Building & Running Kubernetes from Source and Submitting Your First PR

Prerequisites

- A computer with a connection to the Internet
- A GitHub account
- Agree to CNCF code of conduct and sign CLA

Overview of this tutorial

1. Setup a VM or native environment and install CentOS*7
2. Build K8s.
3. Prepare and submit a PR.

Before your first PR, you are required to take a look at Kubernetes Contributors guide:
https://github.com/kubernetes/community/tree/master/contributors/guide

Install Dependencies and Clone Kubernetes

$ sudo yum -y update
$ sudo yum -y install yum-utils device-mapper-persistent-data lvm2 gcc git

Add docker package repo with yum-config-manager and install docker-ce
$ sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
$ sudo yum -y install docker-ce
If you have installed docker 1.13, please remove it first with blow command
$ sudo yum -y remove docker docker-common

Clear up
$ sudo yum -y clean all

Install golang-11
$ curl -O https://dl.google.com/go/go1.11.2.linux-amd64.tar.gz
$ sudo tar -C /usr/local -xzf go1.11.2.linux-amd64.tar.gz
$ rm -f go1.11.2.linux-amd64.tar.gz

Note: you may need to setup proxy to download this package. Also you can download with “wget”

Set up environment via bash profile
$ echo 'export PATH=${PATH}:/usr/local/go/bin' >> ~/.bash_profile
$ echo 'export GOPATH_K8S=${HOME}/go/src/k8s.io/kubernetes' >> ~/.bash_profile
$ echo 'export PATH=${GOPATH_K8S}/third_party/etcd:${PATH}' >> ~/.bash_profile
$ source ~/.bash_profile

Clone kubernetes source code
$ mkdir -p ${GOPATH_K8S}
$ git clone https://github.com/kubernetes/kubernetes ${GOPATH_K8S}
$ cd ${GOPATH_K8S}
$ git remote rename origin upstream

Install etcd

$ hack/install-etcd.sh

Add your user to the docker group, so sudo isn't required for docker commands
$ sudo usermod -a -G docker ${USER}
If it shows: “usermod: group 'docker' does not exist”, then run:
$ sudo groupadd docker
Note: If you are working under proxy, you may need to set proxy for docker

Start the docker daemon

$ sudo systemctl enable docker
$ sudo systemctl start docker
Note: Logout and login again before proceeding, so that docker usermod takes effect.

Build Kubernetes

$ cd ${GOPATH_K8S}
$ git checkout v1.14.0
$ time make quick-release
The success building log output as below:

```
$ time make quick-release
+++ [0401 09:40:34] Verifying Prerequisites....
+++ [0401 09:40:34] Building Docker image kube-build:build-c0383442df-5-v1.12.1-2
```

```
+++ [0401 09:40:37] Syncing sources to container
+++ [0401 09:40:41] Running build command...
+++ [0401 09:41:19] Building go targets for linux/amd64:
    cmd/kube-proxy
    cmd/kube-apiserver
    cmd/kube-controller-manager
    cmd/cloud-controller-manager
    cmd/kubelet
    cmd/kubeadm
    cmd/hyperkube
    cmd/kube-scheduler
    vendor/k8s.io/apiextensions-apiserver
    cluster/gce/gci/mounter
+++ [0401 09:42:18] Building go targets for linux/amd64:
    cmd/kube-proxy
    cmd/kubeadm
    cmd/kubelet
+++ [0401 09:42:47] Building go targets for linux/amd64:
    cmd/kubectl
+++ [0401 09:43:05] Building go targets for linux/amd64:
    cmd/gendocs
    cmd/genkubedocs
    cmd/genman
    cmd/genyaml
    cmd/genswaggertypedocs
    cmd/linkcheck
    vendor/github.com/onsi/ginkgo/ginkgo
    test/e2e/e2e.test
+++ [0401 09:43:52] Building go targets for linux/amd64:
    cmd/kubemark
    vendor/github.com/onsi/ginkgo/ginkgo
    test/e2e_node/e2e_node.test
+++ [0401 09:44:27] Syncing out of container
+++ [0401 09:44:37] Building tarball: manifests
+++ [0401 09:44:37] Building tarball: src
+++ [0401 09:44:37] Starting tarball: client linux-amd64
+++ [0401 09:44:37] Waiting on tarballs
+++ [0401 09:44:42] Building tarball: node linux-amd64
+++ [0401 09:44:42] Building images: linux-amd64
+++ [0401 09:44:42] Starting docker build for image: cloud-controller-manager-amd64
+++ [0401 09:44:42] Starting docker build for image: kube-apiserver-amd64
+++ [0401 09:44:42] Starting docker build for image: kube-controller-manager-amd64
+++ [0401 09:44:42] Starting docker build for image: kube-scheduler-amd64
+++ [0401 09:44:42] Starting docker build for image: kube-proxy-amd64
+++ [0401 09:44:42] Building hyperkube image for arch: amd64
+++ [0401 09:44:42] Building conformance image for arch: amd64
+++ [0401 09:44:46] Deleting docker image k8s.gcr.io/kube-scheduler:v1.14.0
+++ [0401 09:44:46] Deleting docker image k8s.gcr.io/kube-proxy:v1.14.0
+++ [0401 09:44:47] Deleting docker image k8s.gcr.io/cloud-controller-manager:v1.14.0
+++ [0401 09:44:47] Deleting docker image k8s.gcr.io/kube-controller-manager:v1.14.0
+++ [0401 09:44:49] Deleting docker image k8s.gcr.io/kube-apiserver:v1.14.0
+++ [0401 09:44:55] Deleting hyperkube image k8s.gcr.io/hyperkube-amd64:v1.14.0
+++ [0401 09:44:58] Deleting conformance image k8s.gcr.io/conformance-amd64:v1.14.0
+++ [0401 09:44:58] Docker builds done
+++ [0401 09:44:58] Building tarball: server linux-amd64
+++ [0401 09:45:59] Building tarball: final
+++ [0401 09:45:59] Starting tarball: test linux-amd64
```

```
+++ [0401 09:45:59] Waiting on test tarballs
+++ [0401 09:46:32] Building tarball: test portable
+++ [0401 09:46:33] Building tarball: test mondo (deprecated by KEP
sig-testing/20190118-breaking-apart-the-kubernetes-test-tarball)
real    6m32.281s
user    2m37.769s
sys     0m22.010s
```

Run a Local Kubernetes Cluster

Start the cluster:

$ ${GOPATH_K8S}/hack/local-up-cluster.sh

The success up log output as below:

```
$ ${GOPATH_K8S}/hack/local-up-cluster.sh
WARNING : The kubelet is configured to not fail even if swap is enabled; production deployments should
disable swap.
WARNING : This script MAY be run as root for docker socket / iptables functionality; if failures occur,
retry as root.
make: Entering directory `/home/ailin/go/src/k8s.io/kubernetes'
make[1]: Entering directory `/home/ailin/go/src/k8s.io/kubernetes'
make[1]: Leaving directory `/home/ailin/go/src/k8s.io/kubernetes'
+++ [0402 11:20:44] Building go targets for linux/amd64:
    cmd/kubectl
    cmd/hyperkube
make: Leaving directory `/home/ailin/go/src/k8s.io/kubernetes'
Kubelet cgroup driver defaulted to use: cgroupfs
API SERVER insecure port is free, proceeding...
API SERVER secure port is free, proceeding...
Unable to successfully run 'cfssl' from
/home/ailin/go/src/k8s.io/kubernetes/third_party/etcd:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/h
ome/ailin/.local/bin:/home/ailin/bin:/usr/local/go/bin; downloading instead...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  9.8M  100  9.8M    0     0   571k      0  0:00:17  0:00:17 --:--:--  679k
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2224k  100 2224k    0     0   589k      0  0:00:03  0:00:03 --:--:--  589k
Detected host and ready to start services.  Doing some housekeeping first...
Using GO_OUT /home/ailin/go/src/k8s.io/kubernetes/_output/local/bin/linux/amd64
Starting services now!
Starting etcd
etcd --advertise-client-urls http://127.0.0.1:2379 --data-dir /tmp/tmp.7hgTjQll8J --listen-client-urls
http://127.0.0.1:2379 --debug > "/tmp/etcd.log" 2>/dev/null
Waiting for etcd to come up.
+++ [0402 11:21:30] On try 2, etcd: : http://127.0.0.1:2379
{"action":"set","node":{"key":"/_test","value":"","modifiedIndex":4,"createdIndex":4}}
Generating a 2048 bit RSA private key
...............................+++
.....................................................................................................
..............................................+++
writing new private key to '/var/run/kubernetes/server-ca.key'
-----
Generating a 2048 bit RSA private key
..+++
```

```
.............................................+++
writing new private key to '/var/run/kubernetes/client-ca.key'
-----
Generating a 2048 bit RSA private key
.........................................................+++
.......................+++
writing new private key to '/var/run/kubernetes/request-header-ca.key'
-----
2019/04/02 11:21:31 [INFO] generate received request
2019/04/02 11:21:31 [INFO] received CSR
2019/04/02 11:21:31 [INFO] generating key: rsa-2048
2019/04/02 11:21:31 [INFO] encoded CSR
2019/04/02 11:21:31 [INFO] signed certificate with serial number
323909132435090302509666474914791334919001602 39
2019/04/02 11:21:31 [WARNING] This certificate lacks a "hosts" field. This makes it unsuitable for
websites. For more information see the Baseline Requirements for the Issuance and Management
of Publicly-Trusted Certificates, v.1.1.6, from the CA/Browser Forum (https://cabforum.org);
specifically, section 10.2.3 ("Information Requirements").
2019/04/02 11:21:31 [INFO] generate received request
2019/04/02 11:21:31 [INFO] received CSR
2019/04/02 11:21:31 [INFO] generating key: rsa-2048
2019/04/02 11:21:31 [INFO] encoded CSR
2019/04/02 11:21:31 [INFO] signed certificate with serial number
394590602134259875435334910338020016025792024535
2019/04/02 11:21:31 [WARNING] This certificate lacks a "hosts" field. This makes it unsuitable for
websites. For more information see the Baseline Requirements for the Issuance and Management
of Publicly-Trusted Certificates, v.1.1.6, from the CA/Browser Forum (https://cabforum.org);
specifically, section 10.2.3 ("Information Requirements").
2019/04/02 11:21:31 [INFO] generate received request
2019/04/02 11:21:31 [INFO] received CSR
2019/04/02 11:21:31 [INFO] generating key: rsa-2048
2019/04/02 11:21:32 [INFO] encoded CSR
2019/04/02 11:21:32 [INFO] signed certificate with serial number
291820860400025538403722325173160877042764658303
2019/04/02 11:21:32 [WARNING] This certificate lacks a "hosts" field. This makes it unsuitable for
websites. For more information see the Baseline Requirements for the Issuance and Management
of Publicly-Trusted Certificates, v.1.1.6, from the CA/Browser Forum (https://cabforum.org);
specifically, section 10.2.3 ("Information Requirements").
2019/04/02 11:21:32 [INFO] generate received request
2019/04/02 11:21:32 [INFO] received CSR
2019/04/02 11:21:32 [INFO] generating key: rsa-2048
2019/04/02 11:21:32 [INFO] encoded CSR
2019/04/02 11:21:32 [INFO] signed certificate with serial number
552189440632511275132246651627588962272581658958
2019/04/02 11:21:32 [WARNING] This certificate lacks a "hosts" field. This makes it unsuitable for
websites. For more information see the Baseline Requirements for the Issuance and Management
of Publicly-Trusted Certificates, v.1.1.6, from the CA/Browser Forum (https://cabforum.org);
specifically, section 10.2.3 ("Information Requirements").
2019/04/02 11:21:32 [INFO] generate received request
2019/04/02 11:21:32 [INFO] received CSR
2019/04/02 11:21:32 [INFO] generating key: rsa-2048
2019/04/02 11:21:32 [INFO] encoded CSR
2019/04/02 11:21:32 [INFO] signed certificate with serial number
479350650513368594243782437975905808839306291976
2019/04/02 11:21:32 [WARNING] This certificate lacks a "hosts" field. This makes it unsuitable for
websites. For more information see the Baseline Requirements for the Issuance and Management
of Publicly-Trusted Certificates, v.1.1.6, from the CA/Browser Forum (https://cabforum.org);
```

```
specifically, section 10.2.3 ("Information Requirements").
2019/04/02 11:21:32 [INFO] generate received request
2019/04/02 11:21:32 [INFO] received CSR
2019/04/02 11:21:32 [INFO] generating key: rsa-2048
2019/04/02 11:21:32 [INFO] encoded CSR
2019/04/02 11:21:32 [INFO] signed certificate with serial number
277435434919703219302785919829375529549318602l0
2019/04/02 11:21:32 [WARNING] This certificate lacks a "hosts" field. This makes it unsuitable for
websites. For more information see the Baseline Requirements for the Issuance and Management
of Publicly-Trusted Certificates, v.1.1.6, from the CA/Browser Forum (https://cabforum.org);
specifically, section 10.2.3 ("Information Requirements").
2019/04/02 11:21:32 [INFO] generate received request
2019/04/02 11:21:32 [INFO] received CSR
2019/04/02 11:21:32 [INFO] generating key: rsa-2048
2019/04/02 11:21:32 [INFO] encoded CSR
2019/04/02 11:21:32 [INFO] signed certificate with serial number
200968592605692839808947886684478493630912789214
2019/04/02 11:21:32 [WARNING] This certificate lacks a "hosts" field. This makes it unsuitable for
websites. For more information see the Baseline Requirements for the Issuance and Management
of Publicly-Trusted Certificates, v.1.1.6, from the CA/Browser Forum (https://cabforum.org);
specifically, section 10.2.3 ("Information Requirements").
2019/04/02 11:21:32 [INFO] generate received request
2019/04/02 11:21:32 [INFO] received CSR
2019/04/02 11:21:32 [INFO] generating key: rsa-2048
2019/04/02 11:21:33 [INFO] encoded CSR
2019/04/02 11:21:33 [INFO] signed certificate with serial number
156663094996261224694210908404055817279418285 59
2019/04/02 11:21:33 [WARNING] This certificate lacks a "hosts" field. This makes it unsuitable for
websites. For more information see the Baseline Requirements for the Issuance and Management
of Publicly-Trusted Certificates, v.1.1.6, from the CA/Browser Forum (https://cabforum.org);
specifically, section 10.2.3 ("Information Requirements").
2019/04/02 11:21:33 [INFO] generate received request
2019/04/02 11:21:33 [INFO] received CSR
2019/04/02 11:21:33 [INFO] generating key: rsa-2048
2019/04/02 11:21:33 [INFO] encoded CSR
2019/04/02 11:21:33 [INFO] signed certificate with serial number
235874742245530600785332564631578946148065380 21
2019/04/02 11:21:33 [WARNING] This certificate lacks a "hosts" field. This makes it unsuitable for
websites. For more information see the Baseline Requirements for the Issuance and Management
of Publicly-Trusted Certificates, v.1.1.6, from the CA/Browser Forum (https://cabforum.org);
specifically, section 10.2.3 ("Information Requirements").
Waiting for apiserver to come up
+++ [0402 11:21:42] On try 7, apiserver: : ok
Cluster "local-up-cluster" set.
use 'kubectl --kubeconfig=/var/run/kubernetes/admin-kube-aggregator.kubeconfig' to use the aggregated API
server
service/kube-dns created
serviceaccount/kube-dns created
configmap/kube-dns created
deployment.extensions/kube-dns created
Kube-dns addon successfully deployed.
kubelet ( 28821 ) is running.
Create default storage class for
storageclass.storage.k8s.io/standard created
Local Kubernetes cluster is running. Press Ctrl-C to shut it down.

Logs:
```

```
  /tmp/kube-apiserver.log
  /tmp/kube-controller-manager.log

  /tmp/kube-proxy.log
  /tmp/kube-scheduler.log
  /tmp/kubelet.log

To start using your cluster, you can open up another terminal/tab and run:

  export KUBECONFIG=/var/run/kubernetes/admin.kubeconfig
  cluster/kubectl.sh

Alternatively, you can write to the default kubeconfig:

  export KUBERNETES_PROVIDER=local

  cluster/kubectl.sh config set-cluster local --server=https://localhost:6443
--certificate-authority=/var/run/kubernetes/server-ca.crt
  cluster/kubectl.sh config set-credentials myself --client-key=/var/run/kubernetes/client-admin.key
--client-certificate=/var/run/kubernetes/client-admin.crt
  cluster/kubectl.sh config set-context local --cluster=local --user=myself
  cluster/kubectl.sh config use-context local
  cluster/kubectl.sh
```

Note: Local Kubernetes cluster is running. Press Ctrl-C to shut it down.

Verify the cluster is running in another ssh window

$ export KUBERNETES_PROVIDER=local
$ export KUBECONFIG=/var/run/kubernetes/admin.kubeconfig
$ ${GOPATH_K8S}/cluster/kubectl.sh get nodes

```
$ ${GOPATH_K8S}/cluster/kubectl.sh get nodes
NAME        STATUS   ROLES    AGE   VERSION
127.0.0.1   Ready    <none>   85s   v1.12.3-dirty
```

Contributing to Open-Source Kubernetes

Kubernetes Community Prerequisites

If you wish to send a PR, please read and agree to the Code of Conduct:
https://github.com/cncf/foundation/blob/master/code-of-conduct.md
and sign the Contributor License Agreement (CLA)
Please follow up below instructions:
Instructions: https://github.com/kubernetes/community/blob/master/CLA.md

- Sign up to contribute to this project as an individual
- Sign up to contribute to this project as an employee

sign the Contributor License Agreement(CLA), Intel* employee is recommended to sign as “sign up to contribute to this project as an employee”.
https://identity.linuxfoundation.org/projects/cncf
Here you need to input account info and email address and request a new Linux Foundation ID, then you will receive an email to confirm your Linux Foundation ID,
If you have had a github.com account, then A third-party OAuth application (Linux Foundation ID) with user:email scope will be authorized to access your account. Then you will receive an github email as below:
[GitHub] A third-party OAuth application has been added to your account

Fork Kubernetes on GitHub

1. Navigate to https://github.com/kubernetes/kubernetes
2. Click the "Fork" button in the upper-right corner and follow the on-screen instructions.
Then it takes time focking source project to your github.com account:
For example: https://github.com/xxxx/kubernetes

Setup GitHub username

To setup a GitHub account, You will need to provide [YOUR_GITHUB_USER_NAME].
You can export it as a bash variable, and add it to your bash profile.
$ export GITHUB_USER=[INSERT_YOUR_GITHUB_USER_NAME]
$ echo 'export GITHUB_USER=${GITHUB_USER}' >> ~/.bash_profile

Change the Kubernetes Source Code

Navigate to the directory containing your clone of Kubernetes and add your fork:
$ cd ${GOPATH_K8S}
$ git remote add origin https://github.com/${GITHUB_USER}/kubernetes
$ git fetch origin

These instructions make the following assumptions:
- Your `upstream` remote points to `https://github.com/kubernetes/kubernetes`.
- Your `origin` remote points to `https://github.com/${GITHUB_USER}/kubernetes`.
You can double-check by running:
$ git remote -v
Create a branch before making a change:
$ git checkout upstream/master
$ git checkout -b my-branch-of-k8s

If you would like to make a change, we recommend finding some untested code and adding a few unit tests. If you don't know how to write unit tests in Go, read this: https://golang.org/doc/code.html

Run Kubernetes Unit Tests Locally

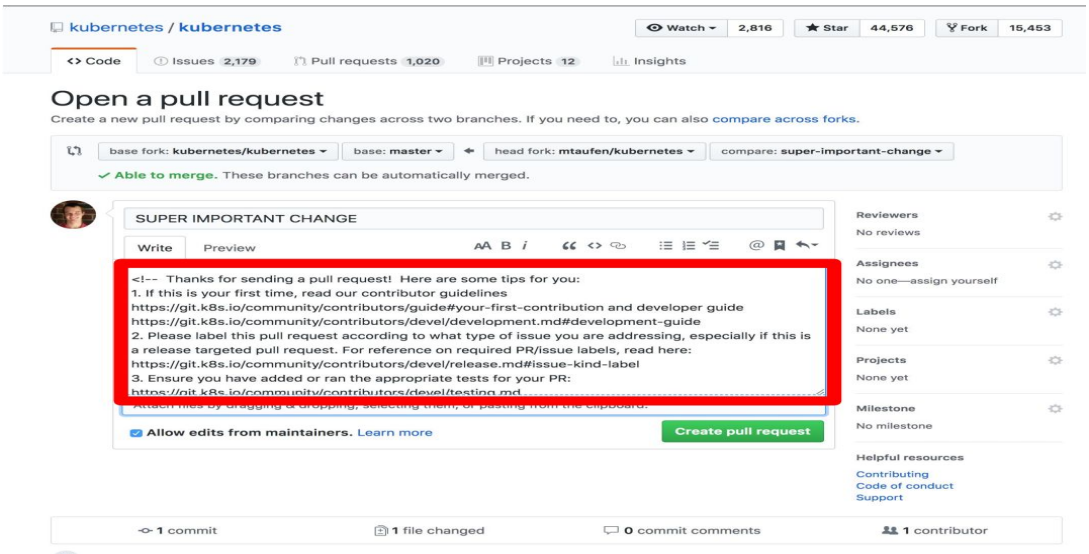
$ make test WHAT=k8s.io/kubernetes/pkg/util/normalizer

Contribute a Pull-Request

Commit your changes and push them to your fork:
$ gofmt -w [INSERT PATH TO CHANGED FILES]
$ gofmt -w pkg/util/normalizer/normalizer_test.go
$ git config --global user.name "[Your Name]"
$ git config --global user.email "[youremail@example.com]
$ git add -A git commit -m "[Commit message here]"
$ git push -u origin/my-branch-of-k8s

Create a pull request via the GitHub UI

1. Navigate to https://github.com/kubernetes/kubernetes
2. Click the "New Pull Request" button
3. Click the "compare across forks" link
4. Select your fork and branch from the two dropdowns on the right
5. Click the "Create pull request" button
6. Follow the instructions in the PR template (example below)



Note that an existing maintainer must comment with the `/ok-to-test` command to initiate the automated tests for new contributors.
Example PR template:
What type of PR is this?
> Uncomment only one, leave it on its own line: > > /kind api-change > /kind bug > /kind cleanup > /kind design > /kind documentation > /kind failing-test > /kind feature > /kind flake
What this PR does / why we need it?
Which issue(s) this PR fixes?
(optional, in `fixes #(, fixes #, ...)` format, will close the issue(s) when PR gets merged):
Fixes # **Special notes for your reviewer**:
Does this PR introduce a user-facing change?