

Containers: How can I add Kubernetes to an air-gapped Bright Cluster?

Kubernetes is most easily installed on a cluster that is able to access the internet. Some clusters don't have Internet access, but it is still possible to deploy Kubernetes with a few additional steps.

The following recipe was tested with Bright 8.2-12.

Install rpm/deb packages

Ideally a local mirror of the deb/rpm repository is configured. If that is the case, then you can skip to the next chapter. Otherwise the following packages (including their dependencies) should be downloaded and installed, both in the head node, and in all the software images from which Kubernetes is to be deployed:

- cm-docker
- cm-kubernetes-master
- cm-kubernetes-node
- cm-etcd
- nginx
- contrack-tools (on RHEL and Suse) or contrack (on Ubuntu)
- cm-nvidia-docker (optional)

In RHEL systems, you can use yumdownloader for getting those packages and related dependencies.

Get a list of container images

From the head node you can create a file with all the images:

```
[root@82 ~]# sed -n 's/^[ -]*image: */p'  
/cm/local/apps/cm-setup/lib/python2.7/site-packages/cmsetup/plugins/kubernetes/config.yaml >  
images.txt
```

You now have a file with all the images:

```
[root@82 ~]# cat images.txt  
docker.io/coredns/coredns:1.2.4  
k8s.gcr.io/kubernetes-dashboard-amd64:v1.10.0  
k8s.gcr.io/heapster-amd64:v1.5.4  
gcr.io/kubernetes-helm/tiller:v2.11.0  
quay.io/calico/typha:v3.2.3  
quay.io/calico/node:v3.2.3  
quay.io/calico/cni:v3.2.3  
quay.io/kubernetes-ingress-controller/nginx-ingress-controller:0.20.0  
k8s.gcr.io/metrics-server-amd64:v0.3.1  
docker.io/nvidia/k8s-device-plugin:1.11
```

Containers: How can I add Kubernetes to an air-gapped Bright Cluster?

Let us also append the pause image to the file:

```
[root@82 ~]# echo 'k8s.gcr.io/pause:3.1' >> images.txt
```

Now the images.txt file must be copied over to a computer with internet connectivity.

Download the images

From this computer with internet connectivity (and Docker installed), pull all the images:

```
[root@82 ~]# for image in $(cat images.txt); do docker pull $image; done
```

Save them in tar archives (this might take some time):

```
[root@82 ~]# for image in $(cat images.txt); do docker save $image -o ${image//\//_}.tar; done
```

At this point you'll have a list of tar archives:

```
[root@82 ~]# ls *.tar
```

```
docker.io_coredns_coredns:1.2.4.tar
```

```
docker.io_nvidia_k8s-device-plugin:1.11.tar
```

```
gcr.io_kubernetes-helm_tiller:v2.11.0.tar
```

```
k8s.gcr.io_heapster-amd64:v1.5.4.tar
```

```
k8s.gcr.io_kubernetes-dashboard-amd64:v1.10.0.tar
```

```
k8s.gcr.io_metrics-server-amd64:v0.3.1.tar
```

```
k8s.gcr.io_pause:3.1.tar
```

```
quay.io_calico_cni:v3.2.3.tar
```

```
quay.io_calico_node:v3.2.3.tar
```

```
quay.io_calico_typha:v3.2.3.tar
```

```
quay.io_kubernetes-ingress-controller_nginx-ingress-controller:0.20.0.tar
```

Those tar archives have to be copied into the head node of the Bright cluster.

Containers: How can I add Kubernetes to an air-gapped Bright Cluster?

On the head node

Deploy a local Docker registry

Run `cm-docker-registry-setup`, and choose `docker-registry`.

Note: in this article we'll assume the local registry address is `"registry:5000"`.

Provide the images to the local registry

From the head node, load the archives:

```
[root@82 ~]# for archive in *.tar; do docker load -i $archive; done
```

Tag the images:

```
[root@82 ~]# registry_address=registry:5000
```

```
[root@82 ~]# for image in $(cat images.txt); do echo $image | sed "s/docker.io/$registry_address/g;s/k8s.gcr.io/$registry_address/g;s/gcr.io/$registry_address/g;s/quay.io/$registry_address/g;" | xargs docker tag $image; done
```

Push the images to the local registry:

```
[root@82 ~]# for image in $(cat images.txt); do echo $image | sed "s/docker.io/$registry_address/g;s/k8s.gcr.io/$registry_address/g;s/gcr.io/$registry_address/g;s/quay.io/$registry_address/g;" | xargs docker push; done
```

Run the Kubernetes setup wizard

Run the `cm-kubernetes-setup` wizard to create the required configuration file. But don't choose to `"Save & Deploy"`. Instead, choose to `"Save & Quit"`.

Replace the official registries with the local one:

```
[root@82 ~]# sed -i "s/docker.io/$registry_address/g;s/k8s.gcr.io/$registry_address/g;s/gcr.io/$registry_address/g;s/quay.io/$registry_address/g;" cm-kubernetes-setup.conf
```

Also, in the same `.conf` file you should add the last line to the Kubelet section:

```
node:
  kubelet_port: 10250
  options:
    - --volume-stats-agg-period=0
```

Containers: How can I add Kubernetes to an air-gapped Bright Cluster?

- --pod-infra-container-image=registry:5000/pause:3.1

If the packages have been already installed, then let us change this key/value setting in the same file:

```
skip_packages: true
```

Run `cm-kubernetes-setup` with the `-C` option, and wait for the installation to complete:

```
[root@82 ~]# cm-kubernetes-setup -c cm-kubernetes-setup.conf
```

If all is well, Kubernetes gets deployed without issues.

Unique solution ID: #1481

Author: Frank Furter

Last update: 2019-10-17 15:17