentation



Then install Putty & generate SSH Key

```
brew install --devel juju
```



UBUNTU

```
sudo add-apt-repository -y ppa:juju/devel
sudo apt update
sudo apt install -yqq juju
```

CENTOS

```
wget
https://launchpad.net/juju/2.1/2.1.2/+download/juju-2.1.2-centos7.tar.gz
tar xfz juju-2.1.2-centos7.tar.gz
mv juju-bin/juju* /usr/local/bin/
```

2. Adding Credentials

```
juju add-credential aws
Enter credential name: canonical

Using auth-type "access-key".

Enter access-key: A...6Q

Enter secret-key: ....

Credentials added for cloud aws.
```

3. Bootstrapping

```
juju bootstrap aws/us-east-1
Creating Juju controller "aws-us-east-1" on aws/us-east-1
Looking for packaged Juju agent version 2.1.2 for amd64
Launching controller instance(s) on aws/us-east-1...
 - i-0b453cdd805f93bd1 (arch=amd64 mem=8G cores=2)
Fetching Juju GUI 2.5.0
Waiting for address
Attempting to connect to 54.242.44.237:22
Attempting to connect to 172.31.15.160:22
Logging to /var/log/cloud-init-output.log on the bootstrap machine
Running apt-get update
Running apt-get upgrade
Installing curl, cpu-checker, bridge-utils, cloud-utils, tmux
Fetching Juju agent version 2.1.2 for amd64
Installing Juju machine agent
Starting Juju machine agent (service jujud-machine-0)
Bootstrap agent now started
Contacting Juju controller at 172.31.15.160 to verify accessibility...
Bootstrap complete, "aws-us-east-1" controller now available.
Controller machines are in the "controller" model.
Initial model "default" added.
```

4. Look at the Juju GUI

```
juju gui --show-credentials
GUI 2.5.0 for model "admin/k8s" is enabled at:
  https://54.89.34.211:17070/gui/u/admin/k8s
Your login credential is:
  username: admin
  password: ...
```

5. Deploy Kubernetes

```
juju add-model k8s
juju model-config resource-tags="KubernetesCluster=workshop"
juju deploy cs:bundle/kubernetes-core-15
Located bundle "cs:bundle/kubernetes-core-15"
Deploying charm "cs:~containers/easyrsa-7"
added resource easyrsa
Deploying charm "cs:~containers/etcd-24"
added resource snapshot
Deploying charm "cs:~containers/flannel-11"
added resource flannel
Deploying charm "cs:~containers/kubernetes-master-12"
added resource kubernetes
application kubernetes-master exposed
Deploying charm "cs:~containers/kubernetes-worker-14"
added resource kubernetes
application kubernetes-worker exposed
Related "kubernetes-master:kube-api-endpoint" and "kubernetes-worker:kube-api-endpoint"
Related "kubernetes-master:cluster-dns" and "kubernetes-worker:kube-dns"
Related "kubernetes-master:certificates" and "easyrsa:client"
Related "kubernetes-master:etcd" and "etcd:db"
Related "kubernetes-worker:certificates" and "easyrsa:client"
Related "etcd:certificates" and "easyrsa:client"
Related "flannel:etcd" and "etcd:db"
Related "flannel:cni" and "kubernetes-master:cni"
Related "flannel:cni" and "kubernetes-worker:cni"
Deploy of bundle completed.
```

6. Wait a few minutes

```
watch -c juju status --color
Model    Controller     Cloud/Region   Version
default  aws-us-east-1  aws/us-east-1  2.2-beta1

App                Version  Status       Scale  Charm              Store       Rev  OS      Notes
easyrsa                     waiting        0/1  easyrsa            jujucharms    7  ubuntu
etcd                        maintenance      1  etcd               jujucharms   24  ubuntu
flannel                     waiting          0  flannel            jujucharms   11  ubuntu
kubernetes-master           waiting        0/1  kubernetes-master  jujucharms   12  ubuntu  exposed
kubernetes-worker           maintenance      1  kubernetes-worker  jujucharms   14  ubuntu  exposed

Unit                  Workload     Agent       Machine  Public address  Ports  Message
easyrsa/0             waiting      allocating  0/lxd/0                        waiting for machine
etcd/0*               maintenance  executing   0        107.22.136.7          (install) installing charm
software
kubernetes-master/0*  waiting      allocating  0        107.22.136.7          agent initializing
kubernetes-worker/0*  maintenance  executing   1        107.23.240.130        (install) installing charm
software

Machine  State    DNS             Inst id              Series  AZ          Message
0        started  107.22.136.7    i-0605727f1a10d126e  xenial  us-east-1a  running
0/lxd/0  pending                  pending              xenial              Starting container
1        started  107.23.240.130  i-01cfaaa4beb8dafe8  xenial  us-east-1a  running

Relation      Provides           Consumes           Type
certificates  easyrsa            etcd               regular
certificates  easyrsa            kubernetes-master  regular
certificates  easyrsa            kubernetes-worker  regular
cluster       etcd               etcd               peer
etcd          etcd               flannel            regular
etcd          etcd               kubernetes-master  regular
cni           flannel            kubernetes-master  regular
```

```
watch -c juju status --color
Model    Controller     Cloud/Region   Version
default  aws-us-east-1  aws/us-east-1  2.2-beta1

App                Version  Status  Scale  Charm              Store       Rev  OS      Notes
easyrsa            3.0.1    active      1  easyrsa            jujucharms    7  ubuntu
etcd               2.2.5    active      1  etcd               jujucharms   24  ubuntu
flannel            0.7.0    active      2  flannel            jujucharms   11  ubuntu
kubernetes-master  1.5.3    active      1  kubernetes-master  jujucharms   12  ubuntu  exposed
kubernetes-worker  1.5.3    active      1  kubernetes-worker  jujucharms   14  ubuntu  exposed

Unit                  Workload  Agent  Machine  Public address  Ports     Message
easyrsa/0*            active    idle   0/lxd/0  10.0.198.142              Certificate Authority connected.
etcd/0*               active    idle   0        107.22.136.7    2379/tcp  Healthy with 1 known peers.
kubernetes-master/0*  active    idle   0        107.22.136.7    6443/tcp  Kubernetes master running.
  flannel/1           active    idle            107.22.136.7              Flannel subnet 10.1.32.1/24
kubernetes-worker/0*  active    idle   1        107.23.240.130            Kubernetes worker running.
  flannel/0*          active    idle            107.23.240.130            Flannel subnet 10.1.81.1/24

Machine  State    DNS             Inst id              Series  AZ          Message
0        started  107.22.136.7    i-0605727f1a10d126e  xenial  us-east-1a  running
0/lxd/0  started  10.0.198.142    juju-626321-0-lxd-0  xenial              Container started
1        started  107.23.240.130  i-01cfaaa4beb8dafe8  xenial  us-east-1a  running

Relation      Provides           Consumes           Type
certificates  easyrsa            etcd               regular
certificates  easyrsa            kubernetes-master  regular
certificates  easyrsa            kubernetes-worker  regular
cluster       etcd               etcd               peer
etcd          etcd               flannel            regular
etcd          etcd               kubernetes-master  regular
cni           flannel            kubernetes-master  regular
cni           flannel            kubernetes-worker  regular
```

7. Scale Out

```
juju add-unit kubernetes-worker
```

8. Accessing K8s

```
mkdir -p .kube
juju scp kubernetes-master/0:config ~/.kube/config

ARCHITECTURE=linux/darwin
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/${A
RCHITECTURE}/amd64/kubectl

# Windows
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/win
dows/amd64/kubectl.exe

chmod +x ~/kubectl && sudo mv kubectl /usr/local/bin/
```

```
kubectl cluster-info
Kubernetes master is running at https://34.209.23.103:6443
Heapster is running at
https://34.209.23.103:6443/api/v1/proxy/namespaces/kube-system/services/heapster
KubeDNS is running at
https://34.209.23.103:6443/api/v1/proxy/namespaces/kube-system/services/kube-dns
kubernetes-dashboard is running at
https://34.209.23.103:6443/api/v1/proxy/namespaces/kube-system/services/kubernetes-dashboard
Grafana is running at
https://34.209.23.103:6443/api/v1/proxy/namespaces/kube-system/services/monitoring-grafana
InfluxDB is running at
https://34.209.23.103:6443/api/v1/proxy/namespaces/kube-system/services/monitoring-influxdb

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Default password: admin / admin

9. Useful Juju commands

```
# Logging to a machine
juju ssh kubernetes-master/0
juju ssh 0


# Executing a command on remote machine
juju ssh kubernetes-master/0 “sudo cat /proc/cpuinfo”


# Downloading a file
juju scp kubernetes-master/0:config ./


# Uploading a file
juju scp ./test.txt kubernetes-master/0:test.txt
```

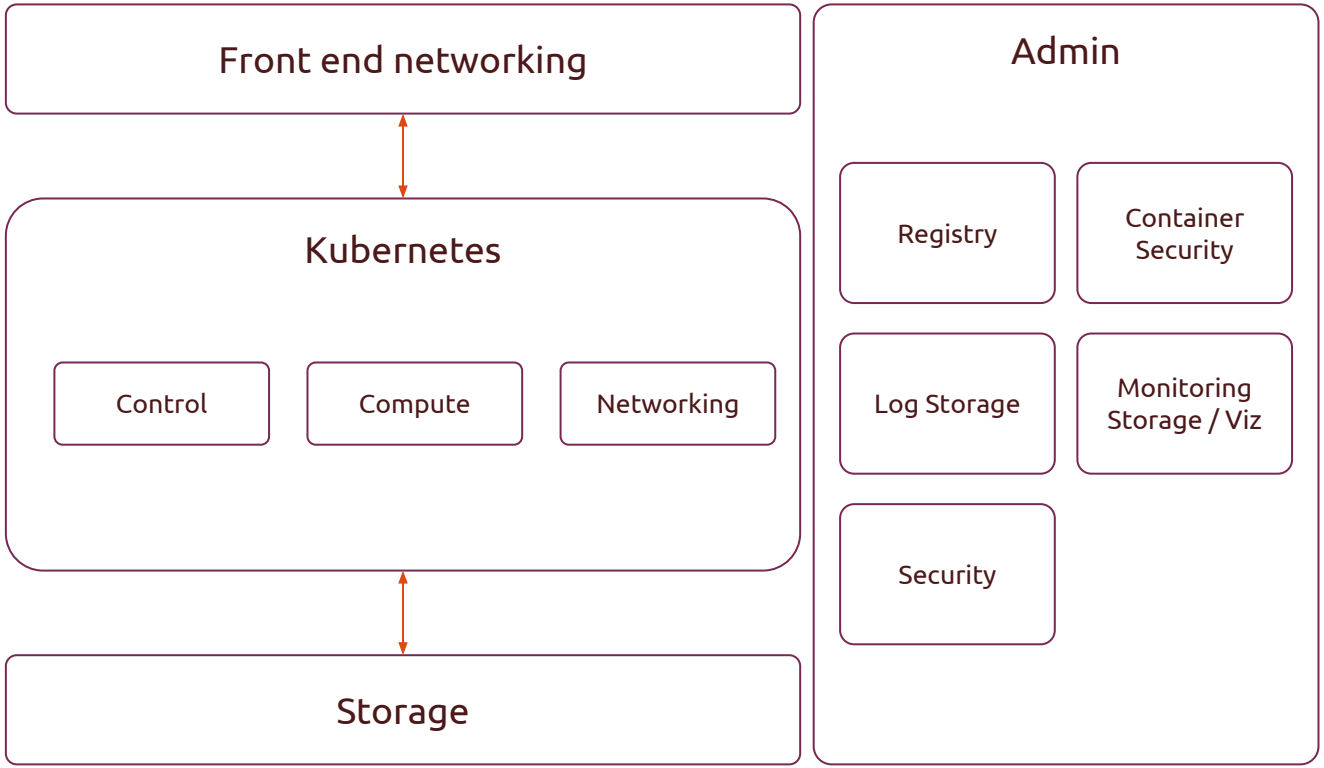
```
# List operational actions available
juju actions kubernetes-worker
Action            Description
clean-containers  Garbage collect non-running containers
clean-images      Garbage collect non-running images
debug             Collect debug data
microbot          Launch microbot containers
pause             Cordon the unit, draining all active workloads.
resume            UnCordon the unit, enabling workload scheduling.

# Executing an action
juju run-action kubernetes-worker/0 microbot
Action queued with id: 2184f1f5-4dea-4e34-8e1c-20778337f5d2

# Looking at the output
$ juju show-action-output 2184f1f5-4dea-4e34-8e1c-20778337f5d2
results:
  address: microbot.52.202.51.82.xip.io
status: completed
timing:
  completed: 2017-04-05 07:29:35 +0000 UTC
  enqueued: 2017-04-05 07:29:31 +0000 UTC
  started: 2017-04-05 07:29:33 +0000 UTC
```

10. More Juju

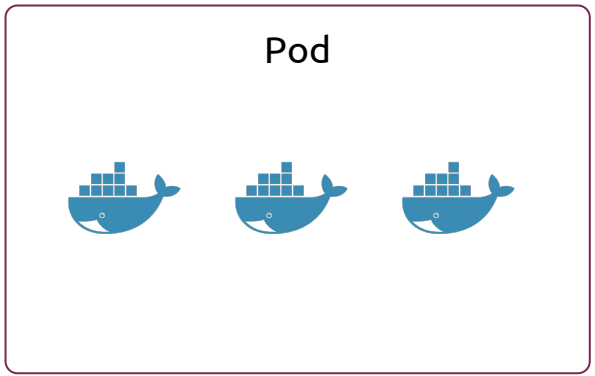
https://jujucharms.com

11. Kubernetes 101

Kubernetes Components



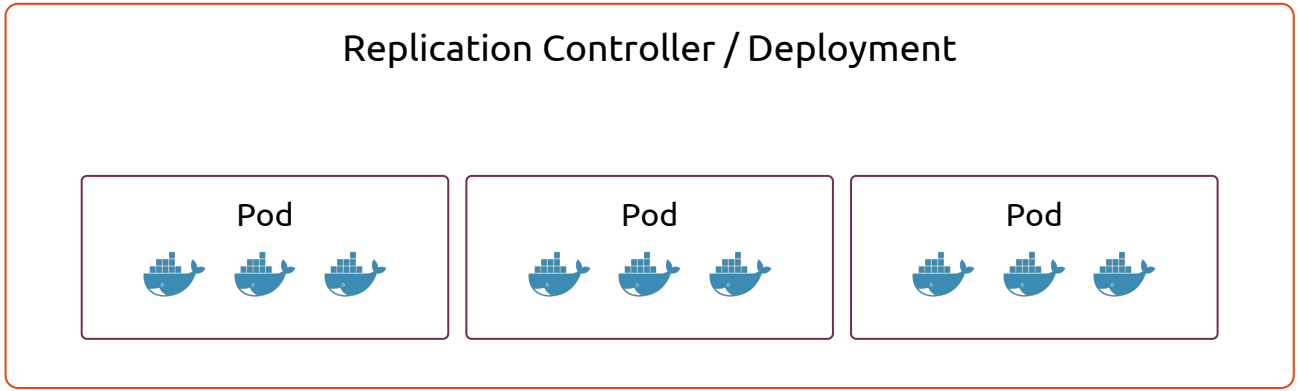

kubeapi-load-bala...
easyrsa
etcd
flannel
kubernetes-worker
kubernetes-master



Set of Containers

- Colocated on the same node
- Share the same IP address in the cluster
- Can talk to each other on localhost

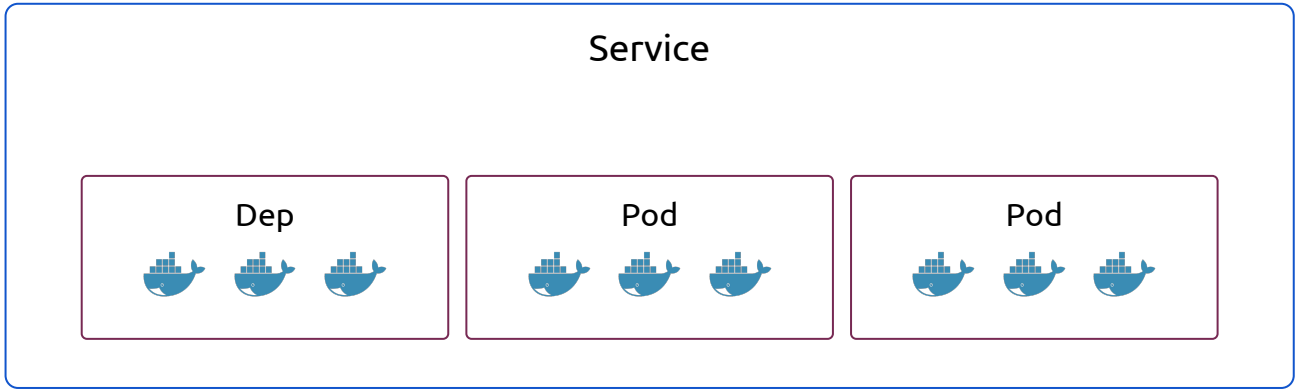
Pods are cattle! If they die… well… who cares?



Set of Pods with given

- Number of replicas
- Set of labels / properties
- Constraints

Spread as much as possible on the nodes
Self Healing properties



Load Balancer to Pods with

- Set of labels / properties

Service Discovery via ENV & DNS

Allows Canary Releases & Genetic Testing (A/B at scale)

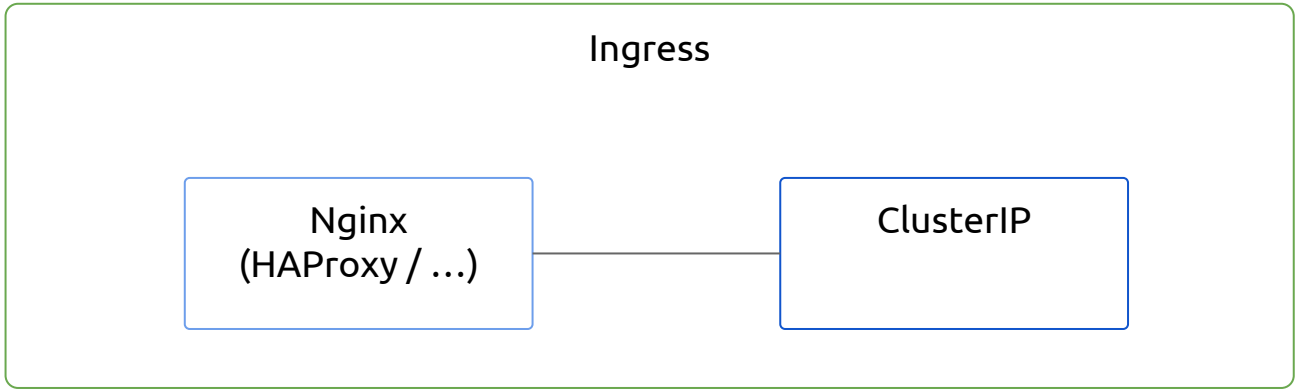
Can be external (Load Balancer) or internal (Cluster IP)

ConfigMap

```
kind: ConfigMap
apiVersion: v1
metadata:
  creationTimestamp: 2016-02-18T19:14:38Z
  name: example-config
  namespace: default
data:
  example.property.1: hello
  example.property.2: world
  example.property.file: |-
    property.1=value-1
    property.2=value-2
    property.3=value-3
```

Secret

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  username: YWRtaW4=
  password: MWYyZDFlMmU2N2Rm
```



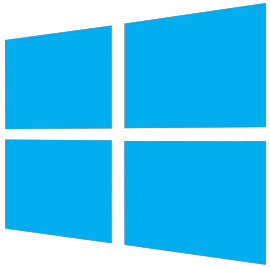
Proxy to Services.

Canonical example = nginx

Configured via ConfigMap, Secrets...

12. Install Helm

[documentation]

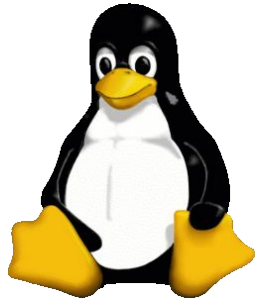


https://kubernetes-helm.storage.googleapis.com/helm-canary-windows-amd64.zip

Unzip to a directory

```
helm.exe init
helm.exe repo update
```

```
brew install kubernetes-helm

helm init
helm repo update
```



```
wget
https://kubernetes-helm.storage.googleapis.com/helm-v2.2.3-linux-amd64.tar.gz

tar xfz helm-v2.2.3-linux-amd64.tar.gz
chmod +x linux-amd64/helm
sudo mv linux-amd64/helm /usr/local/bin/

helm init
helm repo update
```

13. Helm Repository

https://kubeapps.com

14. Deploy an application

https://kubeapps.com/charts/stable/wordpress

```
helm install --name wordpress-ingress --set
wordpressUsername=admin,wordpressPassword=password,mariadb.mariadbRootPassword=secretpassword,persistence.enabled=f
alse,serviceType=ClusterIP,ingress.enabled=true,ingress.hostname=www.34.208.240.229.xip.io
kubernetes-charts/wordpress
NAME:   wordpress-ingress
LAST DEPLOYED: Wed Apr  5 12:21:52 2017
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Service
NAME                          CLUSTER-IP      EXTERNAL-IP  PORT(S)          AGE
wordpress-ingress-wordpress   10.152.183.228  <none>       80/TCP,443/TCP   3s
wordpress-ingress-mariadb     10.152.183.236  <none>       3306/TCP         3s

==> extensions/v1beta1/Deployment
NAME                          DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
wordpress-ingress-mariadb     1        1        1           0          3s
wordpress-ingress-wordpress   1        1        1           0          3s

==> extensions/v1beta1/Ingress
NAME                          HOSTS                        ADDRESS          PORTS  AGE
wordpress-ingress-wordpress   www.34.208.240.229.xip.io    54.68.157.164...  80     3s

==> v1/Secret
NAME                          TYPE    DATA  AGE
wordpress-ingress-mariadb     Opaque  2     3s
wordpress-ingress-wordpress   Opaque  3     3s
```

```
==> v1/ConfigMap
NAME                         DATA  AGE
wordpress-ingress-mariadb    1     3s

==> v1/PersistentVolumeClaim
NAME                         STATUS  VOLUME                                     CAPACITY  ACCESSMODES  AGE
wordpress-ingress-mariadb    Bound   pvc-b0dc9f92-19e9-11e7-af0f-064c860120ef   8Gi       RWO          3s


NOTES:
1. Get the WordPress URL:

  You should be able to access your new WordPress installation through

  http://www.34.208.240.229.xip.io/admin

2. Login with the following credentials to see your blog

  echo Username: admin
  echo Password: $(kubectl get secret --namespace default wordpress-ingress-wordpress -o
jsonpath="{.data.wordpress-password}" | base64 --decode)
```

```
kubectl get pods
NAME                                    READY     STATUS     RESTARTS   AGE
default-http-backend-c76gj              1/1       Running    0          17m
hello-world-2895499144-4fzt2            1/1       Running    0          15m
hello-world-2895499144-7gjh4            1/1       Running    0          15m
hello-world-2895499144-7zh9v            1/1       Running    0          15m
hello-world-2895499144-jk5x8            1/1       Running    0          15m
hello-world-2895499144-x6f6x            1/1       Running    0          15m
nginx-ingress-controller-1p65j          1/1       Running    0          17m
nginx-ingress-controller-hwc86          1/1       Running    0          18m
wordpress-mariadb-3172414723-zgm70      0/1       Init:0/1   0          1m
wordpress-wordpress-2210327583-2jnqm    0/1       Running    0          1m

After a while:
wordpress-mariadb-3172414723-zgm70      1/1       Running   0          22m
wordpress-wordpress-2210327583-2jnqm    1/1       Running   1          22m
```

```
kubectl get secret --namespace default wordpress-wordpress -o jsonpath="{.data.wordpress-password}" | base64
--decode
password

$ kubectl get ing -o wide
NAME                          HOSTS                       ADDRESS
PORTS     AGE
wordpress-ingress-wordpress   www.34.208.240.229.xip.io   54.68.157.164,34.208.240.229,34.208.240.229,35.167.80.144
80        1m
```

http://www.34.208.240.229.xip.io

15. Scale an application

```
kubectl get deploy
NAME                          DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
hello-world                   5         5         5            5           1h
wordpress-ingress-mariadb     1         1         1            1           17m
wordpress-ingress-wordpress   1         1         1            1           17m

kubectl scale deploy wordpress-ingress-wordpress --replicas=3
deployment "wordpress-ingress-wordpress" scaled

$ kubectl get pods -o wide
NAME                                          READY     STATUS    RESTARTS   AGE       IP              NODE
wordpress-ingress-mariadb-4241180659-drs87    1/1       Running   0          21m       10.1.61.9
ip-172-31-19-23.us-west-2.compute.internal
wordpress-ingress-wordpress-2693923398-5pr10  1/1       Running   0          15m       10.1.61.10
ip-172-31-19-23.us-west-2.compute.internal
wordpress-ingress-wordpress-2693923398-pq329  0/1       Running   0          1s        10.1.58.4
ip-172-31-7-167.us-west-2.compute.internal
wordpress-ingress-wordpress-2693923398-qzwtk  1/1       Running   0          3m        10.1.6.16
ip-172-31-43-243.us-west-2.compute.internal
```

16. Add ELB Support

```
aws ec2 describe-instances \
  --filters "Name=tag:juju-units-deployed,Values=*kubernetes-master*" | \
  jq --raw-output '.[][].Instances[].InstanceId' | \
  xargs -I {} aws ec2 associate-iam-instance-profile \
    --iam-instance-profile Name=k8sMaster-Instance-Profile \
    --instance-id {}

aws ec2 describe-instances --filters
"Name=tag:juju-units-deployed,Values=kubernetes-worker*" | \
  jq --raw-output '.[][].Instances[].InstanceId' | \
  xargs -I {} aws ec2 associate-iam-instance-profile --iam-instance-profile
Name=k8sWorker-Instance-Profile --instance-id {}
```

```
for KEY in KubernetesCluster juju-controller-uuid juju-model-uuid; do
  aws ec2 describe-instances | \
    jq --raw-output '.[][].Instances[] | select( .Tags[].Value  | contains ("k8s")) |
.SecurityGroups[1].GroupId' | \
    sort | uniq | \
    xargs -I {} aws ec2 delete-tags --resources {}  --tags "Key="${KEY}
done

aws ec2 describe-instances | \
  jq --raw-output '.[][].Instances[] | select( .Tags[].Value  | contains ("k8s")) |
.SecurityGroups[0].GroupId' | \
  sort | uniq | \
  xargs -I {} aws ec2 delete-tags --resources {} --tags "Key=juju-model-uuid"

aws ec2 describe-instances | \
  jq --raw-output '.[][].Instances[] | select( .Tags[].Value  | contains ("k8s")) |
.SecurityGroups[0].GroupId' | \
  sort | uniq | \
  xargs -I {} aws ec2 delete-tags --resources {} --tags "Key=juju-controller-uuid"

juju status kubernetes-worker --format json | \
  jq -r '.machines[]."instance-id"' | \
  xargs -I {} aws ec2 describe-instances --instance-ids {} | \
  jq --raw-output '.[][].Instances[].SubnetId' | sort | uniq | \
  xargs -I {} aws ec2 create-tags --resources {} --tags
"Key=KubernetesCluster,Value=workshop"
```

```
juju show-status kubernetes-master --format json | \
       jq -r '.applications."kubernetes-master".units | keys[]' | \
       xargs -I UNIT juju ssh UNIT "sudo sed -i
's/KUBE_CONTROLLER_MANAGER_ARGS=\"/KUBE_CONTROLLER_MANAGER_ARGS=\"--cloud-provider=aws\ /'
/etc/default/kube-controller-manager && sudo systemctl restart
kube-controller-manager.service"

juju show-status kubernetes-master --format json | \
       jq -r '.applications."kubernetes-master".units | keys[]' | \
       xargs -I UNIT juju ssh UNIT "sudo sed -i
's/KUBE_API_ARGS=\"/KUBE_API_ARGS=\"--cloud-provider=aws\ /' /etc/default/kube-apiserver
&& sudo systemctl restart kube-apiserver.service"

juju show-status kubernetes-worker --format json | \
       jq -r '.applications."kubernetes-worker".units | keys[]' | \
       xargs -I UNIT juju ssh UNIT "sudo sed -i
's/KUBELET_ARGS=\"/KUBELET_ARGS=\"--cloud-provider=aws\ /' /etc/default/kubelet && sudo
systemctl restart kubelet.service"
```

```
kubectl run hello-world \
     --replicas=5 \
     --labels="run=load-balancer-example" \
     --image=gcr.io/google-samples/node-hello:1.0 \
     --port=8080

kubectl expose deployment hello-world \
     --type=LoadBalancer \
     --name=hello

kubectl get svc -o wide
NAME                   CLUSTER-IP       EXTERNAL-IP
PORT(S)          AGE       SELECTOR
default-http-backend   10.152.183.242   <none>
80/TCP           39m       app=default-http-backend
hello                  10.152.183.206
a11d4e8dc19e111e7af0f064c860120e-480036888.us-west-2.elb.amazonaws.com   8080:30248/TCP
2m        run=load-balancer-example
kubernetes             10.152.183.1     <none>
443/TCP          43m       <none>
```

17. Deploy an application w/ ELB

https://kubeapps.com/charts/stable/wordpress

```
helm install --name wordpress --set
wordpressUsername=admin,wordpressPassword=password,mariadb.mariadbRootPassword=secretpassword,persistence.enabled=f
alse kubernetes-charts/wordpress
NAME:   wordpress
LAST DEPLOYED: Wed Apr  5 11:33:08 2017
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Secret
NAME                 TYPE    DATA  AGE
wordpress-mariadb    Opaque  2     2s
wordpress-wordpress  Opaque  3     2s

==> v1/ConfigMap
NAME               DATA  AGE
wordpress-mariadb  1     2s

==> v1/PersistentVolumeClaim
NAME               STATUS  VOLUME                                     CAPACITY  ACCESSMODES  AGE
wordpress-mariadb  Bound   pvc-e1b3c086-19e2-11e7-af0f-064c860120ef  8Gi       RWO          2s

==> v1/Service
NAME                 CLUSTER-IP     EXTERNAL-IP       PORT(S)                      AGE
wordpress-wordpress  10.152.183.80  ae1bf0c2d19e2...  80:32452/TCP,443:31304/TCP  2s
wordpress-mariadb    10.152.183.83  <none>            3306/TCP                     2s
```

```
==> extensions/v1beta1/Deployment
NAME                 DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
wordpress-wordpress  1        1        1           0          2s
wordpress-mariadb    1        1        1           0          2s


NOTES:
1. Get the WordPress URL:

  NOTE: It may take a few minutes for the LoadBalancer IP to be available.
        Watch the status with: 'kubectl get svc --namespace default -w wordpress-wordpress'

  export SERVICE_IP=$(kubectl get svc --namespace default wordpress-wordpress -o
jsonpath='{.status.loadBalancer.ingress[0].ip}')
  echo http://$SERVICE_IP/admin

2. Login with the following credentials to see your blog

  echo Username: admin
  echo Password: $(kubectl get secret --namespace default wordpress-wordpress -o
jsonpath="{.data.wordpress-password}" | base64 --decode)
```

```
kubectl get pods
NAME                                    READY     STATUS     RESTARTS   AGE
default-http-backend-c76gj              1/1       Running    0          17m
hello-world-2895499144-4fzt2            1/1       Running    0          15m
hello-world-2895499144-7gjh4            1/1       Running    0          15m
hello-world-2895499144-7zh9v            1/1       Running    0          15m
hello-world-2895499144-jk5x8            1/1       Running    0          15m
hello-world-2895499144-x6f6x            1/1       Running    0          15m
nginx-ingress-controller-1p65j          1/1       Running    0          17m
nginx-ingress-controller-hwc86          1/1       Running    0          18m
wordpress-mariadb-3172414723-zgm70      0/1       Init:0/1   0          1m
wordpress-wordpress-2210327583-2jnqm    0/1       Running    0          1m

After a while:
wordpress-mariadb-3172414723-zgm70      1/1       Running   0          22m
wordpress-wordpress-2210327583-2jnqm    1/1       Running   1          22m
```

```
kubectl get secret --namespace default wordpress-wordpress -o jsonpath="{.data.wordpress-password}" | base64
--decode
password

kubectl get svc -o wide
NAME                    CLUSTER-IP       EXTERNAL-IP
PORT(S)                       AGE       SELECTOR
default-http-backend    10.152.183.242    <none>
80/TCP                        57m       app=default-http-backend
hello                   10.152.183.206    a11d4e8dc19e111e7af0f064c860120e-480036888.us-west-2.elb.amazonaws.com
8080:30248/TCP                20m       run=load-balancer-example
kubernetes              10.152.183.1      <none>
443/TCP                       1h        <none>
wordpress-mariadb       10.152.183.83     <none>
3306/TCP                      7m        app=wordpress-mariadb
wordpress-wordpress     10.152.183.80     ae1bf0c2d19e211e7af0f064c860120e-1015045695.us-west-2.elb.amazonaws.com
80:32452/TCP,443:31304/TCP    7m        app=wordpress-wordpress
```

18. High Availability (k8s)

```
# Setting constraints for services
juju set-constraints etcd “mem=4G cores=4 root-disk=16G”

# Scaling etcd
juju add-unit etcd -n2

# Scaling Kubernetes Control Plane
juju add-unit kubernetes-master -n2
```

19. Add EFS Storage

```
VPC_ID=$(aws ec2 describe-vpcs \
    | jq -r '.[][].VpcId')
echo $VPC_ID

SUBNET_IDS=$(aws ec2 describe-subnets \
    | jq -r '.[][].SubnetId')
echo $SUBNET_IDS

SG_ID=”$(aws ec2 describe-instances \
    | jq -r '.[][].Instances[] | select( .Tags[].Value  | contains ("'"k8s"'")) |
.SecurityGroups[1].GroupId' | sort | uniq | tr '\n' ' ')”
echo $SG_ID
# Note: k8s here is the model name

EFS_ID=$(aws efs create-file-system --creation-token $(uuid) \
    | jq -r '.FileSystemId')
echo $EFS_ID

for SUBNET in ${SUBNET_IDS}
do
    aws efs create-mount-target \
        --file-system-id ${EFS_ID} \
        --subnet-id ${SUBNET} \
        --security-groups ${SG_ID}
done
```

20. Tearing down

```
juju destroy-model k8s

juju destroy-controller <controller name>
```

21. Questions!


ubuntu