



Kubernetes Essentials

Activity guide

Release 1 rev46

SAMPLE

Component Soft Ltd.

January 12, 2018

The contents of this course and all its modules and related materials, including handouts to audience members, are copyright © 2018 Component Soft Ltd.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Component Soft Ltd.

This curriculum contains proprietary information which is for the exclusive use of customers of Component Soft Ltd., and is not to be shared with personnel other than those in attendance at this course.

This instructional program, including all material provided herein, is supplied without any guarantees from Component Soft Ltd. Component Soft Ltd. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

Photocopying any part of this manual without prior written consent of Component Soft Ltd. is a violation of law. This manual should not appear to be a photocopy. If you believe that Component Soft Ltd. training materials are being photocopied without permission, please write an email to info@componentsoft.eu.

Component Soft Ltd. accepts no liability for any claims, demands, losses, damages, costs or expenses suffered or incurred howsoever arising from or in connection with the use of this courseware. All trademarks are the property of their respective owners.

SAMPLE

Contents

Lab 1: Introduction to containers	1
Task 1: Health check	1
Task 2: Understand linux namespaces	2
Task 3: Basic docker functionalities	4
Task 4: Cleanup	7
Lab 2: Checking Kubernetes components	9
Task 1: Check Kubernetes status	9
Task 2: Run the first pod	12
Task 3: Cleanup	13
Lab 3: Accessing the kubernetes API	15
Task 1: Browse the kubernetes API	15
Task 2: Use RBAC to controll access to the API	19
Task 3: Cleanup	21
Lab 4: Kubernetes workloads	23
Task 1: Pod operations	23
Task 2: Replication controller operations	28
Task 3: Working with deployments	30
Task 4: Using Jobs	32
Task 5: Using DaemonSets	34
Lab 5: Scheduling and node management	37
Task 1: Scheduling Pods to nodes	37
Task 2: Using affinities	38
Lab 6: Accessing the applications	41
Task 1: Working with services	41
Task 2: working with Ingress	45
Lab 7: Using persistent storage	47
Task 1: Share a volume in two containers	47
Task 2: Set the root password for a mysql pod using Secrets	51

Task 3: Use ConfigMap to pass a file to a pod	52
Lab 8: Logging, monitoring kubernetes	55
Task 1: Investigate the logging in kubernetes	55
Task 2: Monitoring kubernetes	56
Lab 9: Upgrading kubernetes	59

SAMPLE

Lab 1: Introduction to containers

Task 1: Health check

Perform basic health check on your local kubernetes installation.

- On your **lab host**, check whether your nodes are running

```
root@lab_machine $> /labfiles/os_nodes list
Id      Name      State
-----
28      master1   running
29      worker1   running
30      worker2   running
31      worker3   running
```

- Autoaccept the ssh-keys of the nodes

```
root@lab_machine $> for i in master1 worker{1,2,3}; do
> ssh-keyscan -H $i >> ~/.ssh/known_hosts
> done
# master1 SSH-2.0-OpenSSH_6.6.1
# master1 SSH-2.0-OpenSSH_6.6.1
# worker1 SSH-2.0-OpenSSH_6.6.1
# worker1 SSH-2.0-OpenSSH_6.6.1
# worker2 SSH-2.0-OpenSSH_6.6.1
# worker2 SSH-2.0-OpenSSH_6.6.1
# worker3 SSH-2.0-OpenSSH_6.6.1
# worker3 SSH-2.0-OpenSSH_6.6.1
root@lab_machine $>
```

- Check whether VMs have the same system date and time (1-2 seconds of difference is OK).

```
root@lab_machine $> for i in master1 worker{1,2,3}
> do
> echo -en "$i :"; ssh $i 'date'
> done
master1 :Warning: Permanently added the ECDSA host key for IP address
'10.10.10.51' to the list of known hosts.
Tue May 23 07:19:09 UTC 2017
worker1 :Warning: Permanently added the ECDSA host key for IP address
'10.10.10.52' to the list of known hosts.
Tue May 23 07:19:09 UTC 2017
worker2 :Warning: Permanently added the ECDSA host key for IP address
```

```
'10.10.10.53' to the list of known hosts.  
Tue May 23 07:19:09 UTC 2017  
worker3 :Warning: Permanently added the ECDSA host key for IP address  
'10.10.10.54' to the list of known hosts.  
Tue May 23 07:19:09 UTC 2017
```

Task 2: Understand linux namespaces

In this task we introduce a few commands that can be useful to understand how containers are working.

- List the namespaces of a process

```
$> ls -l /proc/<PID>/ns
```

```
root@lab_machine $> ls -l /proc/1/ns  
total 0  
lrwxrwxrwx 1 root root 0 May 23 10:13 ipc -> ipc:[4026531839]  
lrwxrwxrwx 1 root root 0 May 23 10:13 mnt -> mnt:[4026531840]  
lrwxrwxrwx 1 root root 0 May 23 10:13 net -> net:[4026531956]  
lrwxrwxrwx 1 root root 0 May 23 10:13 pid -> pid:[4026531836]  
lrwxrwxrwx 1 root root 0 May 23 10:13 user -> user:[4026531837]  
lrwxrwxrwx 1 root root 0 May 23 10:13 uts -> uts:[4026531838]  
root@lab_machine $>
```

- Execute a command in a new namespace

```
root@lab_machine $> unshare --net bash  
root@lab_machine $> ls -l /proc/$$/ns/net  
lrwxrwxrwx 1 root root 0 May 23 13:01 /proc/18189/ns/net -> net:[4026532484]  
root@lab_machine $> ip a  
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN qlen 1  
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
root@lab_machine $> ip link add ns-test type dummy  
root@lab_machine $> ip a  
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN qlen 1  
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
2: ns-test: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN qlen 1000  
   link/ether ce:df:bb:7a:e8:90 brd ff:ff:ff:ff:ff:ff  
root@lab_machine $>
```

- Execute a command in the namespace of an other command

First we need the PID of the command to whose namespace we want to attach (we assume that we are in the shell that we have started using unshare)

```
root@lab_machine $> echo $$  
18189
```

Run the following commands in a **new** terminal

```
root@lab_machine $> nsenter -t 18189 --net bash
root@lab_machine $> ip a
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ns-test: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN qlen 1000
   link/ether ce:df:bb:7a:e8:90 brd ff:ff:ff:ff:ff:ff
root@lab_machine $>
```

Exit the new shell and check the network links

```
root@lab_machine $> exit
exit
root@lab_machine $> ip a | head
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: br_management: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
   qlen 1000
   link/ether fe:00:27:00:01:01 brd ff:ff:ff:ff:ff:ff
   inet 10.10.10.1/24 brd 10.10.10.255 scope global br_management
       valid_lft forever preferred_lft forever
root@lab_machine $>
```

Close the second terminal, and make sure that the shell in the current terminal can see the system's network interfaces (if needed exit the shell that was started with unshare).

```
root@lab_machine $> ip a | head
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ns-test: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN qlen 1000
   link/ether ba:3e:1b:13:67:14 brd ff:ff:ff:ff:ff:ff
root@lab_machine $> exit
exit
root@lab_machine $> ip a | head
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: br_management: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
   qlen 1000
   link/ether fe:00:27:00:01:01 brd ff:ff:ff:ff:ff:ff
   inet 10.10.10.1/24 brd 10.10.10.255 scope global br_management
       valid_lft forever preferred_lft forever

root@lab_machine $>
```

Task 3: Basic docker functionalities

In this task we are presenting basic docker functionality.

Run the following commands on the **master1** node

- Check locally available docker images

```
root@master1 $> docker images
REPOSITORY                                TAG                                IMAGE ID
gcr.io/google_c..s/kube-proxy-amd64      v1.6.3                            7d1bd9707c45    13 days ago    109.2 MB
gcr.io/kube-controller-manager-amd64    v1.6.3                            d2888c09d1e6    13 days ago    132.8 MB
gcr.io/goog..s/kube-apiserver-amd64      v1.6.3                            b80d5b7319cc    13 days ago    150.6 MB
gcr.io/goog..s/kube-scheduler-amd64      v1.6.3                            71a568bd21be    13 days ago    76.76 MB
docker.io/weaveworks/weave-npc           1.9.5                             edd184a7e1e4    2 weeks ago    58.23 MB
docker.io/weaveworks/weave-kube          1.9.5                             01af0bd5fdaf    2 weeks ago    163.3 MB
gcr.io/google_containers/etcd-amd64      3.0.17                             243830dae7dd    12 weeks ago   168.9 MB
gcr.io/google_containers/pause-amd64     3.0                                 99e59f495ffa    12 months ago  746.9 kB

root@master1 $>
```

- Retrieve an image

```
root@master1 $> docker pull ubuntu
Using default tag: latest
Trying to pull repository docker.io/library/ubuntu ...
latest: Pulling from docker.io/library/ubuntu
b6f892c0043b: Pull complete
55010f332b04: Pull complete
2955fb827c94: Pull complete
3deef3fcbd30: Pull complete
cf9722e506aa: Pull complete
```



```
Digest: sha256:382452f82a8bbd34443b2c727650af46aced0f94a44463c62a9848133ecb1aa8
root@master1 $> docker images
REPOSITORY                                TAG                IMAGE ID           CREATED           SIZE
docker.io/ubuntu                          latest            ebcd9d4fca80     8 days ago      117.9 MB
gcr.io/google_c..s/kube-proxy-amd64      v1.6.3          7d1bd9707c45     13 days ago     109.2 MB
gcr.io/kube-controller-manager-amd64    v1.6.3          d2888c09d1e6     13 days ago     132.8 MB
gcr.io/goog..s/kube-apiserver-amd64     v1.6.3          b80d5b7319cc     13 days ago     150.6 MB
gcr.io/goog..s/kube-scheduler-amd64    v1.6.3          71a568bd21be     13 days ago     76.76 MB
docker.io/weaveworks/weave-npc          1.9.5           edd184a7e1e4     2 weeks ago     58.23 MB
docker.io/weaveworks/weave-kube        1.9.5           01af0bd5fdaf     2 weeks ago     163.3 MB
gcr.io/google_containers/etcd-amd64    3.0.17         243830dae7dd     12 weeks ago    168.9 MB
gcr.io/google_containers/pause-amd64   3.0            99e59f495ffa     12 months ago   746.9 kB
root@master1 $>
```

- Run a shell in a container using the ubuntu image

```
root@master1 $> docker run --name test-cnt -ti ubuntu bash
root@blcfd78942d3 $>
```

- Check what processes are running in the container

```
root@blcfd78942d3 $> ps -ef
UID          PID    PPID  C   STIME TTY          TIME CMD
root         1      0   0  15:11 ?           00:00:00 bash
root        10     1   0  15:12 ?           00:00:00 ps -ef
root@blcfd78942d3 $>
```

- Install the iproute2 and iptutils-ping commands

```
root@blcfd78942d3 $> apt update
Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
...
Fetched 24.0 MB in 6s (3701 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@blcfd78942d3 $> apt install -y iproute2 iptutils-ping
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libatml libffi6 libgmp10 libgnutls-openssl27 libgnutls30 libhogweed4
  libidn11 libmn10 libnettle6 libp11-kit0 libtasn1-6 libxtables11
Suggested packages:
  iproute2-doc gnutls-bin
The following NEW packages will be installed:
  iproute2 iptutils-ping libatml libffi6 libgmp10 libgnutls-openssl27
  libgnutls30 libhogweed4 libidn11 libmn10 libnettle6 libp11-kit0 libtasn1-6
  libxtables11
0 upgraded, 14 newly installed, 0 to remove and 3 not upgraded.
Need to get 1889 kB of archives.
After this operation, 5586 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu xenial/main amd64 libatml amd64 1:2.5.1-1.5 [24.2 kB]i
...
```

```
Setting up libgnutls-openssl27:amd64 (3.4.10-4ubuntu1.2) ...
Setting up iputils-ping (3:20121221-5ubuntu2) ...
Setcap is not installed, falling back to setuid
Setting up libxtables11:amd64 (1.6.0-2ubuntu3) ...
Processing triggers for libc-bin (2.23-0ubuntu7) ...
root@b1cfd78942d3 $>
```

- Test network configuration and connectivity

```
root@b1cfd78942d3 $> ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
22: eth0@if23: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:acff:fe11:2/64 scope link
        valid_lft forever preferred_lft forever
root@b1cfd78942d3 $> ping www.google.com
PING www.google.com (172.217.18.68) 56(84) bytes of data.
64 bytes from bud02s26-in-f4.1e100.net (172.217.18.68): icmp_seq=1 ttl=52 time=17.9 ms
64 bytes from bud02s26-in-f4.1e100.net (172.217.18.68): icmp_seq=2 ttl=52 time=16.1 ms
64 bytes from bud02s26-in-f4.1e100.net (172.217.18.68): icmp_seq=3 ttl=52 time=13.6 ms
^C
--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 13.660/15.923/17.979/1.769 ms
root@b1cfd78942d3 $>
```

- Start a long running process in the container, and check its visibility from the OS

```
root@b1cfd78942d3 $> sleep 1000 &
[1] 363
root@b1cfd78942d3 $> ps -ef
UID          PID    PPID  C   STIME TTY          TIME CMD
root           1        0   0   May24 ?           00:00:00 bash
root          363        1   0   07:57 ?           00:00:00 sleep 1000
root          364        1   0   07:57 ?           00:00:00 ps -ef
root@b1cfd78942d3 $>
```

Press Ctrl+P followed by Ctrl+q to detach from the container environment

```
root@master1 $> ps -ef | grep [s]leep\ 1000
root          6258  22613   0   07:57 pts/1       00:00:00 sleep 1000
root@master1 $>
```

- Stop the container and check whether the sleep process is still running

```
root@master1 $> docker stop test-cnt
test-cnt
root@master1 $> ps -ef | grep [s]leep\ 1000
root@master1 $>
```

Task 4: Cleanup

- Delete the test-cnt container and the ubuntu image

```
root@master1 $> docker rm test-cnt
test-cnt
root@master1 $> docker rmi ubuntu
Untagged: ubuntu:latest
Untagged: docker.io/ubuntu@sha256:382452f82a8bbd34443b2c727650af46aced0f94a44463c62a9848133eck
Deleted: sha256:ebcd9d4fca80e9e8afc525d8a38e7c56825dfb4a220ed77156f9fb13b14d4ab7
Deleted: sha256:ef5b99eed7c2ed19ef39f72ac19bb66e16ed6c0868053daae60306a73858fbd4
Deleted: sha256:257e51479af1e9d2e0c9b958e68f6b992329904df24d81efa191cef515a9bf8b
Deleted: sha256:6e1d2d371500e2fe6df75f5755d0b9f2a3b69a42fe88100d514212bbba7ad23f
Deleted: sha256:afa9e7a5e3f3b006942d128c562a3273947c7ab50cdac33fea7213890072a5b6
Deleted: sha256:2df9b8def18a090592bf1cbd1079e1ac2274435c53f027ee5ce0a8faaa5d6d4b
root@master1 $> docker ps -a | grep ubuntu
root@master1 $> docker images | grep ubuntu
root@master1 $>
```

SAMPLE

Lab 2: Checking Kubernetes components

Task 1: Check Kubernetes status

Kubernetes 1.6.3 has been installed on the nodes using kubeadm. Kubeadm is configuring the kubelet system service to run the kubernetes control plane components as docker containers.

- Verify that the kubelet service is running on the nodes

```
root@lab_machine $> NODES="master1 worker1 worker2 worker3"

root@lab_machine $> for i in $NODES
> do echo $i
> ssh $i "systemctl status kubelet.service | head -10"
> echo
> done

kubelet.service - kubelet: The Kubernetes Node Agent
  Loaded: loaded (/etc/systemd/system/kubelet.service; enabled; vendor preset: disabled)
  Drop-In: /etc/systemd/system/kubelet.service.d
           \-10-kubeadm.conf
  Active: active (running) since Ma 2017-05-23 14:42:54 UTC; 3 weeks 0 days ago
  Docs: http://kubernetes.io/docs/
  Main PID: 1260 (kubelet)
  CGroup: /system.slice/kubelet.service
          -1260 /usr/bin/kubelet --kubeconfig=/etc/kubernetes/kubelet.conf
            --require-kubeconfig=true --pod-manifest-path=/etc/kubernetes/manifests
            --allow-privileged=true --network-plugin=cni --cni-conf-dir=/etc/cni/net.d
            --cni-bin-dir=/opt/cni/bin --cluster-dns=10.96.0.10
            --cluster-domain=cluster.local
            --authorization-mode=Webhook --client-ca-file=/etc/kubernetes/pki/ca.crt
            --cgroup-driver=systemd
          \-1336 journalctl -k -f

worker1
kubelet.service - kubelet: The Kubernetes Node Agent
  Loaded: loaded (/etc/systemd/system/kubelet.service; enabled; vendor preset: disabled)
  Drop-In: /etc/systemd/system/kubelet.service.d
           |-10-kubeadm.conf
  Active: active (running) since Ma 2017-05-23 14:46:12 UTC; 3 weeks 0 days ago
  Docs: http://kubernetes.io/docs/
  Main PID: 1113 (kubelet)
  CGroup: /system.slice/kubelet.service
          |-1113 /usr/bin/kubelet --kubeconfig=/etc/kubernetes/kubelet.conf
            --require-kubeconfig=true --pod-manifest-path=/etc/kubernetes/manifests
```

```
    --allow-privileged=true --network-plugin=cni --cni-conf-dir=/etc/cni/net.d
    --cni-bin-dir=/opt/cni/bin --cluster-dns=10.96.0.10
    --cluster-domain=cluster.local
    --authorization-mode=Webhook --client-ca-file=/etc/kubernetes/pki/ca.crt
    --cgroup-driver=systemd
    \-1301 journalctl -k -f

worker2
kubelet.service - kubelet: The Kubernetes Node Agent
  Loaded: loaded (/etc/systemd/system/kubelet.service; enabled; vendor preset: disabled)
Drop-In: /etc/systemd/system/kubelet.service.d
         \-10-kubeadm.conf
  Active: active (running) since Ma 2017-05-23 14:46:13 UTC; 3 weeks 0 days ago
  Docs: http://kubernetes.io/docs/
Main PID: 1132 (kubelet)
CGroup: /system.slice/kubelet.service
        |-1132 /usr/bin/kubelet --kubeconfig=/etc/kubernetes/kubelet.conf
        --require-kubeconfig=true --pod-manifest-path=/etc/kubernetes/manifests
        --allow-privileged=true --network-plugin=cni --cni-conf-dir=/etc/cni/net.d
        --cni-bin-dir=/opt/cni/bin --cluster-dns=10.96.0.10
        --cluster-domain=cluster.local
        --authorization-mode=Webhook --client-ca-file=/etc/kubernetes/pki/ca.crt
        --cgroup-driver=systemd
        \-1292 journalctl -k -f

worker3
kubelet.service - kubelet: The Kubernetes Node Agent
  Loaded: loaded (/etc/systemd/system/kubelet.service; enabled; vendor preset: disabled)
Drop-In: /etc/systemd/system/kubelet.service.d
         \-10-kubeadm.conf
  Active: active (running) since Ma 2017-05-23 14:46:13 UTC; 3 weeks 0 days ago
  Docs: http://kubernetes.io/docs/
Main PID: 1103 (kubelet)
CGroup: /system.slice/kubelet.service
        |-1103 /usr/bin/kubelet --kubeconfig=/etc/kubernetes/kubelet.conf
        --require-kubeconfig=true --pod-manifest-path=/etc/kubernetes/manifests
        --allow-privileged=true --network-plugin=cni --cni-conf-dir=/etc/cni/net.d
        --cni-bin-dir=/opt/cni/bin --cluster-dns=10.96.0.10
        --cluster-domain=cluster.local
        --authorization-mode=Webhook --client-ca-file=/etc/kubernetes/pki/ca.crt
        --cgroup-driver=systemd
        \-1282 journalctl -k -f
```

The `--pod-manifest-path` parameter of the kubelet service is specifying the path for the static pods that are started automatically by the kubelet.

- Verify that the control plane processes are started by the kubelet as static pods

```
root@lab_machine $> ssh master1 ls /etc/kubernetes/manifests
etcd.yaml
kube-apiserver.yaml
kube-controller-manager.yaml
kube-scheduler.yaml
root@lab_machine $>
```

- Log in to the master1 node and verify that the control plane services are running under docker.

```

root@master1 $> docker ps --format "table {{.Names}}\n{{.Command}}"
NAMES
COMMAND
k8s_weave-npc_weave-net-b5p71_kube-system_40a2f529-3a4e-11e7-bc73-080027000101_2
"/usr/bin/weave-npc"

k8s_weave_weave-net-b5p71_kube-system_40a2f529-3a4e-11e7-bc73-080027000101_2
"/home/weave/launch.s"

k8s_kube-proxy_kube-proxy-rfsp0_kube-system_759bec85-3a42-11e7-bc73-080027000101_2
"/usr/local/bin/kube-"

k8s_POD_weave-net-b5p71_kube-system_40a2f529-3a4e-11e7-bc73-080027000101_2
"/pause"

k8s_POD_kube-proxy-rfsp0_kube-system_759bec85-3a42-11e7-bc73-080027000101_2
"/pause"

k8s_kube-apiserver_kube-apiserver-master1_kube-system_d5f1bedc4fffb15ee67e786526eef73_2
"kube-apiserver --req"

k8s_POD_kube-apiserver-master1_kube-system_d5f1bedc4fffb15ee67e786526eef73_2
"/pause"

k8s_kube-controller-manager_kube-controller-manager-master1_kube-system_47728d9474eb99_2
"kube-controller-mana"

k8s_etcd_etcd-master1_kube-system_7075157cfd4524dbe0951e00a8e3129e_2
"etcd --listen-client"

k8s_kube-scheduler_kube-scheduler-master1_kube-system_cd96302b5ee69cddc1b533f361310930_2
"kube-scheduler --kub"

k8s_POD_kube-controller-manager-master1_kube-system_47728d9474eb9677641678a27d49ef69_2
"/pause"

k8s_POD_kube-scheduler-master1_kube-system_cd96302b5ee69cddc1b533f361310930_2
"/pause"

k8s_POD_etcd-master1_kube-system_7075157cfd4524dbe0951e00a8e3129e_2
"/pause"
root@master1 $>

```

- Use the `kubectl get pod --namespace=kube-system -o wide` command to verify the pods that are running currently in the kube-system namespace.

Note:

By default, `kubectl` looks for a file named `config` in the `$HOME/.kube` directory. This file is used to configure access to clusters. If that file is missing, and no connection parameters are defined as environmental variables, or parameters of the command line then you will get the error:

```
The connection to the server localhost:8080 was refused - did you specify
the right host or port?
```

```
root@master1 $> cp admin.conf .kube/config
root@master1 $> kubectl get pod --namespace=kube-system -o wide
NAME                                READY STATUS    RESTARTS AGE   IP              NODE
etcd-master1                        1/1   Running    2       28d   10.10.10.51    master1
kube-apiserver-master1              1/1   Running    2       28d   10.10.10.51    master1
kube-controller-manager-master1     1/1   Running    2       28d   10.10.10.51    master1
kube-dns-3913472980-l2xdd           3/3   Running    6       28d   10.44.0.1      worker3
kube-proxy-3w681                    1/1   Running    2       28d   10.10.10.52    worker3
kube-proxy-5qflv                     1/1   Running    2       28d   10.10.10.54    worker3
kube-proxy-d965n                     1/1   Running    2       28d   10.10.10.53    worker2
kube-proxy-rfsp0                     1/1   Running    2       28d   10.10.10.51    master1
kube-scheduler-master1              1/1   Running    2       28d   10.10.10.51    master1
kubernetes-dashboard-2039414953     1/1   Running    1       28d   10.47.0.1      worker2
weave-net-4rb6s                      2/2   Running    5       28d   10.10.10.53    worker2
weave-net-86qfc                      2/2   Running    4       28d   10.10.10.52    worker1
weave-net-b5p7l                      2/2   Running    4       28d   10.10.10.51    master1
weave-net-tlvwm                      2/2   Running    5       28d   10.10.10.54    worker3
root@master1 $>
```

- Use the `kubectl cluster-info` command to get basic information about the API endpoint.

```
root@master1 $> kubectl cluster-info
Kubernetes master is running at https://10.10.10.51:6443
KubeDNS is running at https://10.10.10.51:6443/api/v1/proxy/namespaces/kube-
system/services/kube-dns

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
root@master1 $>
```

- Use the `kubectl get nodes` command to verify the status of the nodes in the cluster

```
root@master1 $> kubectl get nodes
NAME     STATUS    AGE      VERSION
master1  Ready     28d     v1.6.3
worker1  Ready     28d     v1.6.3
worker2  Ready     28d     v1.6.3
worker3  Ready     28d     v1.6.3
root@master1 $>
```

Task 2: Run the first pod

Save the following template in a file, and use it to create the first pod in the cluster (you will find this in your lab2 directory):

```
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu
spec:
  containers:
  - name: ubuntu
    image: ubuntu
    command:
```



```
- sleep
args:
- "3600"
```

- Create a pod using the previous pod manifest file

```
root@master1 $> kubectl create -f lab2/ubuntu.yaml
pod "ubuntu" created
root@master1 $> kubectl get pod -o wide
NAME         READY   STATUS    RESTARTS   AGE   IP           NODE
ubuntu       1/1     Running   0          26s   10.44.0.2    worker3
root@master1 $>
```

You can connect to the node where the pod is running and check it with docker

```
root@master1 $> ssh worker3
Last login: Wed Jun 14 07:56:24 2017 from master1.openstack.local
root@worker3 $> docker ps --format "table {{.Names}} {{.Command}}" | grep ubuntu
k8s_ubudu_ubuntu_default_6c109f3e-50d6-11e7-b33d-080027000101_0 "sleep 3600"
k8s_POD_ubuntu_default_6c109f3e-50d6-11e7-b33d-080027000101_0 "/pause"
root@worker3 $>
```

Task 3: Cleanup

- Delete the pod

```
root@master1 $> kubectl delete pod ubuntu
pod "ubuntu" deleted
root@master1 $>
```

SAMPLE

Lab 3: Accessing the kubernetes API

Task 1: Browse the kubernetes API

- Check the help page of the kubectl command

```
root@master1 $> kubectl --help
kubectl controls the Kubernetes cluster manager.

Find more information at https://github.com/kubernetes/kubernetes.

Basic Commands (Beginner):
  create      Create a resource by filename or stdin
  expose      Take a replication controller, service, deployment or pod and expose
              it as a new Kubernetes Service
  run         Run a particular image on the cluster
  set         Set specific features on objects

Basic Commands (Intermediate):
  get         Display one or many resources
  ...
  proxy      Run a proxy to the Kubernetes API server
  ...

Other Commands:
  api-versions  Print the supported API versions on the server, in the form of
                "group/version"
  config        Modify kubeconfig files
  help         Help about any command
  version      Print the client and server version information

Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options
(applies to all commands).
root@master1 $>
```

- Start the kubectl proxy command to have a proxy to the API

```
root@master1 $> kubectl proxy &
[1] 17409
Starting to serve on 127.0.0.1:8001
root@master1 $>
```

- Use curl to access the API