|  | gUIDance for window  1. Install VirtualBox   Select window hosts <https://www.virtualbox.org/wiki/Downloads>   1. Install Vagrant   Select window 64bit <https://www.vagrantup.com/downloads.html>   1. Setup virtual environment   Go to any directory, save the file below to the directory    Run command  *vagrant init*  It will create a Vagrantfile, replace the generated file with above file, then run  *vagrant up*  you should have three virtual environments named master, worker1 and worker2 up  *vagrant status*   1. Open window PowerShell, run   *vagrant ssh master*  if your see message -- [vagrant@127.0.0.1](mailto:vagrant@127.0.0.1): Permission denied(publickey), run  *$Env:VAGRANT\_PREFER\_SYSTEM\_BIN += 0*   1. Install Kubernetes   *sudo -s*  *curl -s* [*https://packages.cloud.google.com/apt/doc/apt-key.gpg*](https://packages.cloud.google.com/apt/doc/apt-key.gpg) *| apt-key add -*  *cat <<EOF >/etc/apt/sources.list.d/kubernetes.list*  *deb* [*http://apt.kubernetes.io/*](http://apt.kubernetes.io/) *kubernetes-xenial main*  *EOF*  *apt-get update*  *apt-get install -y kubelet kubeadm kubectl*  *kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=192.168.33.10*  After *init* finish, at last line of the finished output screen, copy whole line to a file for node join, the line start with kubeadm *join ---token xxx*………….  Continue in master, exit from root user  *mkdir -p $HOME/.kube*  *sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config*  *sudo chown $(id -u):$(id -g) $HOME/.kube/config*  *sudo -s*  *kubectl apply -f* [*https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml*](https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml)   1. Go to worker 1   *vagrant ssh worker1*  *sudo -s*  *curl -s* [*https://packages.cloud.google.com/apt/doc/apt-key.gpg*](https://packages.cloud.google.com/apt/doc/apt-key.gpg) *| apt-key add –*  *cat <<EOF >/etc/apt/sources.list.d/kubernetes.list*  *deb* [*http://apt.kubernetes.io/*](http://apt.kubernetes.io/) *kubernetes-xenial main*  *EOF*  *apt-get update*  *apt-get install -y kubelet kubeadm kubectl*  *kubeadm join 192.168.33.10:6443 --token 70oz7i.52312hgl37dj7n1d --discovery-token-ca-cert-hash sha256:6851845c665bc964c9c8679f665a8e110c9e610391a1ac62a3a1fed5caf0d2b9*   1. Go to worker 2, follow the exact steps of worker 1 2. once you are done, go to master, issue a command   *kubectl get nodes*  You should be able to see three nodes. Congratulation !! you are done, have a couple of tea, enjoy **k8s.**   1. Install kubernetes dashboard   kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/recommended/kubernetes-dashboard.yaml  curl <https://gist.githubusercontent.com/chukaofili/9e94d966e73566eba5abdca7ccb067e6/raw/0f17cd37d2932fb4c3a2e7f4434d08bc64432090/k8s-dashboard-admin-user.yaml> > k8s-dashboard-admin-user.yaml  kubectl apply –f k8s-dashboard-admin-user.yaml  kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep admin-user | awk '{print $1}')  eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2VyLXRva2VuLXRuNnhkIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkbWluLXVzZXIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiJiMTAyMWQ4Mi0yMDI3LTExZTktODBlYi0wMjBkMzExNGM3YTciLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6a3ViZS1zeXN0ZW06YWRtaW4tdXNlciJ9.m7vrvqUrZdZtuPJc-T7Cxbppohxkvfm7jz9saqWOH86d8u8amvBTHikx8xppWrGvefWLd1y3w17x3b8Uiu5lqNAT3RqsekqXFMgqOot4HJVU3QJMyVRJbDuKKnh0nMvLoa1cgoMyZwtg2uKzAXU7IyUjr5gpxkmJS7xU1B0aNJtUtnxEbIRD\_fNkfv\_4af3KNNCuGXDoTxY6lJdW00vTjZC0\_Jm45lGVxXoWgOKqHKE4tkrlQ8dbnRbk9mQAh1fJAIS46\_Wov9DZw-9T1icJ5dy-eeiWIi\_N3S7N8hBJhrqSEkpzBGNLIGBVctDonlFR0sQ5Cvi-O-eVtFcz0S1twQ  In the Chrome address bar type: chrome://flags/#allow-insecure-localhost  Type: htts://<installed-node-address>:port  If Chrome doesn’t work, try Firefox, should work after add exception   1. Add above generated token to dashboard ui. 2. Install docker image BigDataUI   Create Dockerfile       1. Copy BigdataUI angular files to shared dir, ssh to vagrant master, got to shared directory, run   docker build –t bigdataui .   1. Login to docker   Docker login –username=charleslicai –email=charleslicai@gmail.com   1. Tag image   Docker tag xxxx charleslicai/bigdataui:firsttry   1. Push to docker   Docker push charleslicai@bigdataui   1. sfsdf |
| --- | --- |
|  |  |