Member-only story

Guide: OKD 4.5 Single Node Cluster

Craig Robinson · Follow

Published in The Startup · 12 min read · Jul 15, 2020

Updated: 7/29/2020

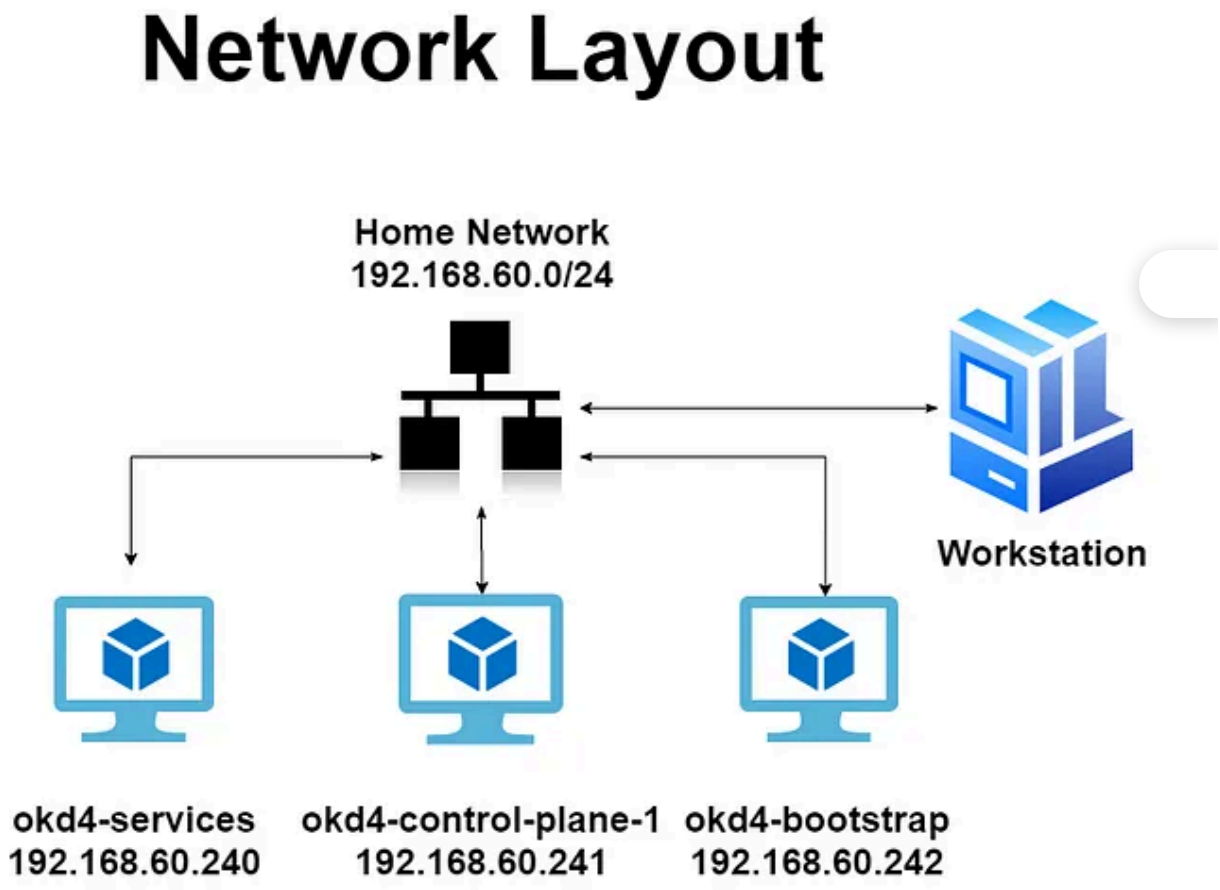
After listening to some feedback in the chat on a recent Twitch stream and the okd-wg mailing list I decided to create a guide for installing an OKD 4.5 SNC (single node cluster). This guide will use no worker nodes and only a single control-plane node with the goal of reducing the total amount of resources needed to get OKD up and running for a test drive.

VM Overview:

For this installation, I used an ESXi 6.5 host on my local network and allocated the total virtual machine RAM usage to 22GB. ESXi 6.5 has a minimum of 4GB of RAM but you should be able to run this on an ESXi host

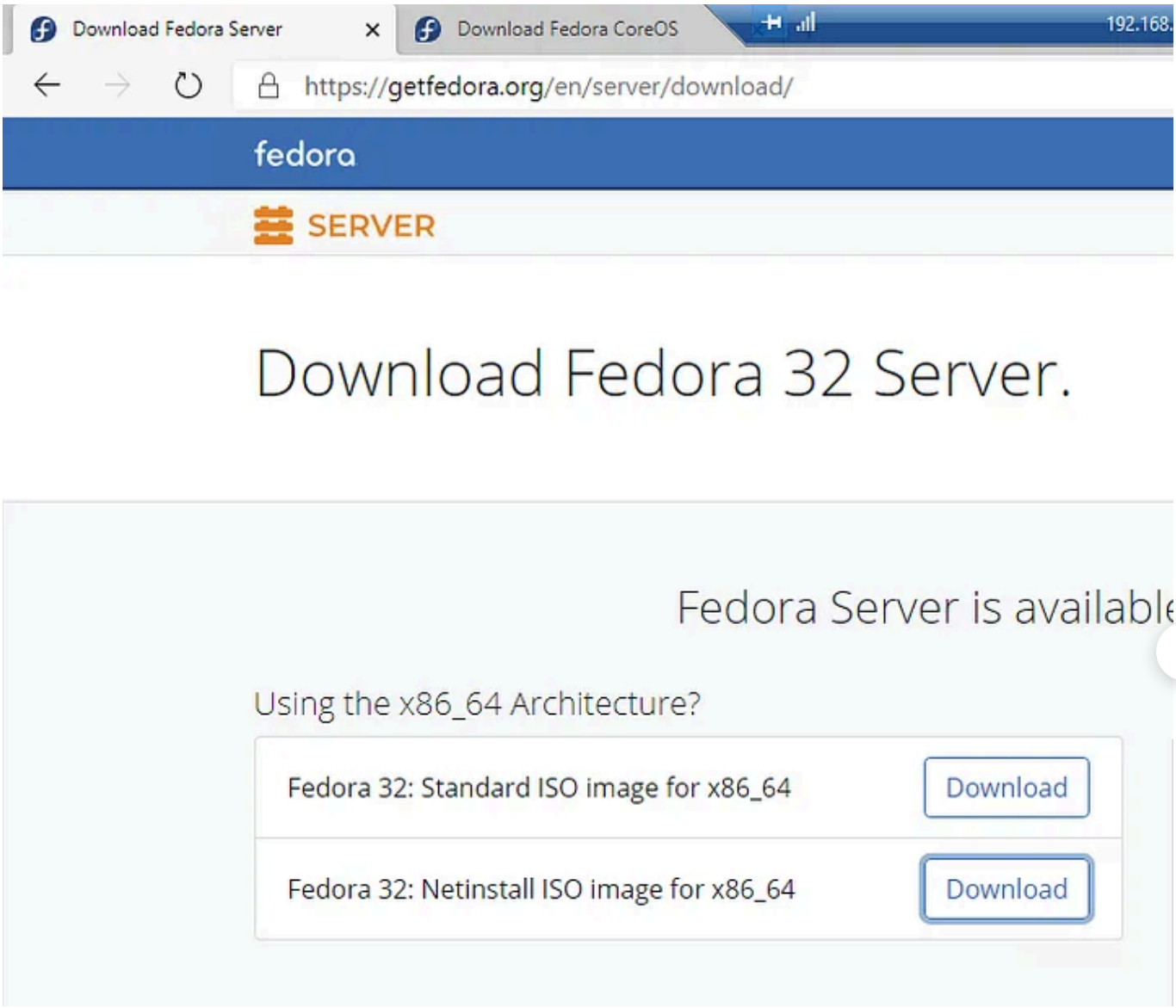
with 24GB of RAM. Here is a breakdown of the virtual machines and network layout:

Machine	Type	OS	vCPU	RAM	Storage	IP Address
okd4-services	DNS/LB/Web/NFS	Fedora Server	2	2	100	192.168.60.240
okd4-control-plane-1	Master	Fedora CoreOS	8	16	120	192.168.60.241
okd4-boostrap	Bootstrap	Fedora CoreOS	4	4	120	192.168.60.242

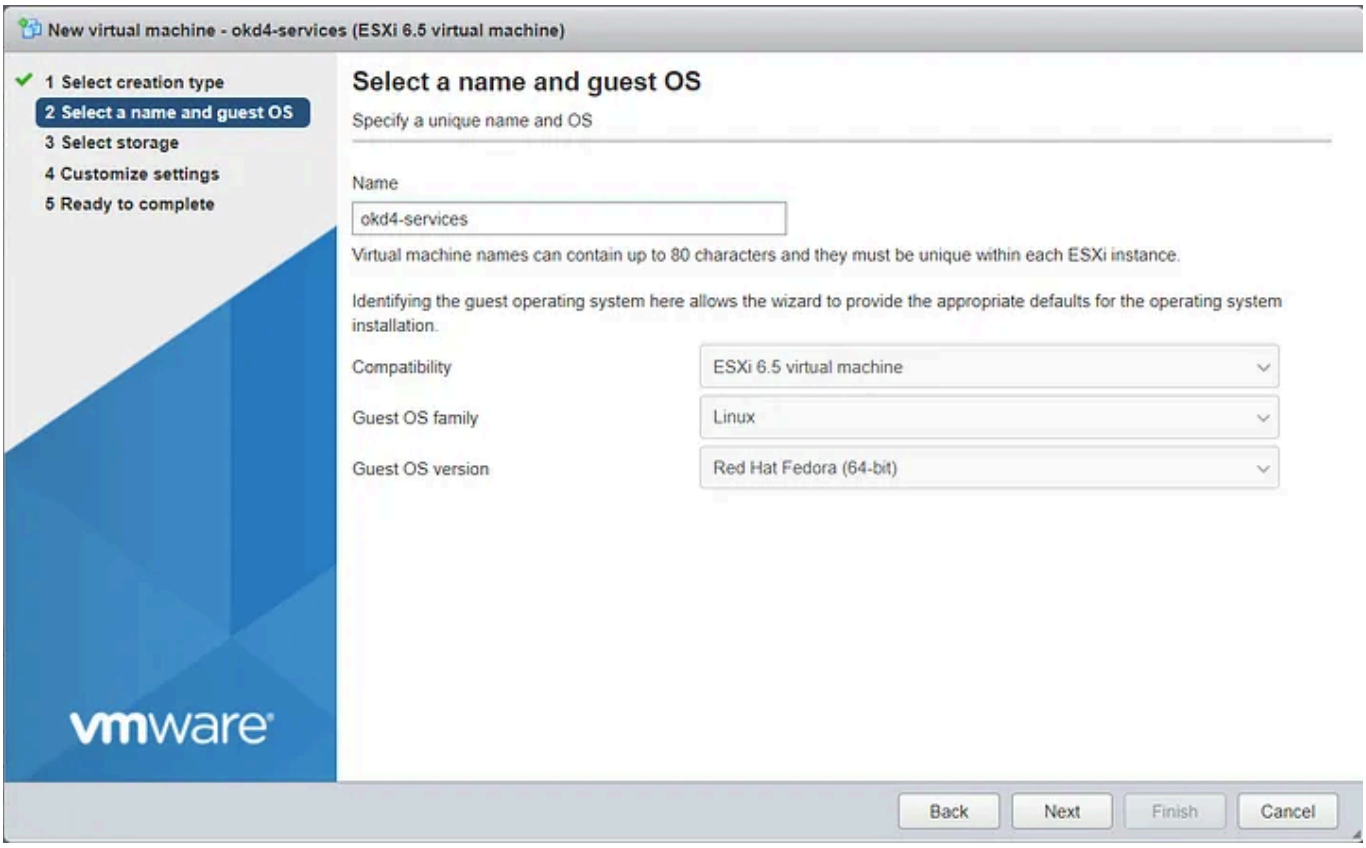


Create the okd4-services VM:

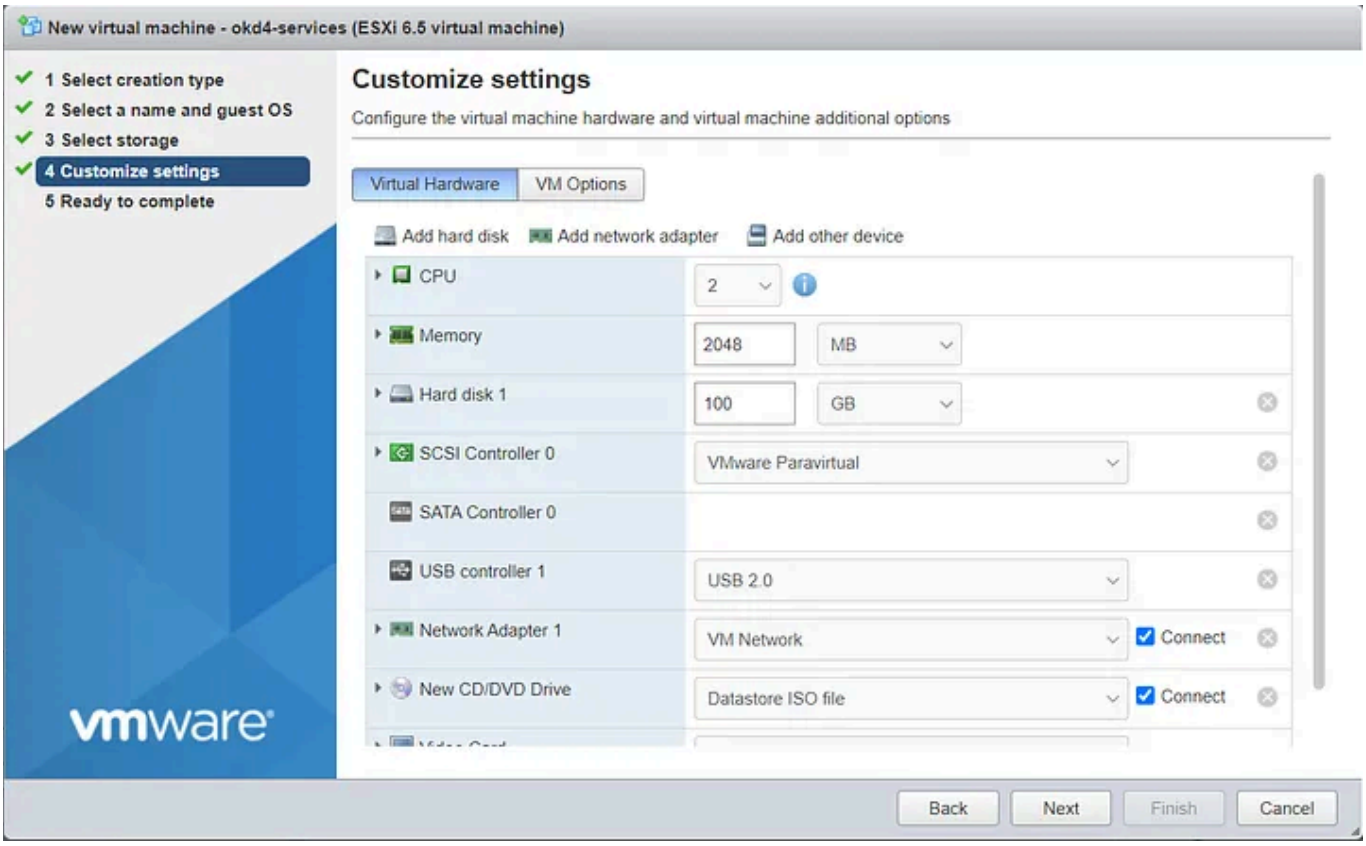
Download the Fedora Server Standard ISO and upload it to your ESXi host datastore.



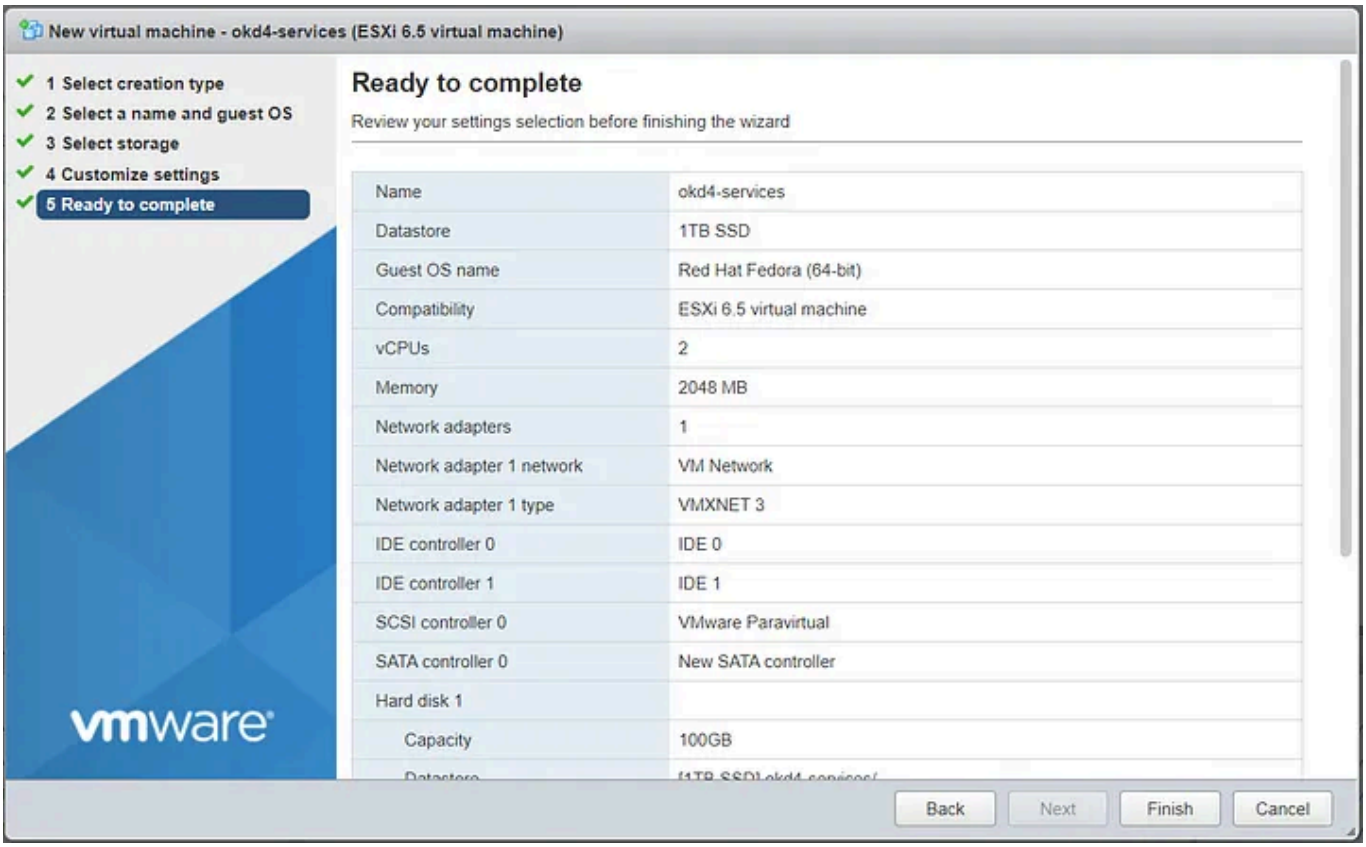
Create a new Virtual Machine. Choose Guest OS as *Linux* and Select CentOS 7 (64-bit).



Select the Datastore and customize the settings to 2 vCPU, 2GB RAM, 100GB HD. Attach the Fedora Serer ISO to the CD/DVD drive.

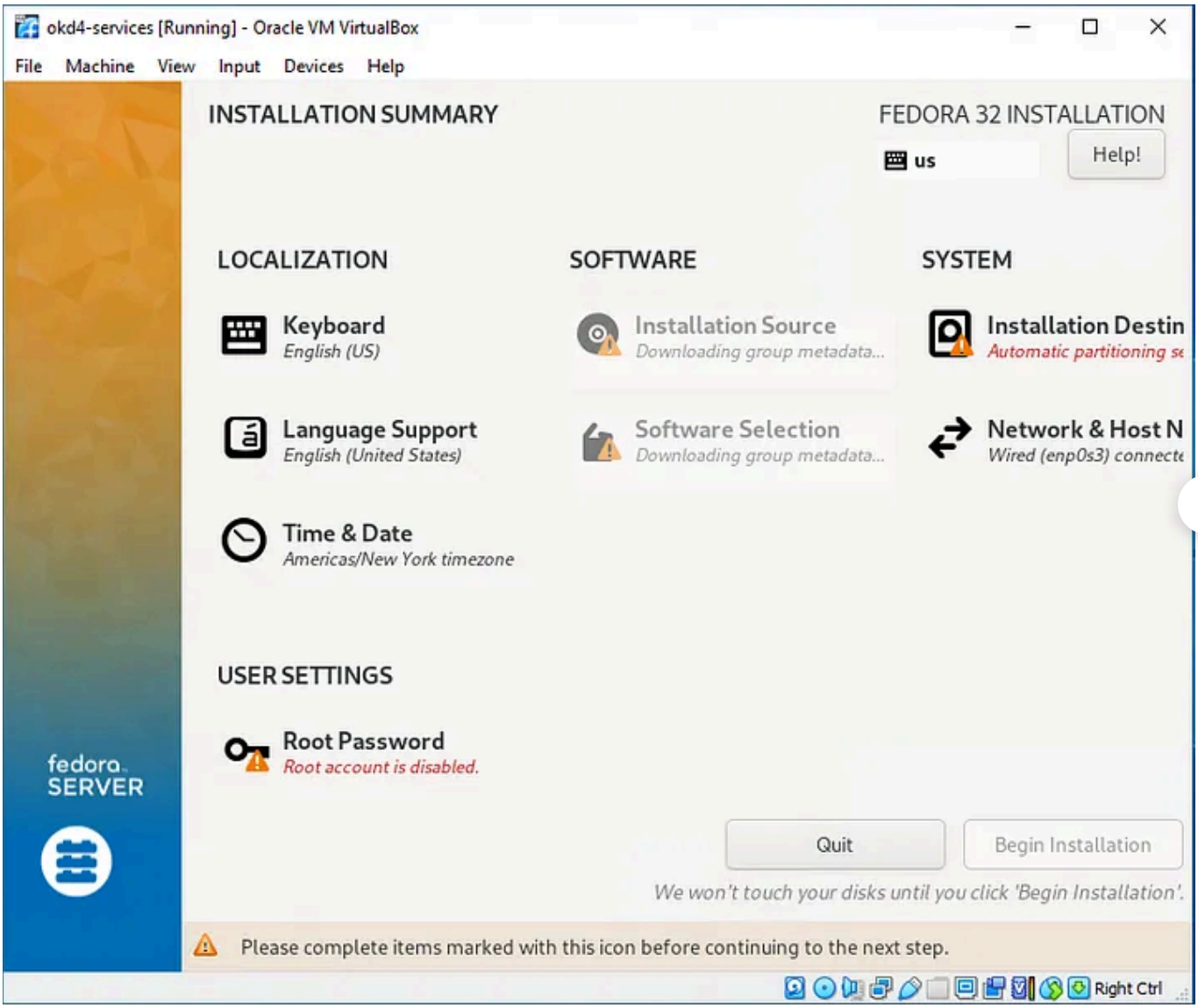


Review your settings and click Finish.



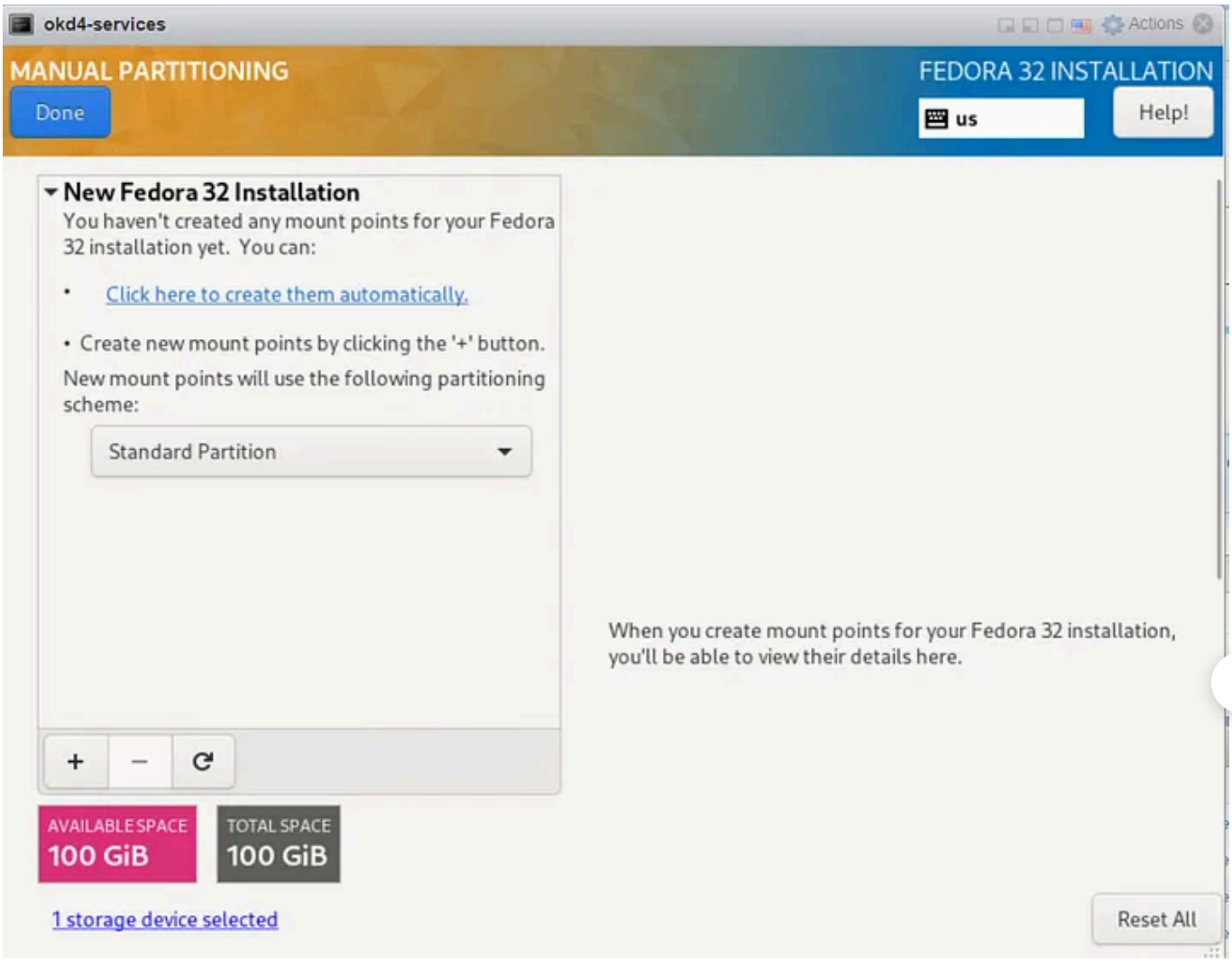
Install Fedora Server 32 on the okd4-services VM:

Run through the installation.

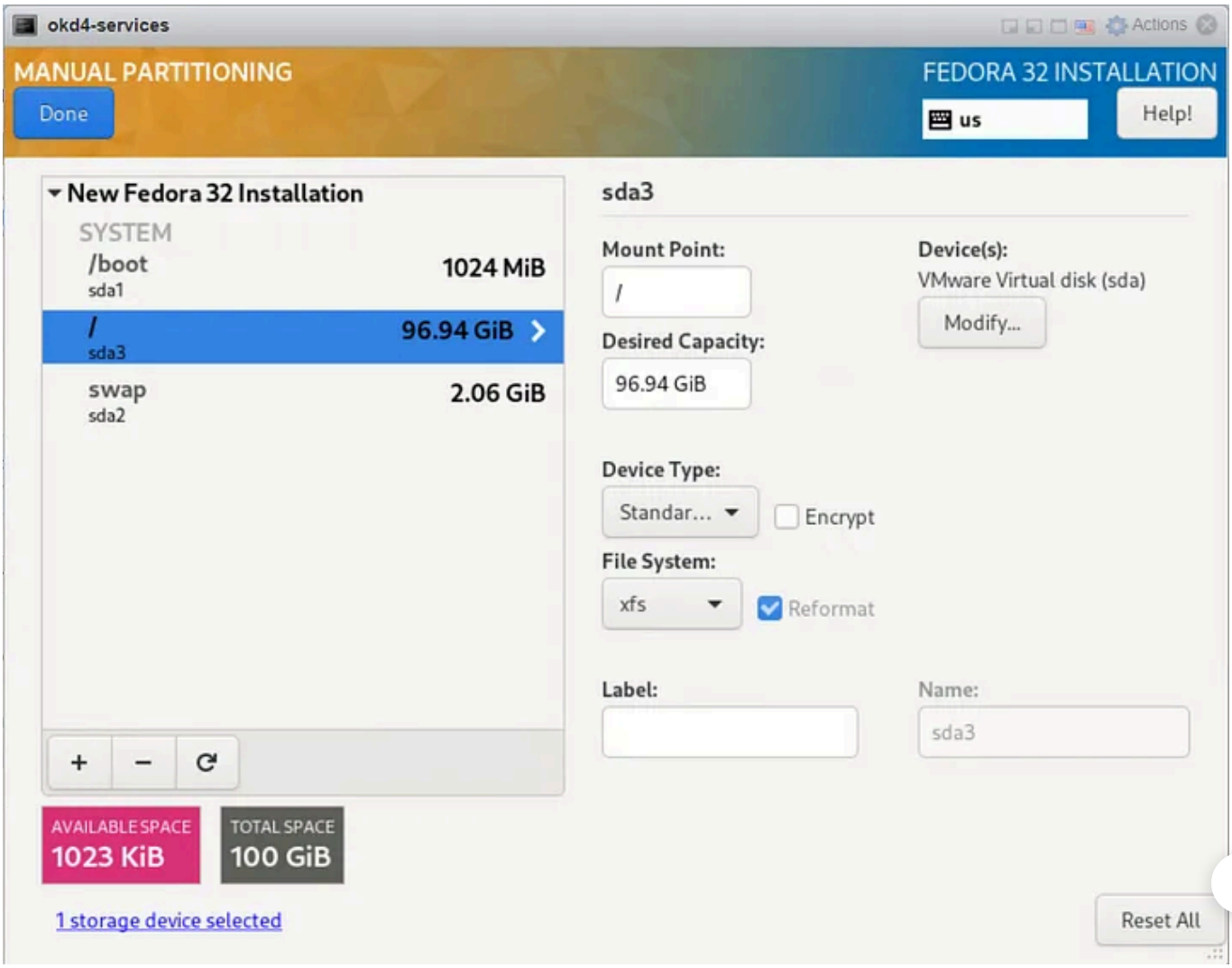


I prefer to use the “Standard Partition” storage configuration to use the entire drive at “/”. On the “Installation Destination” page, click on Custom under Storage Configuration, then Done.

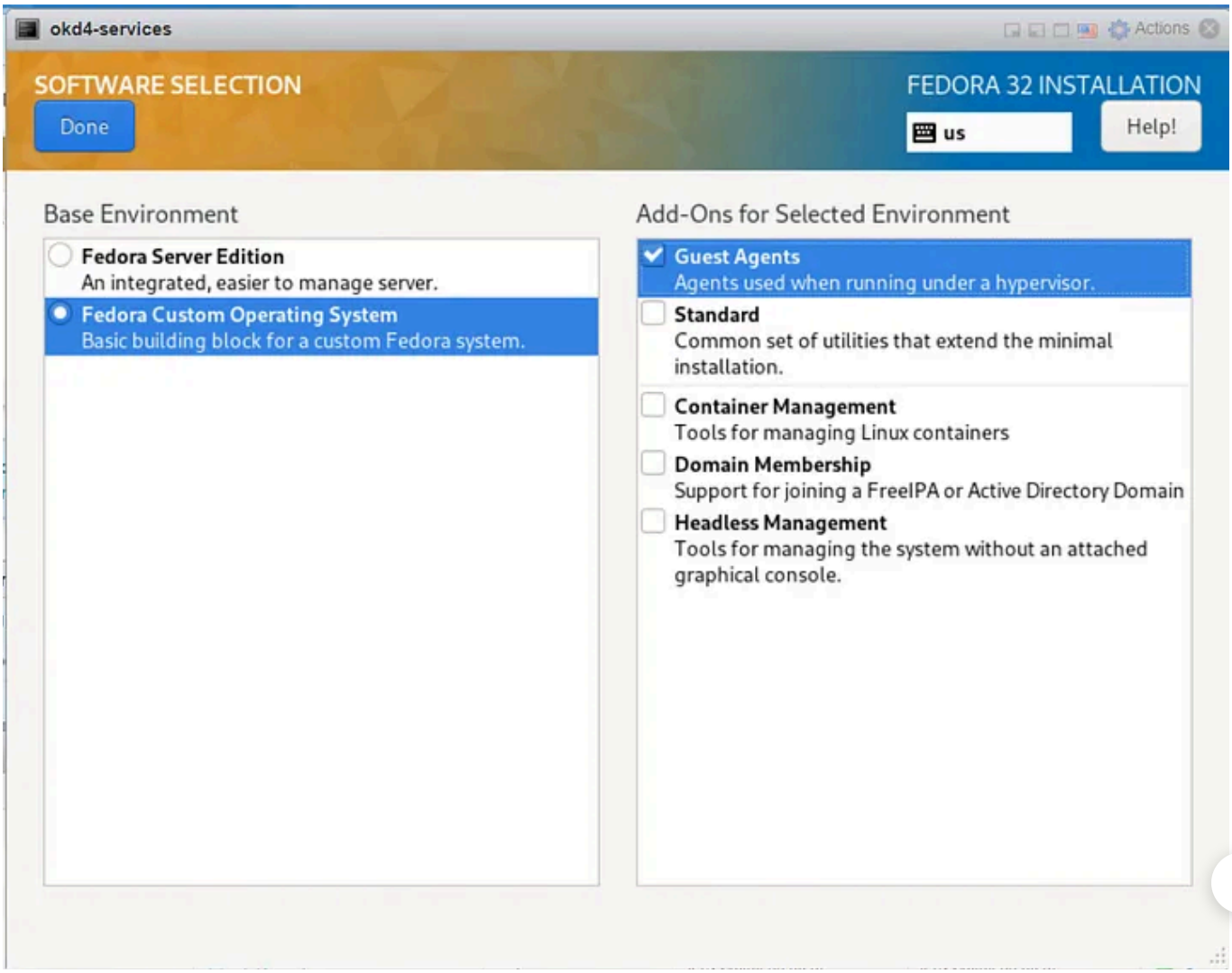
On the Manual Partitioning page, select Standard Partition, then click “Click Here to Create them automatically.”



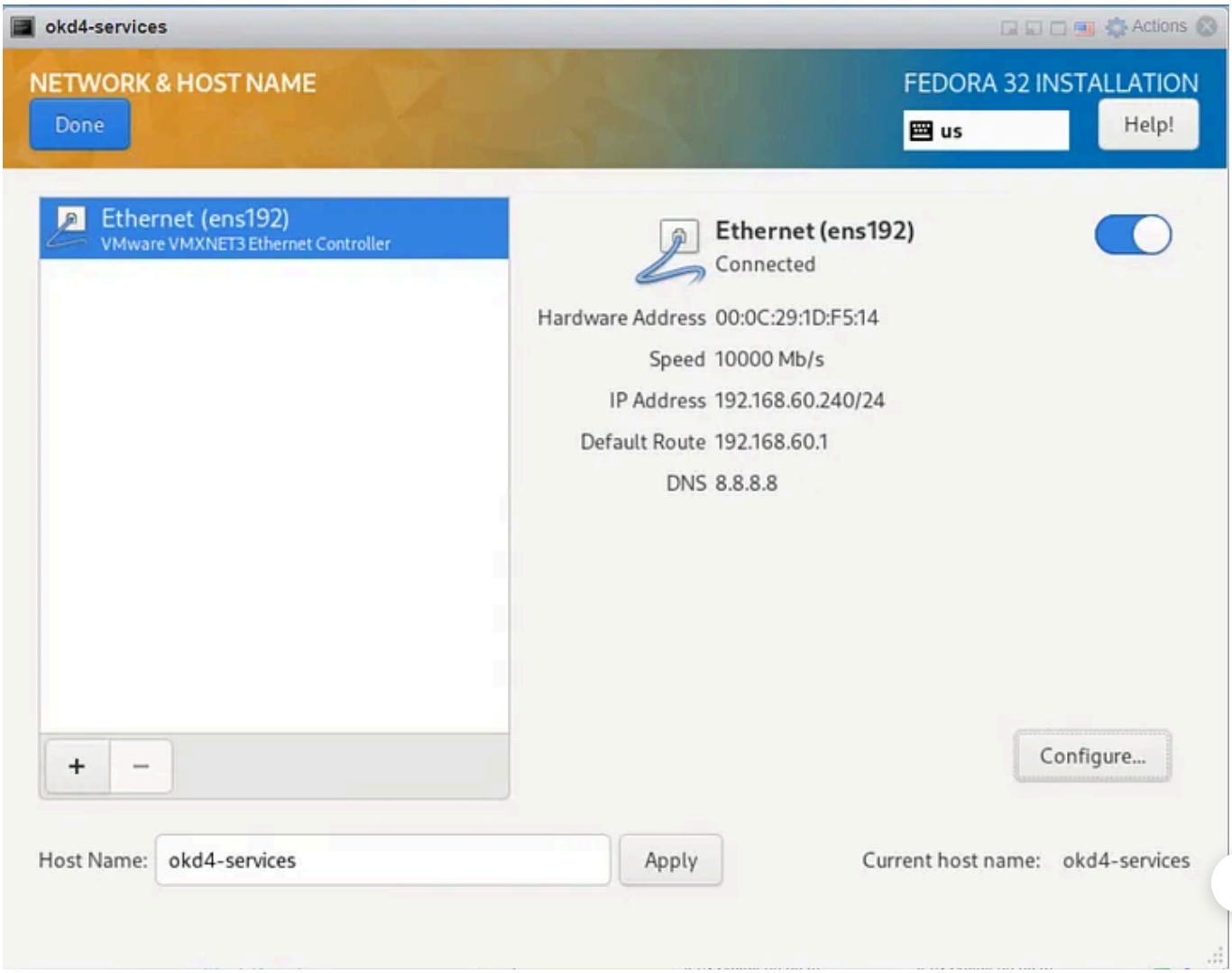
Select the “/” partition and click the “-” to delete it. Then click the “+” to re-add it. At this point, it should use the entire drive. Click Done, then Accept the changes.



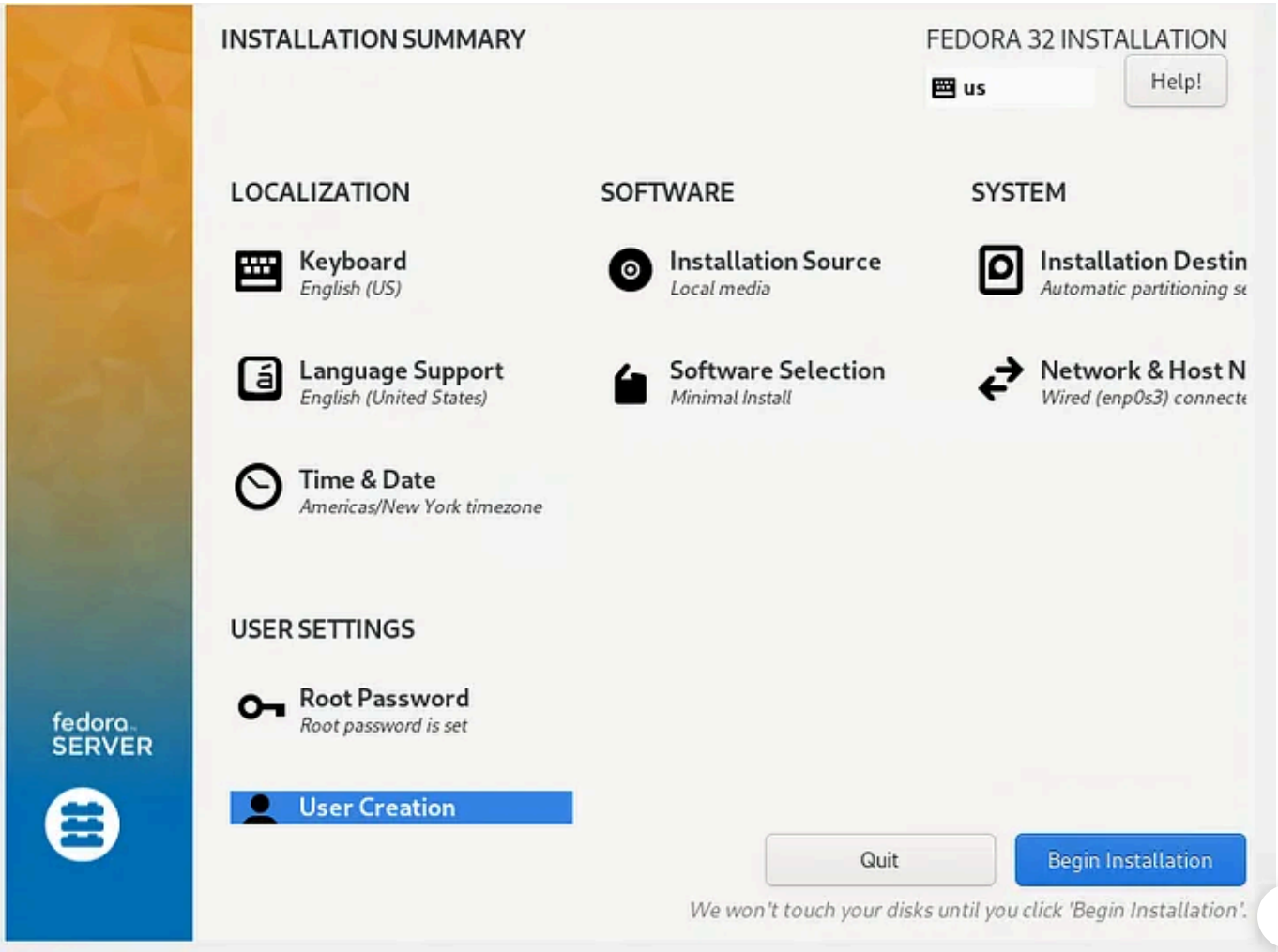
For Software Selection, choose “Fedora Custom Operating System” and include the “Guest Agents” add-on.



Enable the NIC connected to the VM Network and set the hostname as okd4-services, then click Apply and Done.



Set the Root password, and create an admin user. Click “Begin Installation” to start the install.



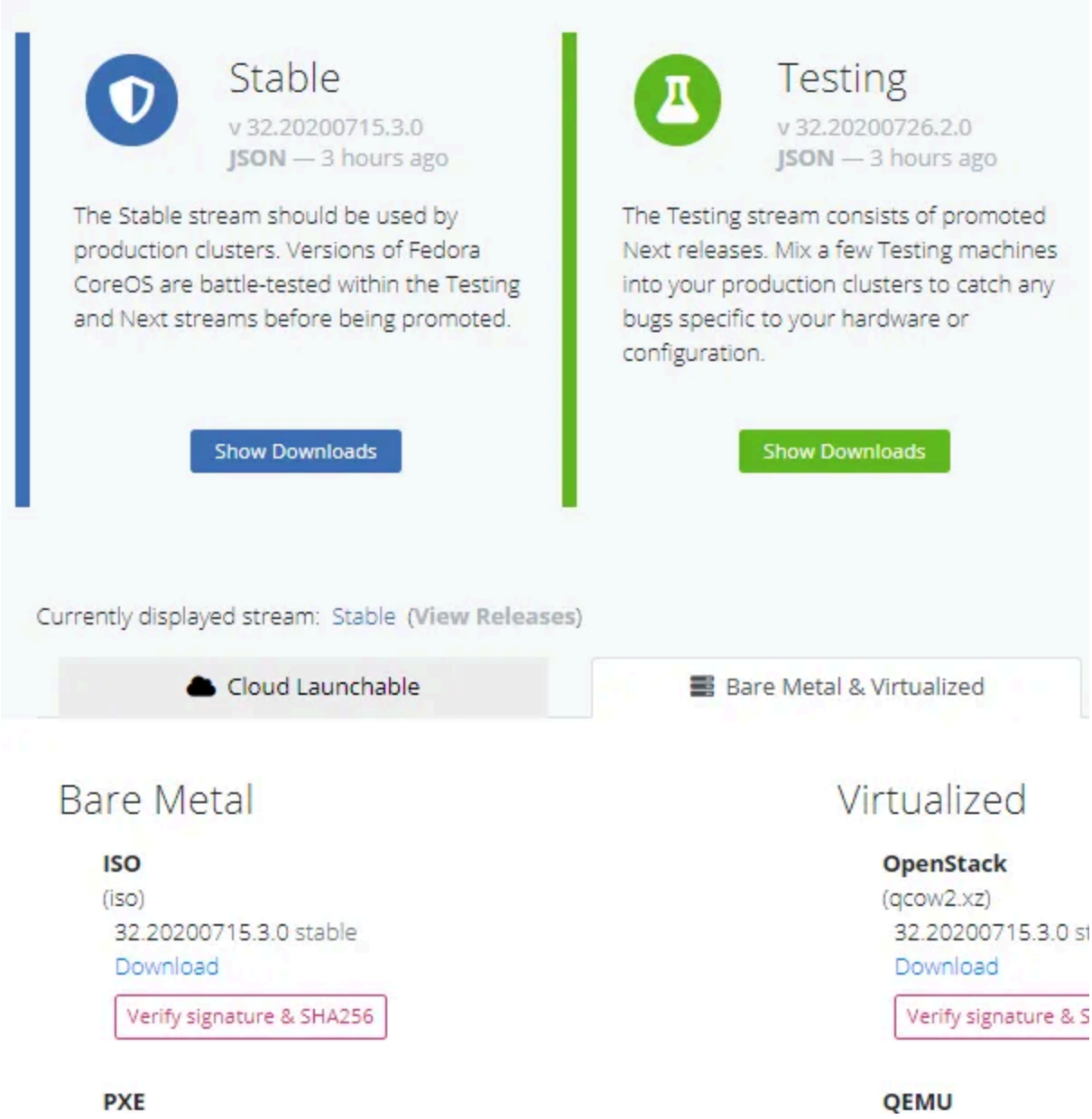
After the installation has completed, login, and update the OS.

```
sudo dnf update -y
sudo dnf install -y git wget vim
sudo init 6
```

Create bootstrap and master nodes:

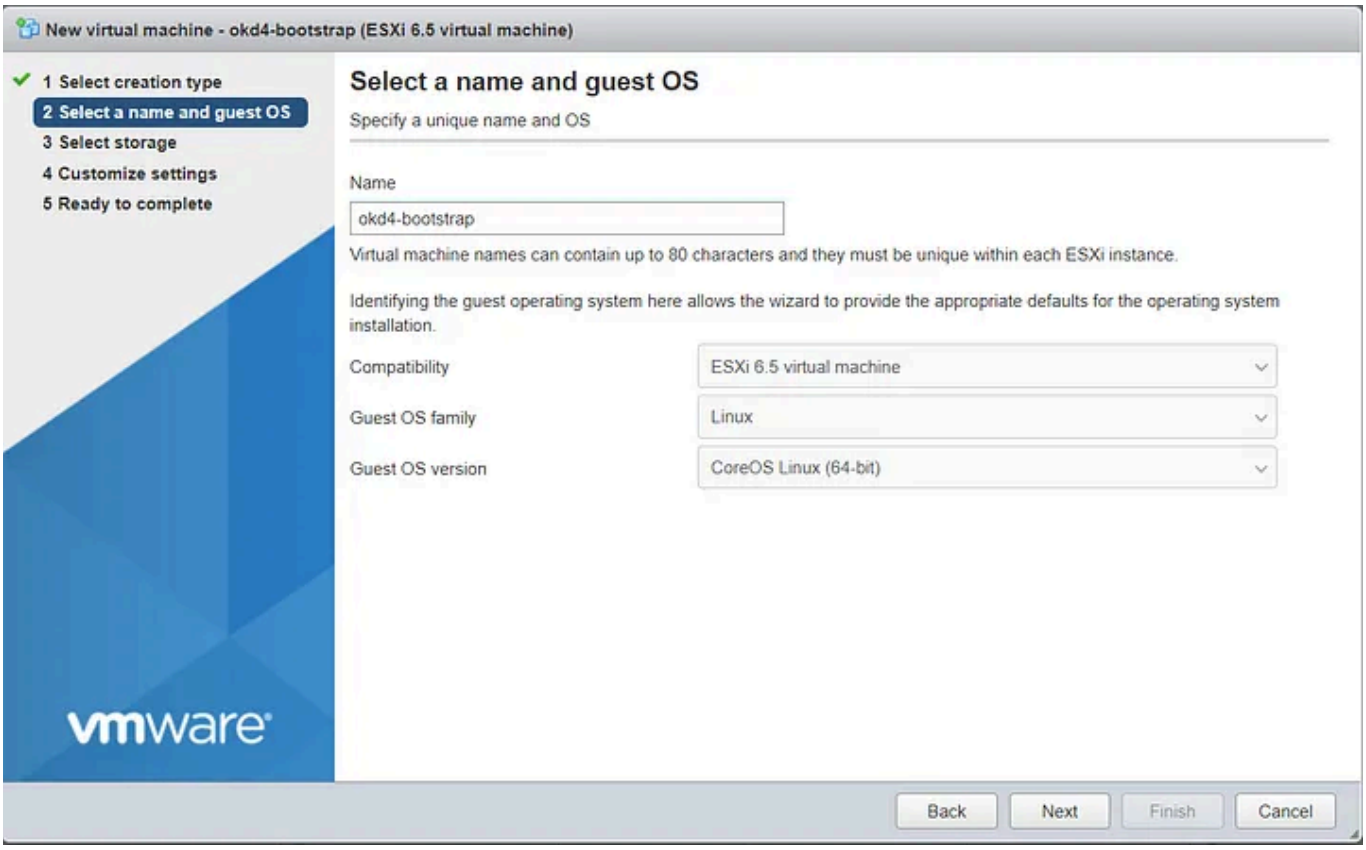
Download the Fedora CoreOS Bare Metal ISO and upload it to your ESXi datastore.

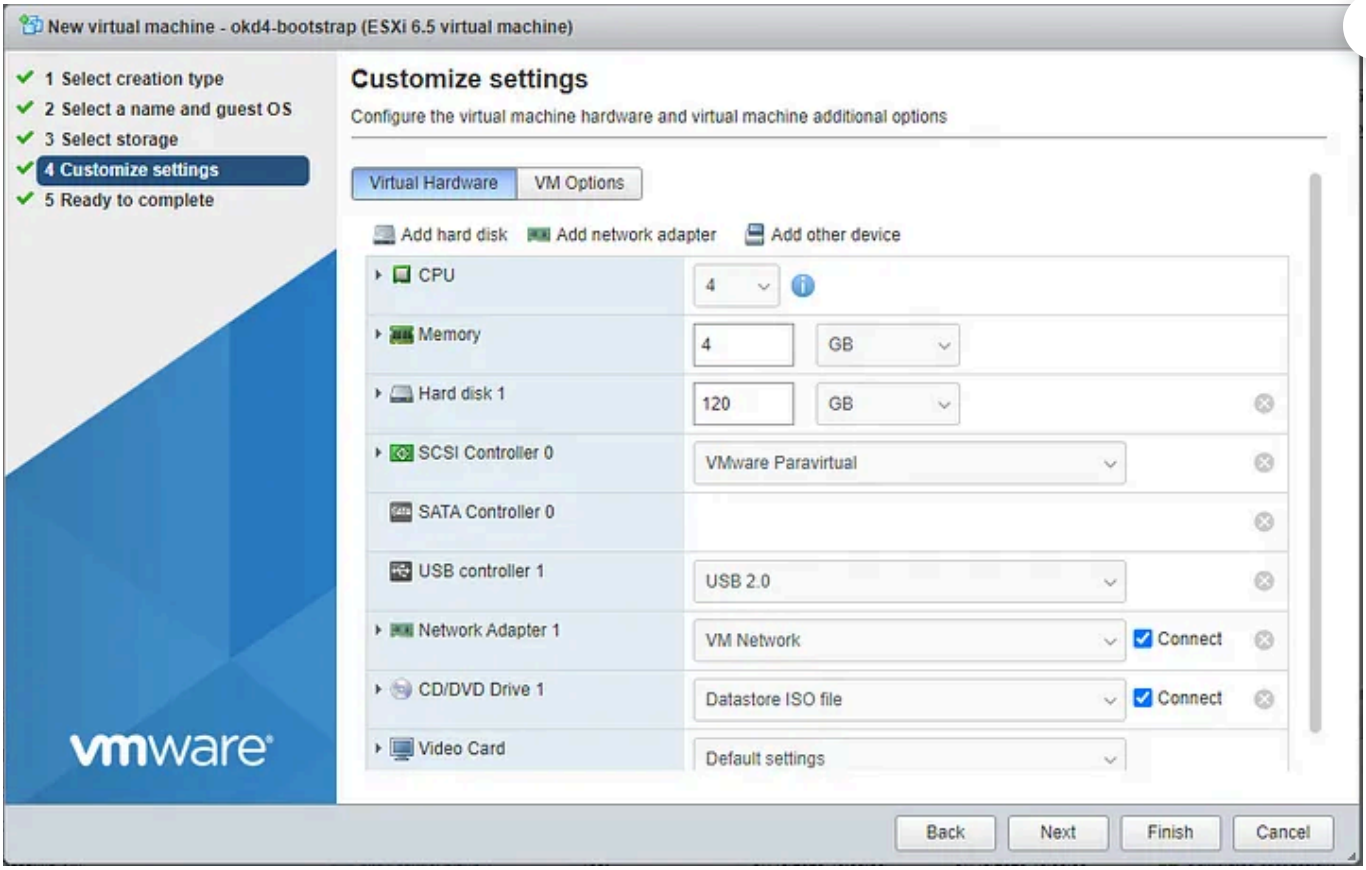
The latest stable version at the time of writing is 32.20200715.3.0

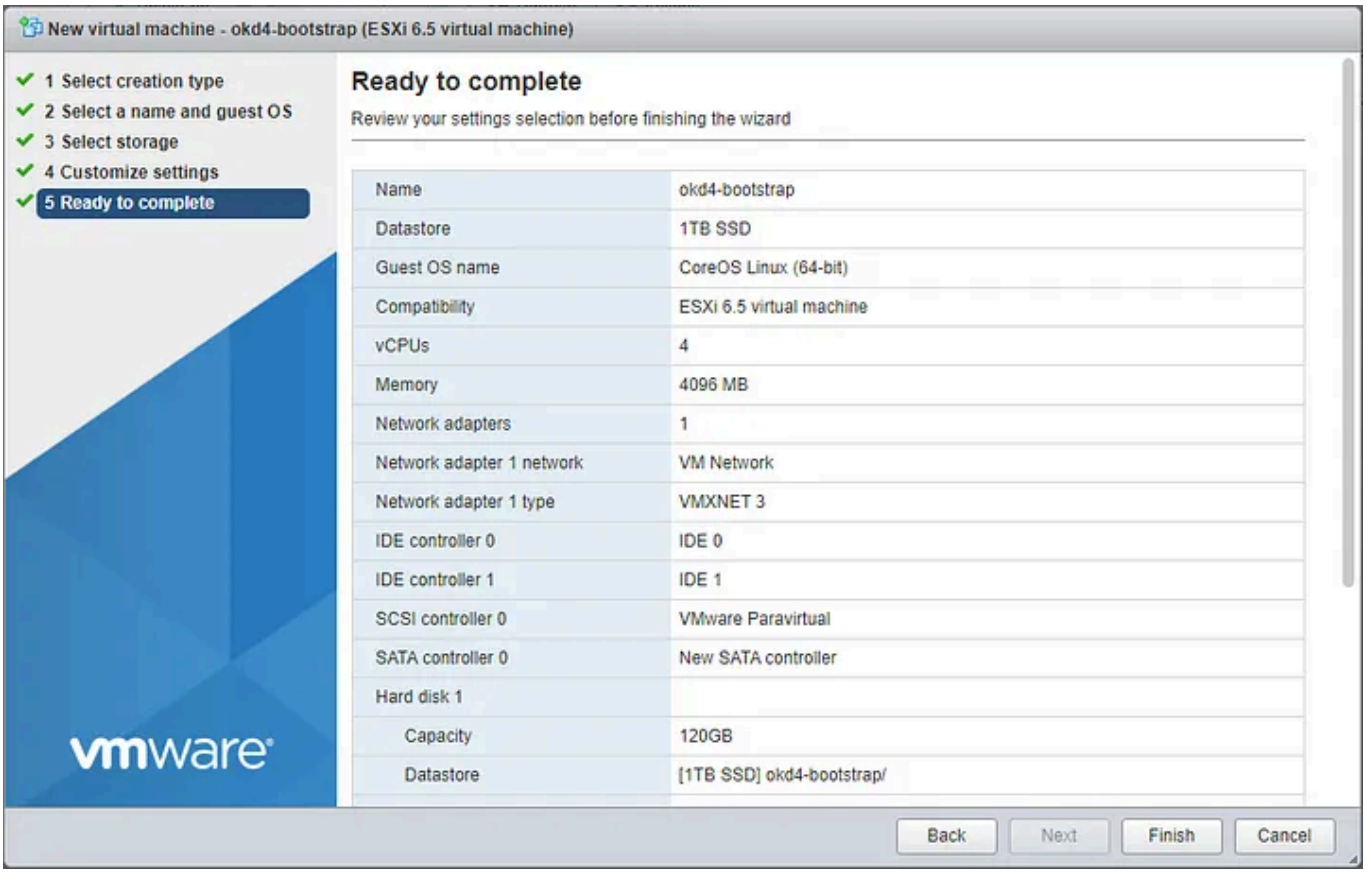


Create the two ODK nodes (okd4-bootstrap and okd4-control-plane-1) on your ESXi host using the values in the spreadsheet at the beginning of this post:

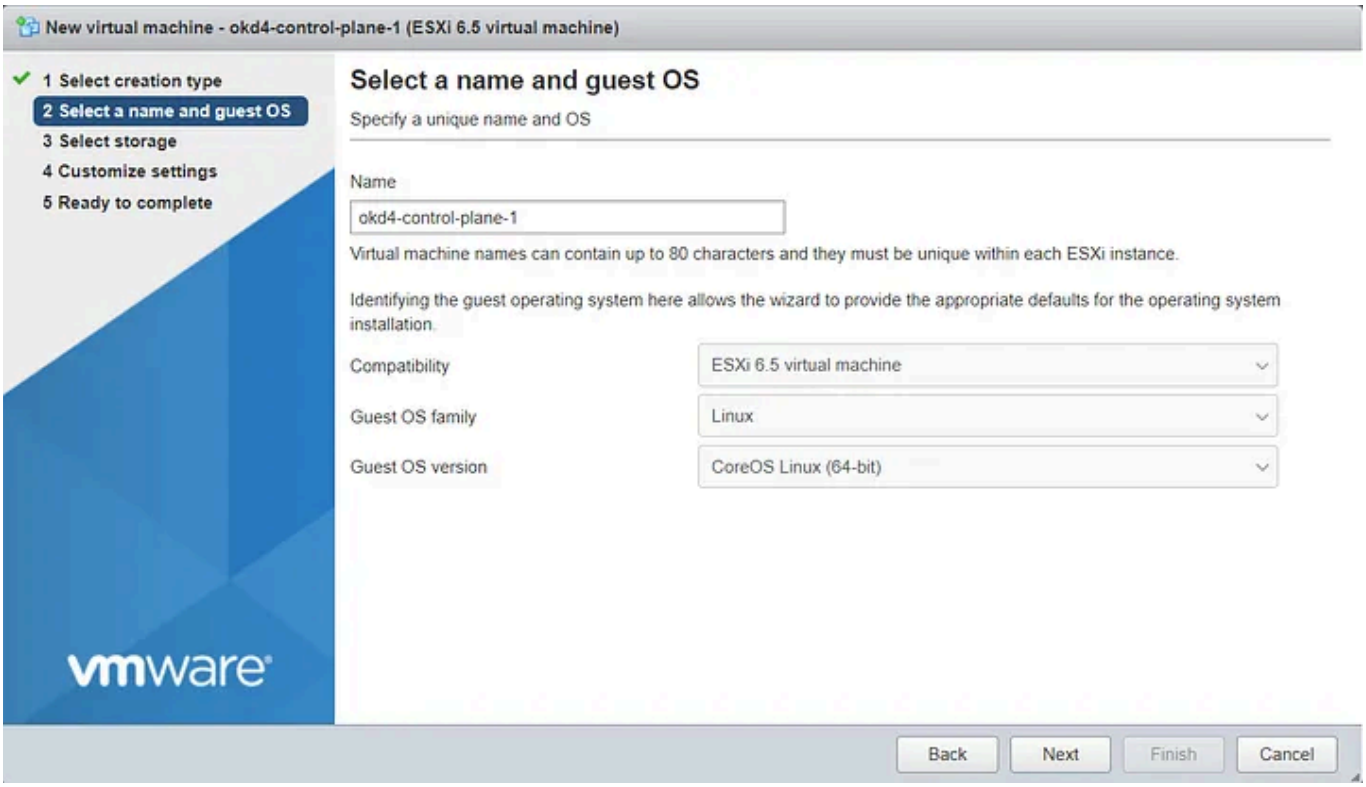
okd4-bootstrap:

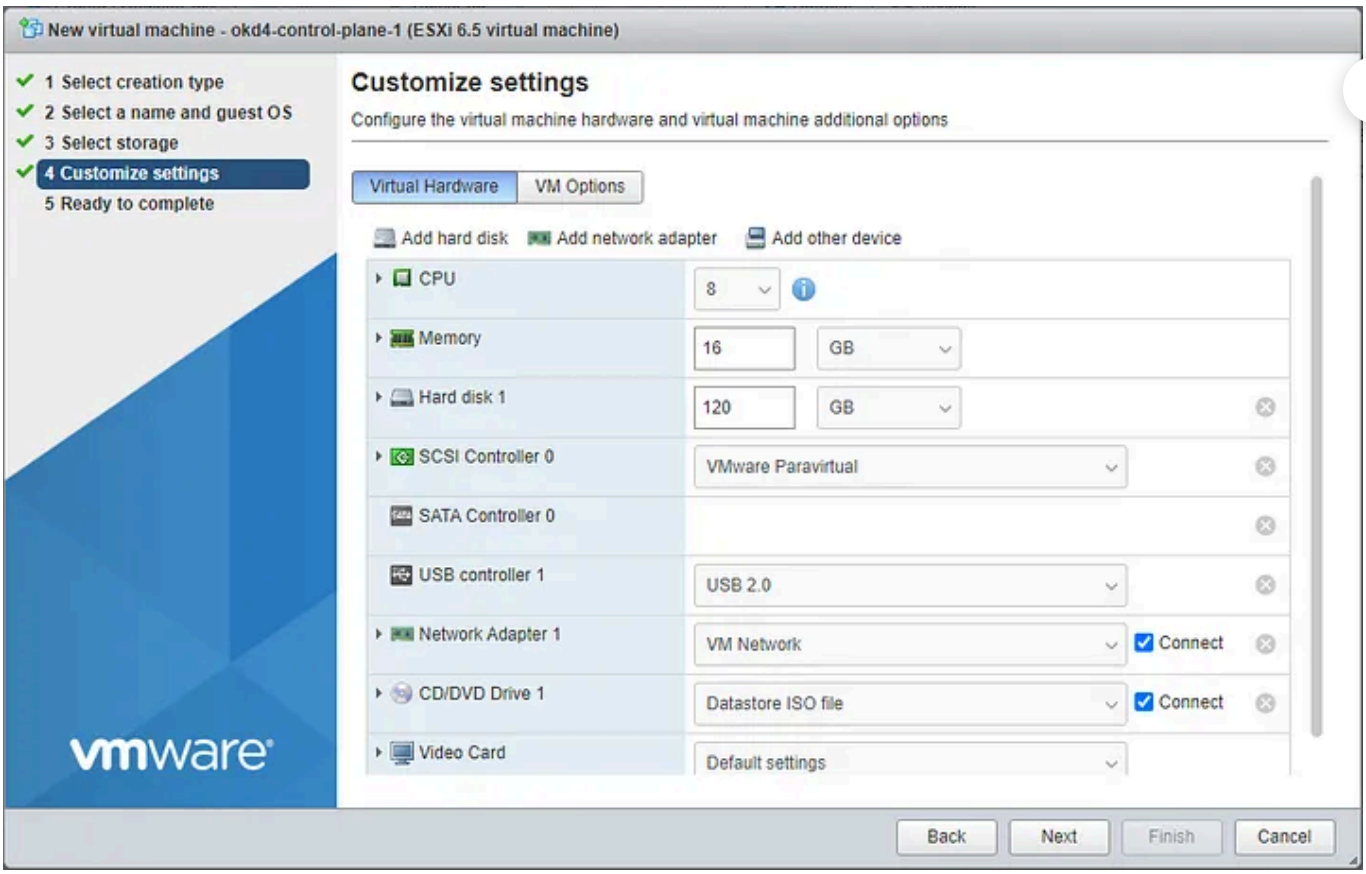

New virtual machine - okd4-bootstrap (ESXi 6.5 virtual machine)
1 Select creation type
2 Select a name and guest OS
3 Select storage
4 Customize settings
5 Ready to complete
Select a name and guest OS
Specify a unique name and OS
Name
okd4-bootstrap
Virtual machine names can contain up to 80 characters and they must be unique within each ESXi instance.
Identifying the guest operating system here allows the wizard to provide the appropriate defaults for the operating system installation.
Compatibility
ESXi 6.5 virtual machine
Guest OS family
Linux
Guest OS version
CoreOS Linux (64-bit)
vmware
Back
Next
Finish
Cancel

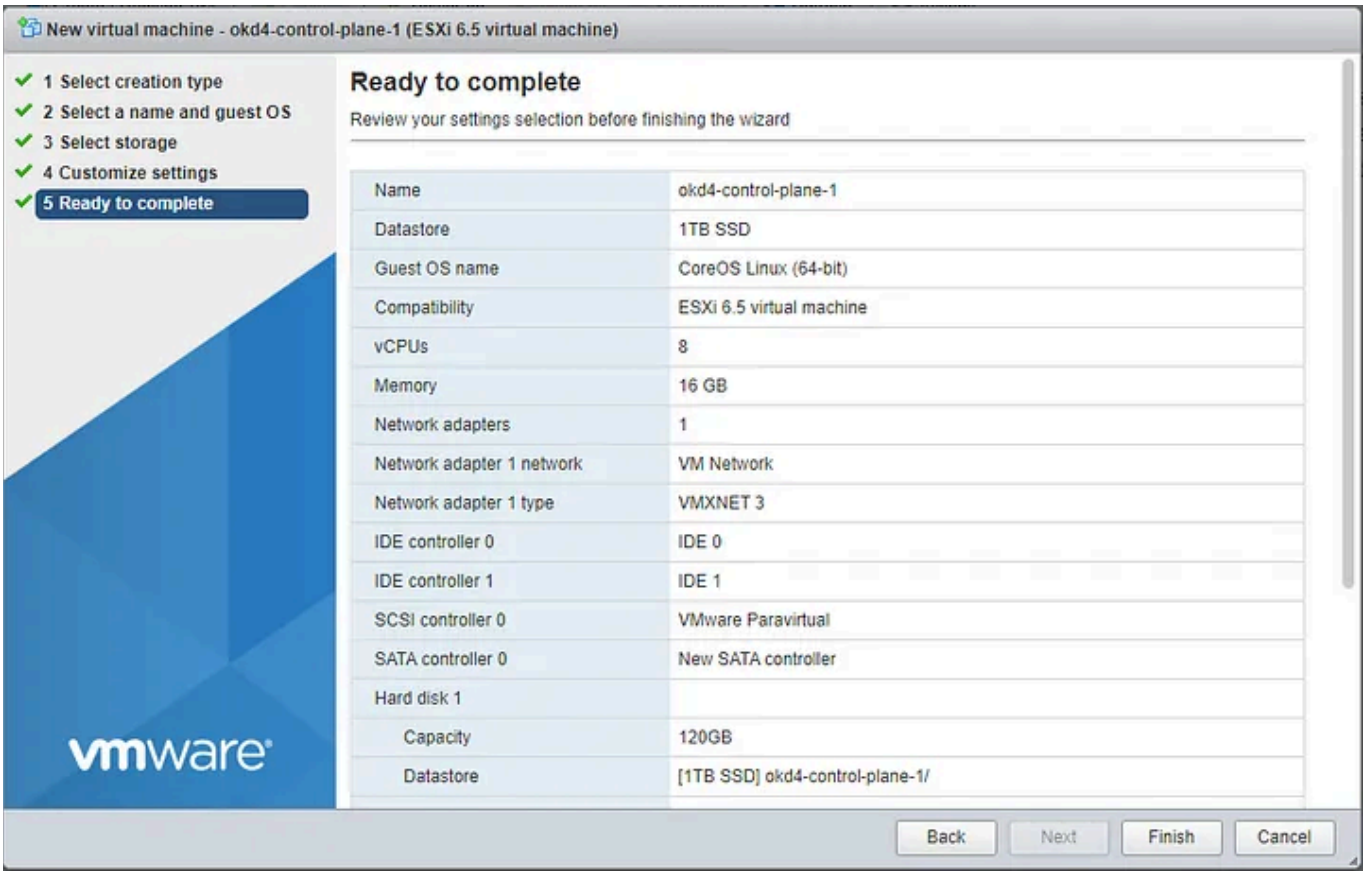

New virtual machine - okd4-bootstrap (ESXi 6.5 virtual machine)
1 Select creation type
2 Select a name and guest OS
3 Select storage
4 Customize settings
5 Ready to complete
Customize settings
Configure the virtual machine hardware and virtual machine additional options
Virtual Hardware
VM Options
Add hard disk
Add network adapter
Add other device
CPU
4
Memory
4
GB
Hard disk 1
120
GB
SCSI Controller 0
VMware Paravirtual
SATA Controller 0
USB controller 1
USB 2.0
Network Adapter 1
VM Network
Connect
CD/DVD Drive 1
Datastore ISO file
Connect
Video Card
Default settings
vmware
Back
Next
Finish
Cancel



okd4-control-plane-1


New virtual machine - okd4-control-plane-1 (ESXi 6.5 virtual machine)
1 Select creation type
2 Select a name and guest OS
3 Select storage
4 Customize settings
5 Ready to complete
Select a name and guest OS
Specify a unique name and OS
Name
okd4-control-plane-1
Virtual machine names can contain up to 80 characters and they must be unique within each ESXi instance.
Identifying the guest operating system here allows the wizard to provide the appropriate defaults for the operating system installation.
Compatibility
ESXi 6.5 virtual machine
Guest OS family
Linux
Guest OS version
CoreOS Linux (64-bit)
vmware
Back
Next
Finish
Cancel


New virtual machine - okd4-control-plane-1 (ESXi 6.5 virtual machine)
1 Select creation type
2 Select a name and guest OS
3 Select storage
4 Customize settings
5 Ready to complete
Customize settings
Configure the virtual machine hardware and virtual machine additional options
Virtual Hardware
VM Options
Add hard disk
Add network adapter
Add other device
CPU
8
Memory
16
GB
Hard disk 1
120
GB
SCSI Controller 0
VMware Paravirtual
SATA Controller 0
USB controller 1
USB 2.0
Network Adapter 1
VM Network
Connect
CD/DVD Drive 1
Datastore ISO file
Connect
Video Card
Default settings
vmware
Back
Next
Finish
Cancel



You should end up with the following VMs:

Configure okd4-services VM to host various services:

The okd4-services VM is used to provide DNS, NFS exports, web server, and load balancing.

Open a terminal on the okd4-services VM and clone the okd4_files repo (branch snc) that contains the DNS, HAProxy, and install-conf.yaml example files:

```
cd
git clone -b snc https://github.com/cragr/okd4_files.git
cd okd4_files
```

Install bind (DNS)

```
sudo dnf -y install bind bind-utils
```

Copy the named config files and zones:

```
sudo cp named.conf /etc/named.conf
sudo cp named.conf.local /etc/named/
sudo mkdir /etc/named/zones
sudo cp db* /etc/named/zones
```

Enable and start named:

```
sudo systemctl enable named
sudo systemctl start named
sudo systemctl status named
```

Create firewall rules:

```
sudo firewall-cmd --permanent --add-port=53/udp
```

```
sudo firewall-cmd --reload
```

Change the DNS on the okd4-service to 127.0.0.1:

```
sudo nmcli connection modify ens192 ipv4.dns "127.0.0.1"
```

Restart the network services on the okd4-services VM:

```
sudo systemctl restart NetworkManager
```

Test DNS on the okd4-services.

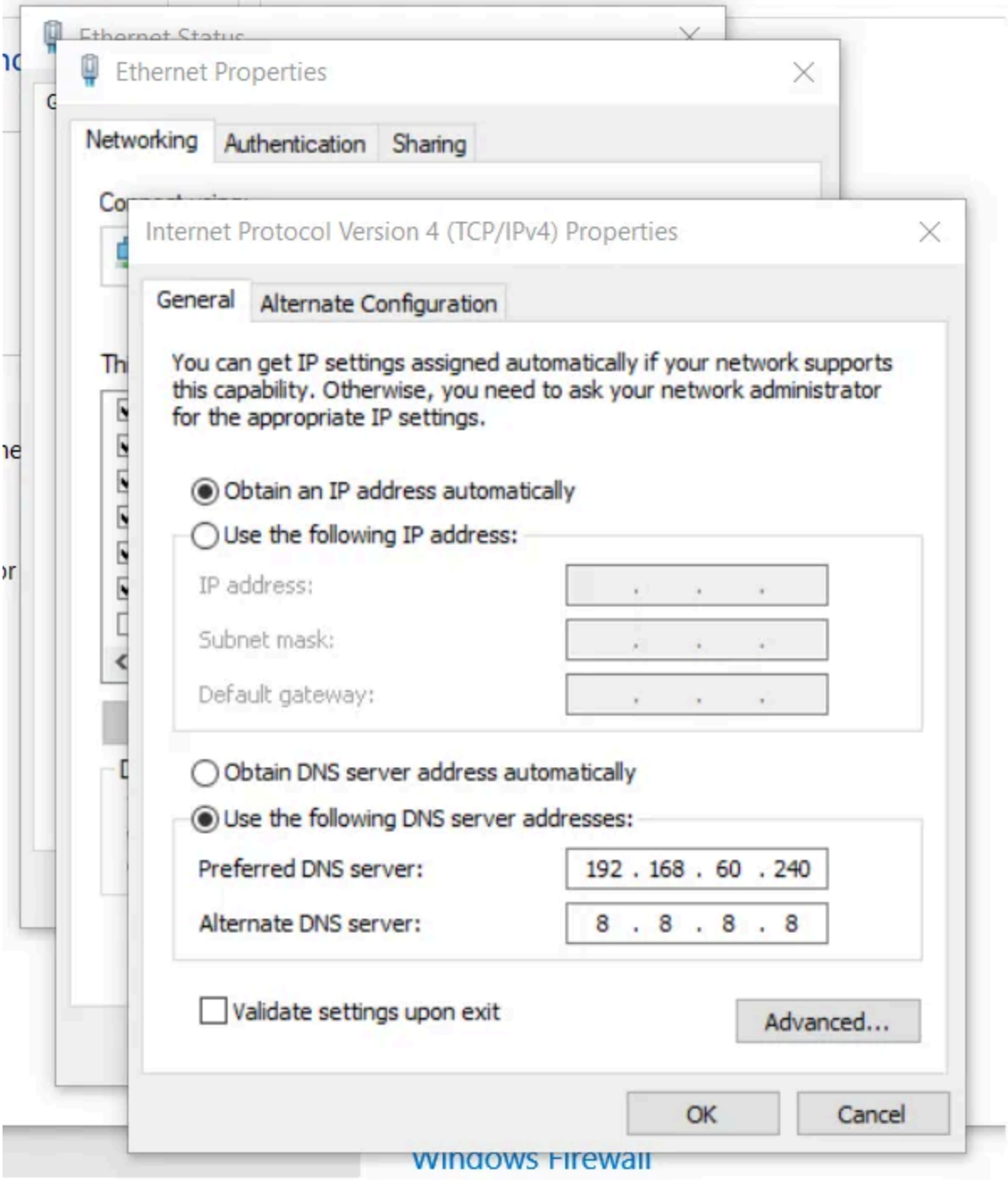
```
dig okd.local
dig -x 192.168.60.240
```

With DNS working correctly, you should see the following results:

```
[crobinson@okd4-services okd4_files]$ dig okd.local

; <<>> DiG 9.11.13-RedHat-9.11.13-3.el8 <<>> okd.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64424
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7009863cfd0c3194df61cc045f04c51c67e06b2f24fd0618 (good)
;; QUESTION SECTION:
;okd.local.                     IN      A

;; AUTHORITY SECTION:
okd.local.              604800  IN      SOA     okd4-services.okd.local. admin.o
kd.local. 1 604800 86400 2419200 604800

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Jul 07 14:55:24 EDT 2020
;; MSG SIZE  rcvd: 122
```

```
[crobinson@okd4-services okd4_files]$ dig -x 192.168.1.210

; <<>> DiG 9.11.13-RedHat-9.11.13-3.el8 <<>> -x 192.168.1.210
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65007
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 23bfcaa3f3140ab1261acff65f04c4d921fa9f8011631ba6 (good)
;; QUESTION SECTION:
;210.1.168.192.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
210.1.168.192.in-addr.arpa. 604800 IN   PTR     api-int.lab.okd.local.
210.1.168.192.in-addr.arpa. 604800 IN   PTR     okd4-services.okd.local.
210.1.168.192.in-addr.arpa. 604800 IN   PTR     api.lab.okd.local.

;; AUTHORITY SECTION:
1.168.192.in-addr.arpa. 604800  IN      NS      okd4-services.okd.local.

;; ADDITIONAL SECTION:
okd4-services.okd.local. 604800 IN      A       192.168.1.210

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Jul 07 14:54:17 EDT 2020
;; MSG SIZE  rcvd: 194
```

Change your workstation's primary DNS to match the okd4-services IP, 192.168.60.240 so you can resolve the DNS entries.



Install HAProxy:

```
sudo dnf install haproxy -y
```

Copy haproxy config from the git okd4_files directory :

```
sudo cp haproxy.cfg /etc/haproxy/haproxy.cfg
```

Start, enable, and verify HA Proxy service:

```
sudo setsebool -P haproxy_connect_any 1
sudo systemctl enable haproxy
sudo systemctl start haproxy
sudo systemctl status haproxy
```

Add OKD firewall ports:

```
sudo firewall-cmd --permanent --add-port=6443/tcp
sudo firewall-cmd --permanent --add-port=22623/tcp
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https
sudo firewall-cmd --reload
```

Install Apache/HTTPD

```
sudo dnf install -y httpd
```

Change httpd to listen port to 8080:

```
sudo sed -i 's/Listen 80/Listen 8080/' /etc/httpd/conf/httpd.conf
```

Enable and Start httpd service/Allow port 8080 on the firewall:

```
sudo setsebool -P httpd_read_user_content 1
sudo systemctl enable httpd
sudo systemctl start httpd
sudo firewall-cmd --permanent --add-port=8080/tcp
sudo firewall-cmd --reload
```

Test the webserver:

```
curl localhost:8080
```

A successful curl should look like this:

```
          <h2>推广 Apache 及 CentOS</h2>
          <p>你可在 Apache 及 CentOS Linux 驱动的 HTTP 服务器上随意采用下列图像。多
谢采用 Apache 及 CentOS! </p>
          <p>
            <a href="http://httpd.apache.org/">
              <img src="noindex/common/images/pb-apache.png" alt="[ Powered by Apach
e ]"/>
            </a>
            <a href="http://www.centos.org/">
              <img src="noindex/common/images/pb-centos.png" alt="[ Powered by CentO
S Linux ]"/>
            </a>
          </p>
        </div>
        <div class="col-md-6">
        </div>
        <div class="col-md-6">
          <h2>CentOS 计划</h2>
          <p>CentOS Linux 发行版本是一个稳定、高预测性、高管理性、高重复性的平台，它
源于 Red Hat 企业级 Linux（RHEL）的源代码。</p>
          <p>除了是寄存网站的热门选择外，CentOS 亦为开源社群提供一个可扩展的丰富平台
。请拜访 <a href="http://www.centos.org/">CentOS 网站</a> 获取更多信息。</p>
        </div>
      </section>
      <hr/>
    </main>
    <footer class="container">
      <p>© 2019 CentOS 计划 | <a href="https://www.centos.org/legal/">法律条文</a> |
 <a href="https://www.centos.org/legal/privacy/">私隐</a></p>
    </footer>
  </body>
</html>
[crobinson@okd4-services okd4_files]$ S
```

Congratulations, You Are Half Way There!

Download the openshift-installer and oc client:

SSH to the okd4-services VM

To download the latest oc client and openshift-install binaries, you need to use an existing version of the oc client.

Download the 4.5 version of the oc client and openshift-install from the OKD releases page. Example:

```
cd
wget https://github.com/openshift/okd/releases/download/4.5.0-0.okd-2020-07-29-070316/openshift-client-linux-4.5.0-0.okd-2020-07-29-070316.tar.gz
wget https://github.com/openshift/okd/releases/download/4.5.0-0.okd-2020-07-29-070316/openshift-install-linux-4.5.0-0.okd-2020-07-29-070316.tar.gz
```


```
tar -zxvf openshift-client-linux-4.5.0-0.okd-2020-07-29-070316.tar.gz
tar -zxvf openshift-install-linux-4.5.0-0.okd-2020-07-29-070316.tar.gz
```

Move the kubectl, oc, and openshift-install to /usr/local/bin and show the version:

```
sudo mv kubectl oc openshift-install /usr/local/bin/
oc version
openshift-install version
```

The latest and recent releases are available at https://origin-release.svc.ci.openshift.org

Setup the openshift-installer:

In the install-config.yaml, you can either use a pull-secret from RedHat or the default of “{“auths”:{“fake”:{“auth”: “bar”}}}” as the pull-secret.

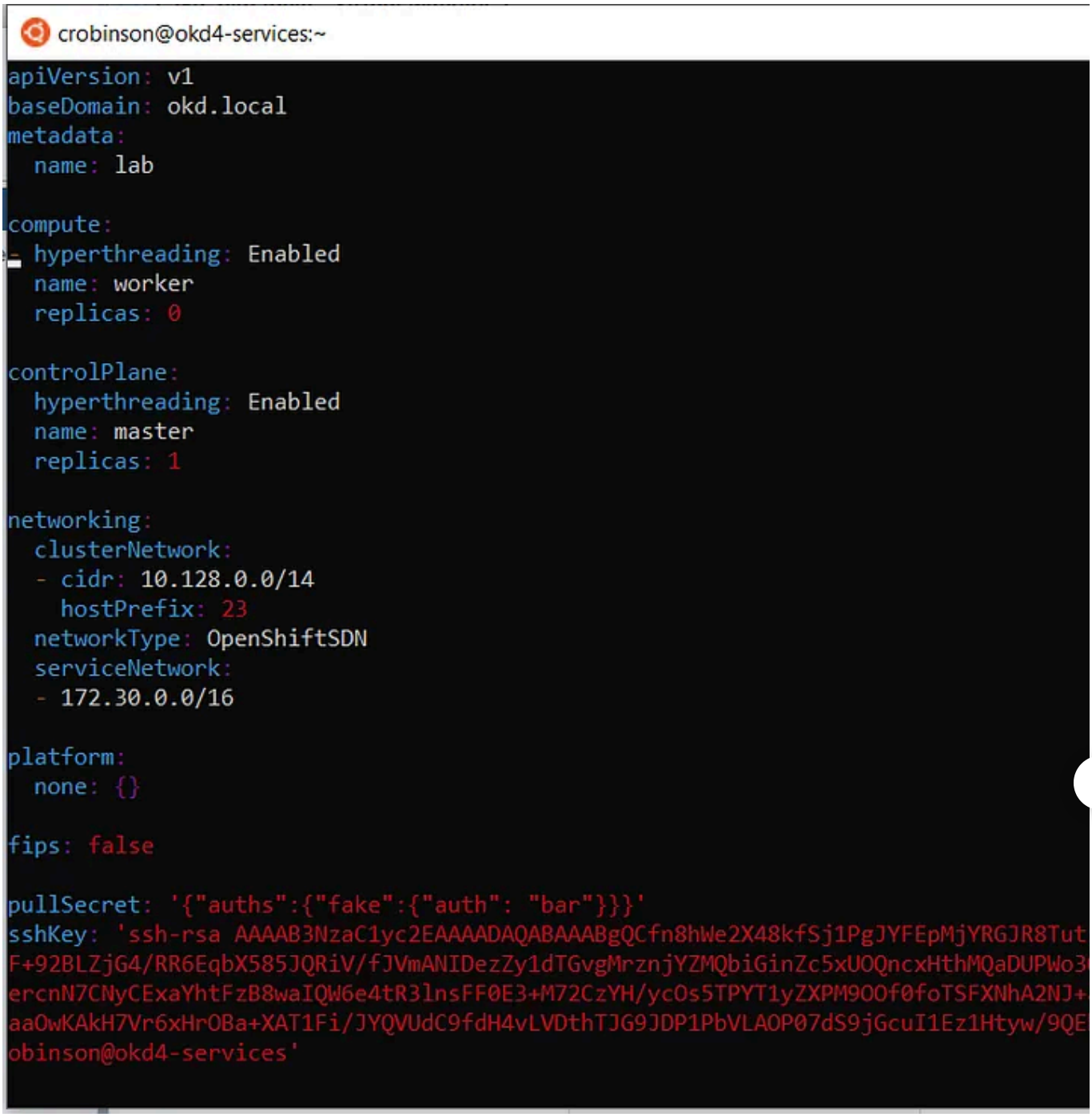
Generate an SSH key if you do not already have one.

```
ssh-keygen
```

Create an install directory and copy the install-config.yaml file:

```
cd
mkdir install_dir
cp okd4_files/install-config.yaml ./install_dir
```

Edit the install-config.yaml in the install_dir, add your ssh key, and edit the controlPlane master replicas to 1. Backup the install-config.yaml as it will be removed when generating the manifests:




```
vim ./install_dir/install-config.yaml
cp ./install_dir/install-config.yaml ./install_dir/install-
config.yaml.bak
```

Generate the Kubernetes manifests for the cluster, ignore the warning:

```
openshift-install create manifests --dir=install_dir/
```

Now you can create the ignition-configs:

```
openshift-install create ignition-configs --dir=install_dir/
```

Note: If you reuse the install_dir, make sure it is empty. Hidden files are created after generating the configs, and they should be removed before you use the same folder on a 2nd attempt.

Host ignition and Fedora CoreOS files on the webserver:

Create okd4 directory in /var/www/html:

```
sudo mkdir /var/www/html/okd4
```

Copy the install_dir contents to /var/www/html/okd4 and set permissions:

```
sudo cp -R install_dir/* /var/www/html/okd4/
sudo chown -R apache: /var/www/html/
sudo chmod -R 755 /var/www/html/
```

Test the webserver:

```
curl localhost:8080/okd4/metadata.json
```

Download the Fedora CoreOS bare-metal bios image and sig files and shorten the file names:

```
cd /var/www/html/okd4/
sudo wget
https://builds.coreos.fedoraproject.org/prod/streams/stable/builds/32
.20200715.3.0/x86_64/fedora-coreos-32.20200715.3.0-
metal.x86_64.raw.xz
sudo wget
https://builds.coreos.fedoraproject.org/prod/streams/stable/builds/32
.20200715.3.0/x86_64/fedora-coreos-32.20200715.3.0-
metal.x86_64.raw.xz.sig
sudo mv fedora-coreos-32.20200715.3.0-metal.x86_64.raw.xz fcos.raw.xz
sudo mv fedora-coreos-32.20200715.3.0-metal.x86_64.raw.xz.sig
fcos.raw.xz.sig
sudo chown -R apache: /var/www/html/
sudo chmod -R 755 /var/www/html/
```

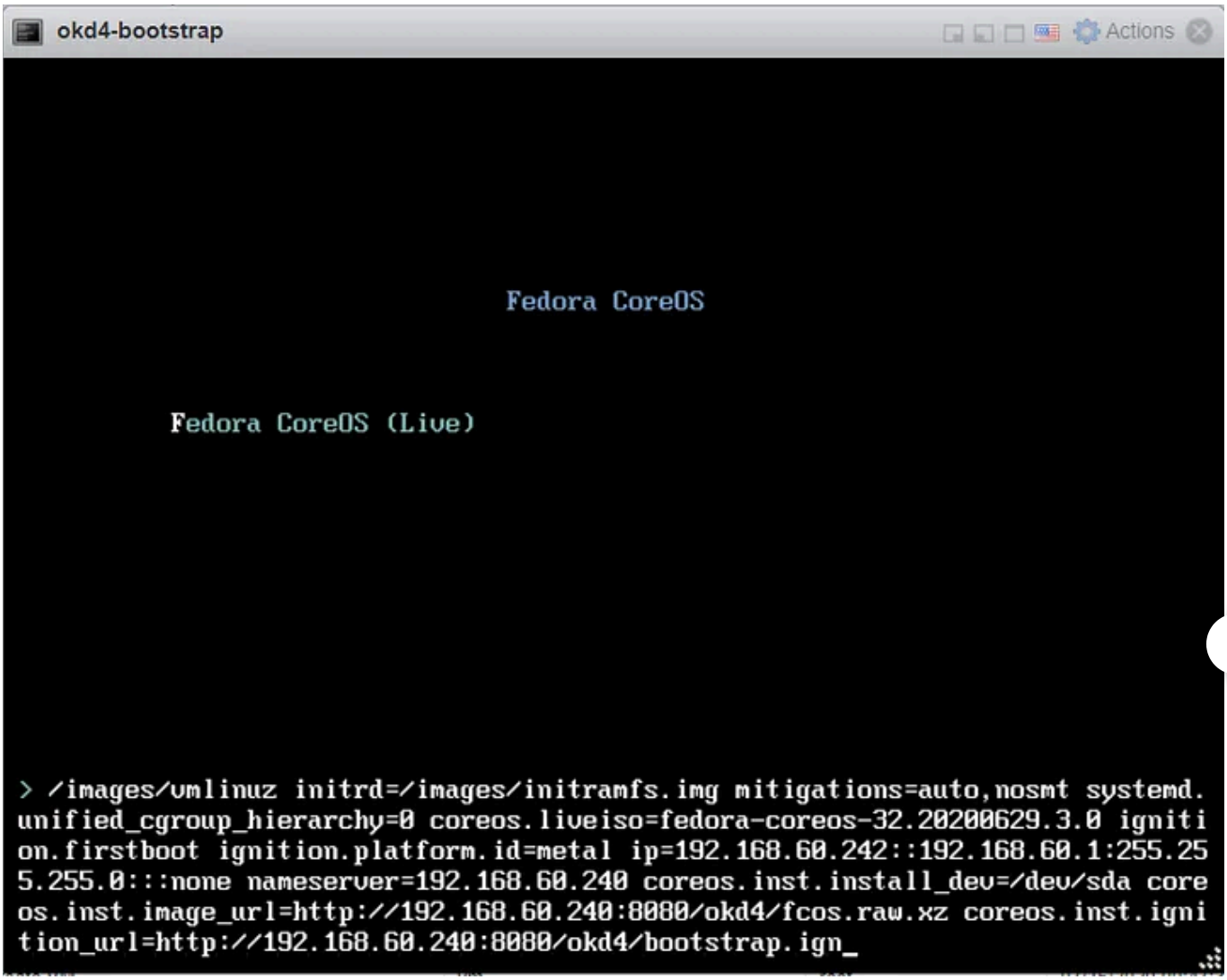
Starting the bootstrap node:

NOTE: I recommend you use a DHCP lease for both the bootstrap and control-plane nodes if possible instead of the line below beginning with "ip=" which will set static IP addresses.

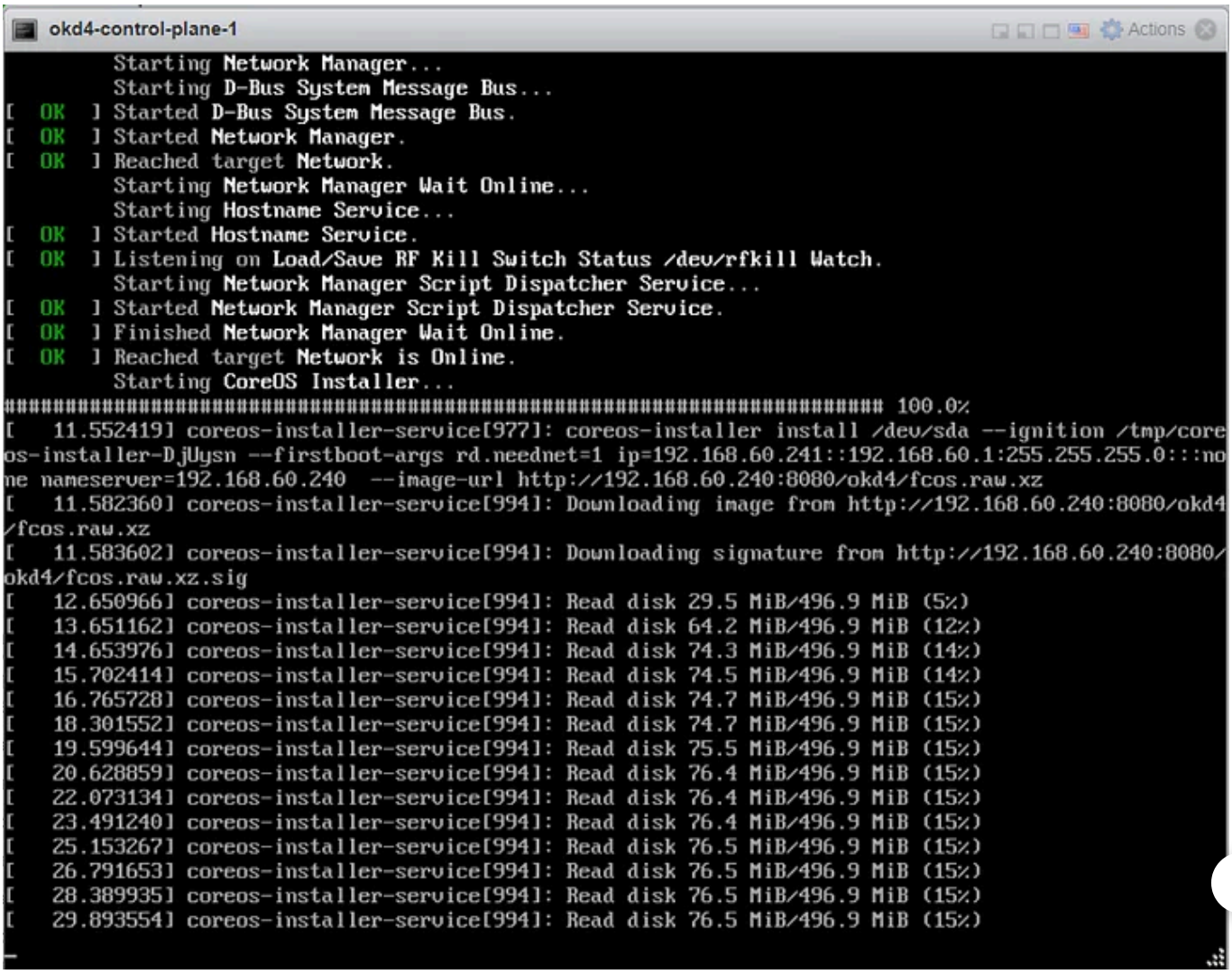
Power on the odk4-bootstrap VM. Press the TAB key to edit the kernel boot options. Once completed, press Enter to boot.

```
ip=192.168.60.242::192.168.60.1:255.255.255.0:::none
nameserver=192.168.60.240 coreos.inst.install_dev=/dev/sda
```

```
coreos.inst.image_url=http://192.168.60.240:8080/okd4/fcos.raw.xz
coreos.inst.ignition_url=http://192.168.60.240:8080/okd4/bootstrap.ig
n
```



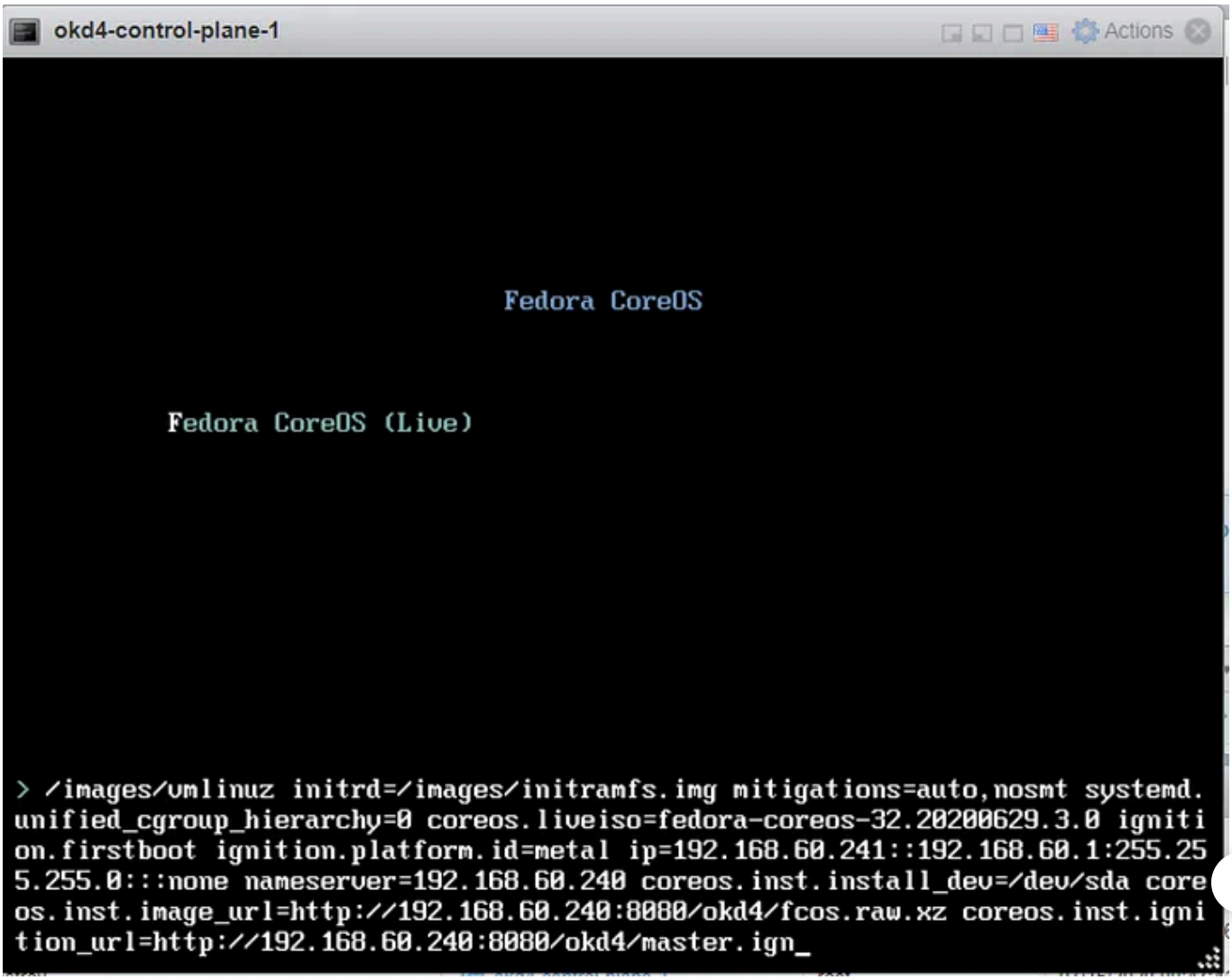
You should see that the fcos.raw.xz image and signature are downloading:



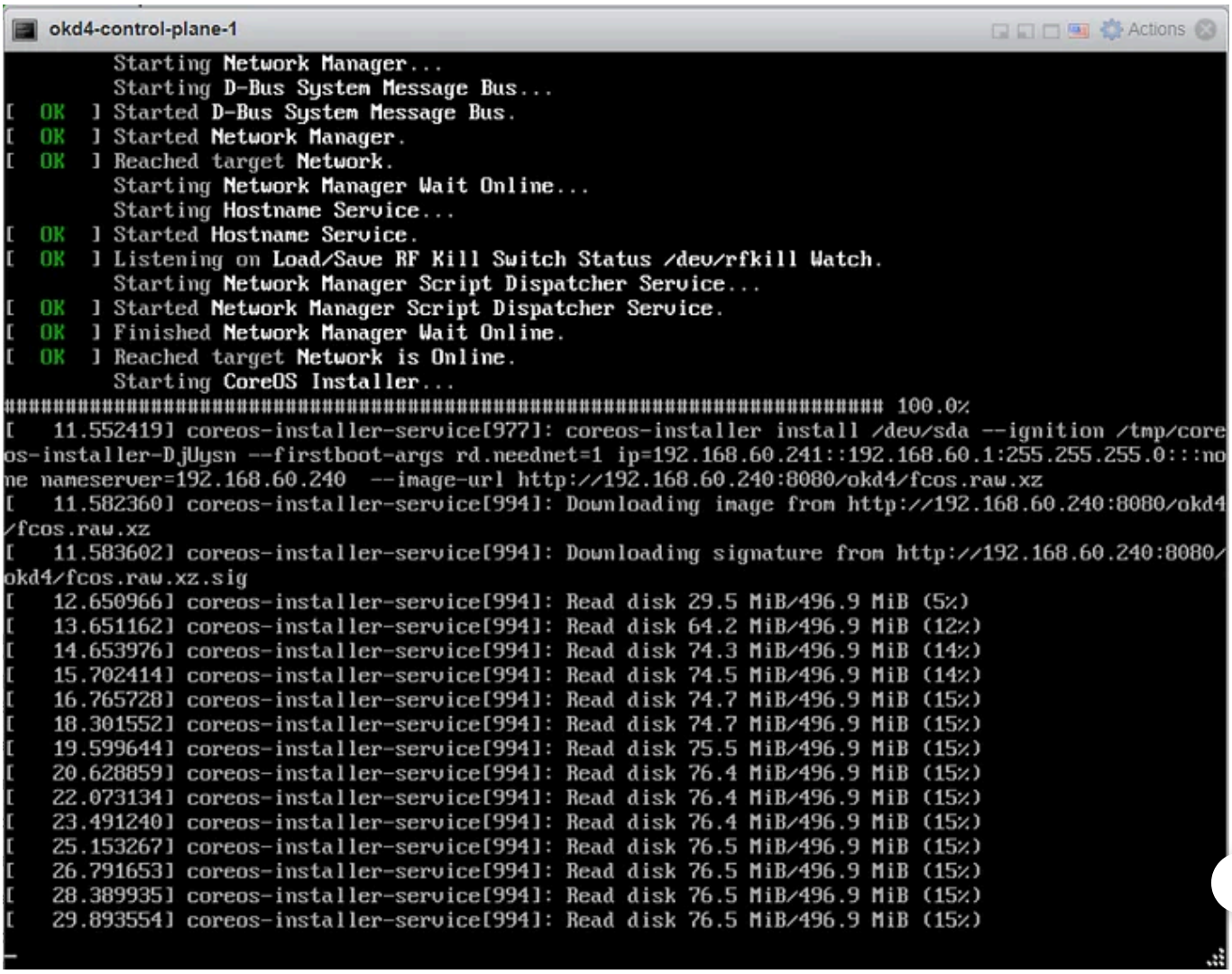
Starting the control-plane-1 node:

Power on the odk4-control-plane-1 VM. Press the TAB key to edit the kernel boot options. Once completed, press Enter to boot.

```
ip=192.168.60.241::192.168.60.1:255.255.255.0:::none
nameserver=192.168.60.240 coreos.inst.install_dev=/dev/sda
coreos.inst.image_url=http://192.168.60.240:8080/okd4/fcos.raw.xz
coreos.inst.ignition_url=http://192.168.60.240:8080/okd4/master.ign
```



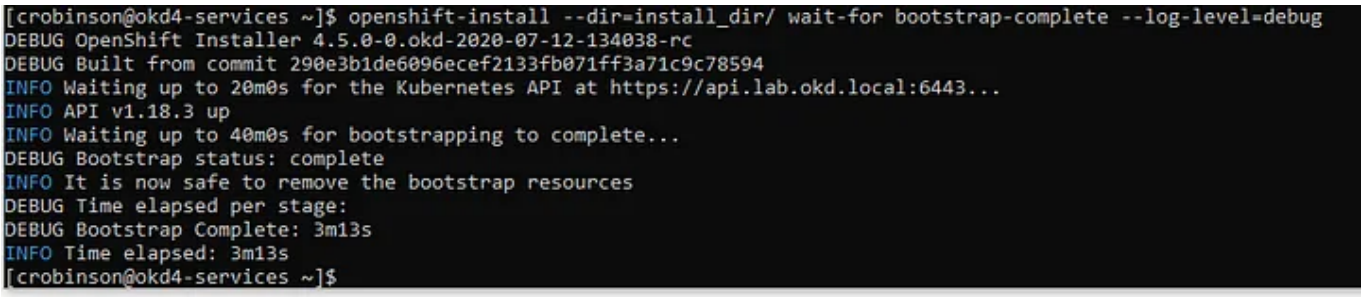
You should see that the fcos.raw.xz image and signature are downloading:



Monitor the bootstrap installation:

You can monitor the bootstrap process from the okd4-services node:

```
openshift-install --dir=install_dir/ wait-for bootstrap-complete --
log-level=info
```



Once the bootstrap process is complete, which can take upwards of 30 minutes, you can shutdown your bootstrap node. Now is a good time to edit the /etc/haproxy/haproxy.cfg, comment out the bootstrap node, and reload the haproxy service.

```
sudo sed -i '/ okd4-bootstrap /s/^/#/' /etc/haproxy/haproxy.cfg
sudo systemctl reload haproxy
```

Login to the cluster and check the status:

Now that the control-plane node is online, you should be able to login with the oc client. Use the following commands to log in and check the status of your cluster:

```
export KUBECONFIG=~/install_dir/auth/kubeconfig
oc whoami
oc get nodes
```

```
[crobinson@okd4-services ~]$ export KUBECONFIG=~/install_dir/auth/kubeconfig
[crobinson@okd4-services ~]$ oc whoami
system:admin
[crobinson@okd4-services ~]$ oc get nodes
NAME                                 STATUS   ROLES           AGE     VERSION
okd4-control-plane-1.lab.okd.local   Ready    master,worker   7m50s   v1.18.3
```

Check the status of the cluster operators.

```
oc get clusteroperators
```

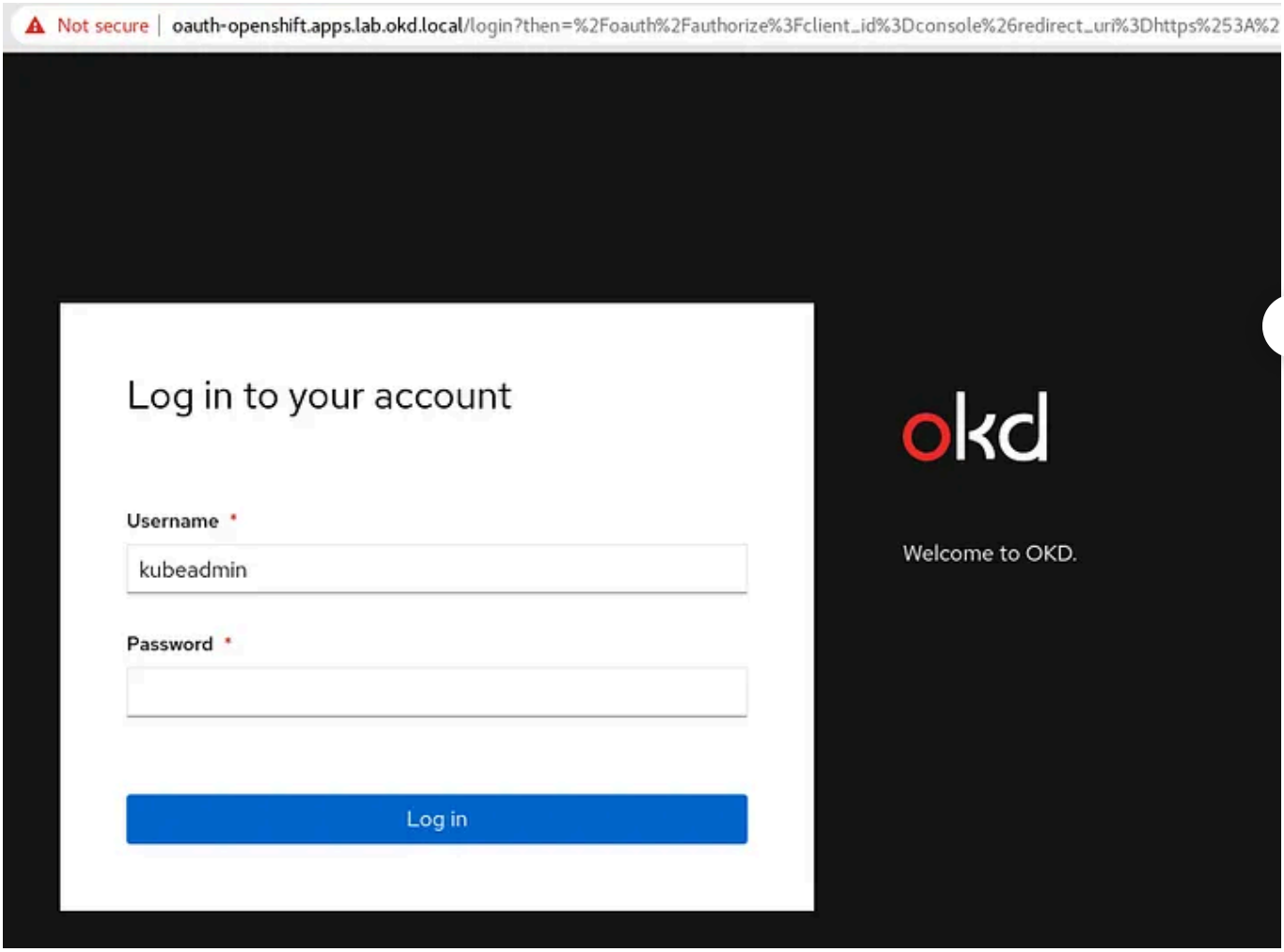
```
[crobinson@okd4-services ~]$ oc get clusteroperators
NAME                                       VERSION                              AVAILABLE   PROGRESSING   DEGRADED   SINCE
authentication                                                                  Unknown     Unknown       True       6m32s
cloud-credential                                                                True        False         False      10m
cluster-autoscaler
config-operator
console
csi-snapshot-controller                    4.5.0-0.okd-2020-07-12-134038-rc     False       True          False      5m7s
dns                                        4.5.0-0.okd-2020-07-12-134038-rc     True        False         False      4m56s
etcd                                       4.5.0-0.okd-2020-07-12-134038-rc     True        False         False      4m52s
image-registry
ingress
insights
kube-apiserver                             4.5.0-0.okd-2020-07-12-134038-rc     True        True          False      4m34s
kube-controller-manager                    4.5.0-0.okd-2020-07-12-134038-rc     True        True          False      4m31s
kube-scheduler                             4.5.0-0.okd-2020-07-12-134038-rc     True        True          False      3m49s
kube-storage-version-migrator              4.5.0-0.okd-2020-07-12-134038-rc     True        False         False      4m14s
machine-api
machine-approver                           4.5.0-0.okd-2020-07-12-134038-rc     True        False         False      4m10s
machine-config                             4.5.0-0.okd-2020-07-12-134038-rc     True        False         False      4m50s
marketplace
monitoring
network                                    4.5.0-0.okd-2020-07-12-134038-rc     True        False         False      7m10s
node-tuning                                4.5.0-0.okd-2020-07-12-134038-rc     True        False         False      6m25s
openshift-apiserver                        4.5.0-0.okd-2020-07-12-134038-rc     False       False         False      6m31s
openshift-controller-manager                                                    False       True          False      6m37s
openshift-samples
operator-lifecycle-manager                 4.5.0-0.okd-2020-07-12-134038-rc     True        True          False      5m27s
operator-lifecycle-manager-catalog         4.5.0-0.okd-2020-07-12-134038-rc     True        True          False      5m28s
operator-lifecycle-manager-packageserver                                        False       True          False      5m27s
service-ca                                 4.5.0-0.okd-2020-07-12-134038-rc     True        False         False      6m18s
storage
[crobinson@okd4-services ~]$ A
```

Once the console operator is available login to the web console.Get your kubeadmin password from the install_dir/auth folder:

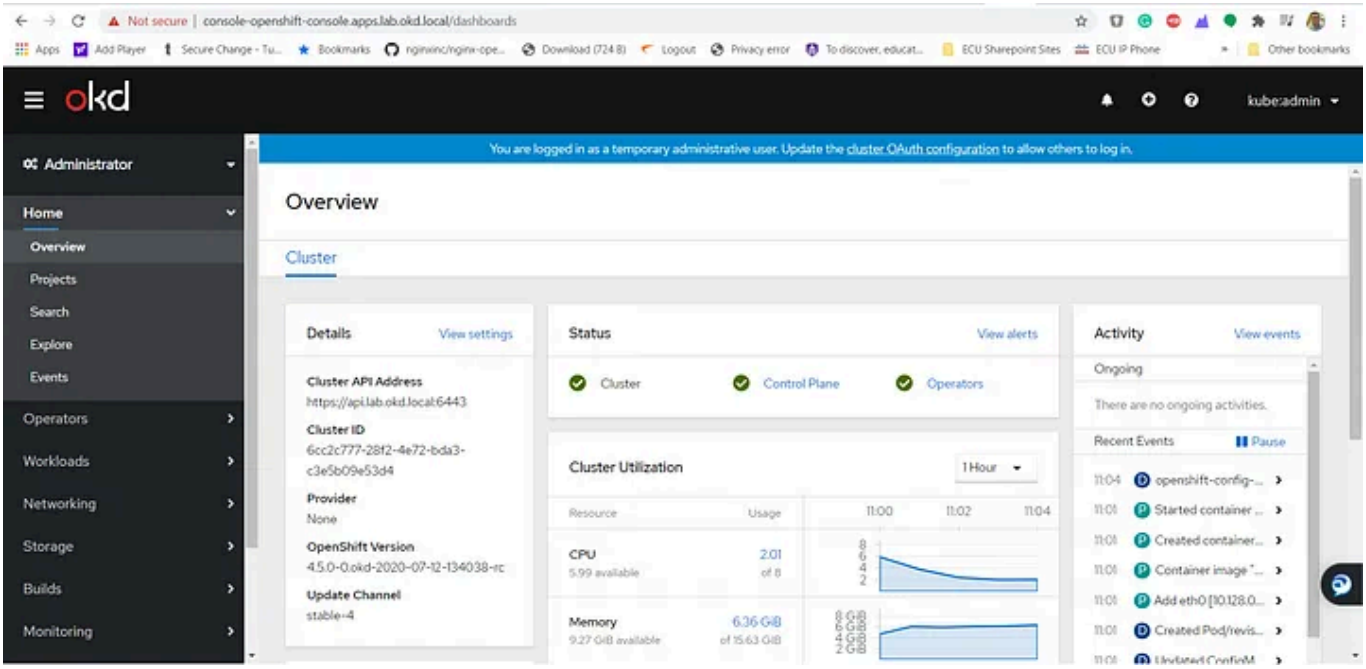
```
cat install_dir/auth/kubeadmin-password
```

```
crobinson@okd4-services:~
File  Edit  View  Search  Terminal  Help
[crobinson@okd4-services ~]$ cat install_dir/auth/kubeadmin-password
xmLBi-xUC26-n3EU5-Mrtxb[crobinson@okd4-services ~]$
```

Open your web browser to https://console-openshift-console.apps.lab.okd.local/ and login as kubeadmin with the password from above:



The cluster status may still say upgrading, and it continues to finish the installation.



Persistent Storage:

We need to create some persistent storage for our registry before we can complete this project. Let's configure our okd4-services VM as an NFS server and use it for persistent storage.

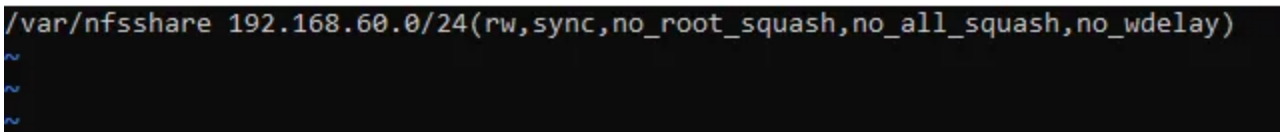
Login to your okd4-services VM and begin to set up an NFS server. The following commands install the necessary packages, enable services, and configure file and folder permissions.

```
sudo dnf install -y nfs-utils
sudo systemctl enable nfs-server rpcbind
sudo systemctl start nfs-server rpcbind
sudo mkdir -p /var/nfsshare/registry
sudo chmod -R 777 /var/nfsshare
sudo chown -R nobody:nobody /var/nfsshare
```

Create an NFS Export

Add this line in the new /etc/exports file “/var/nfsshare 192.168.60.0/24(rw,sync,no_root_squash,no_all_squash,no_wdelay)”

```
echo '/var/nfsshare
192.168.60.0/24(rw,sync,no_root_squash,no_all_squash,no_wdelay)' |
sudo tee /etc/exports
```



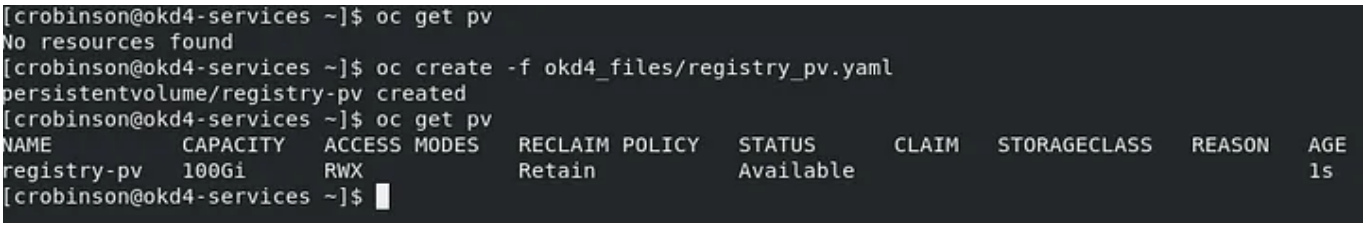
Restart the nfs-server service and add firewall rules:

```
sudo setsebool -P nfs_export_all_rw 1
sudo systemctl restart nfs-server
sudo firewall-cmd --permanent --zone=public --add-service mountd
sudo firewall-cmd --permanent --zone=public --add-service rpc-bind
sudo firewall-cmd --permanent --zone=public --add-service nfs
sudo firewall-cmd --reload
```

Registry configuration:

Create a persistent volume on the NFS share. Use the registry_py.yaml in okd4_files folder from the git repo:

```
oc create -f okd4_files/registry_pv.yaml
oc get pv
```

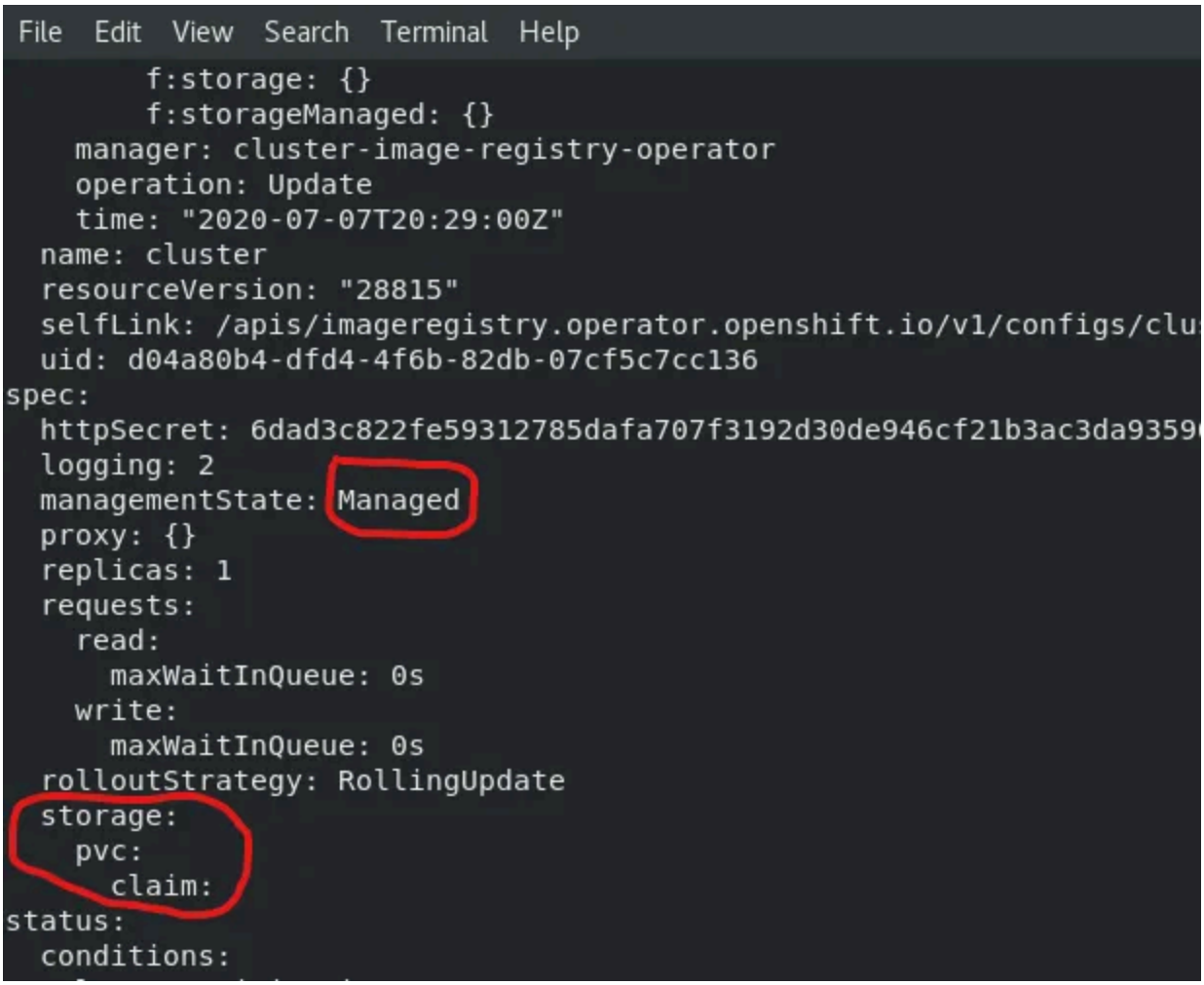


Edit the image-registry operator:

```
oc edit configs.imageregistry.operator.openshift.io
```

Change the managementState: from Removed to Managed. Under storage: add the pvc: and claim: blank to attach the PV and save your changes automatically:

```
managementState: Managedstorage:
    pvc:
      claim:
```



Check your persistent volume, and it should now be claimed:

```
oc get pv
```

```
[crobinson@okd4-services ~]$ oc get pv
NAME          CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
                        STORAGECLASS   REASON   AGE
registry-pv   100Gi      RWX            Retain           Bound    openshift-image-registry/i
mage-registry-storage                            5m19s
[crobinson@okd4-services ~]$
```

Check the export size, and it should be zero. In the next section, we will push to the registry, and the file size should not be zero.

```
du -sh /var/nfsshare/registry
```

```
[crobinson@okd4-services ~]$ du -sh /var/nfsshare/registry/
0       /var/nfsshare/registry/
[crobinson@okd4-services ~]$
```

In the next section, we will create a WordPress project and push it to the registry. After the push, the NFS export should show 200+ MB.

Create WordPress Project:

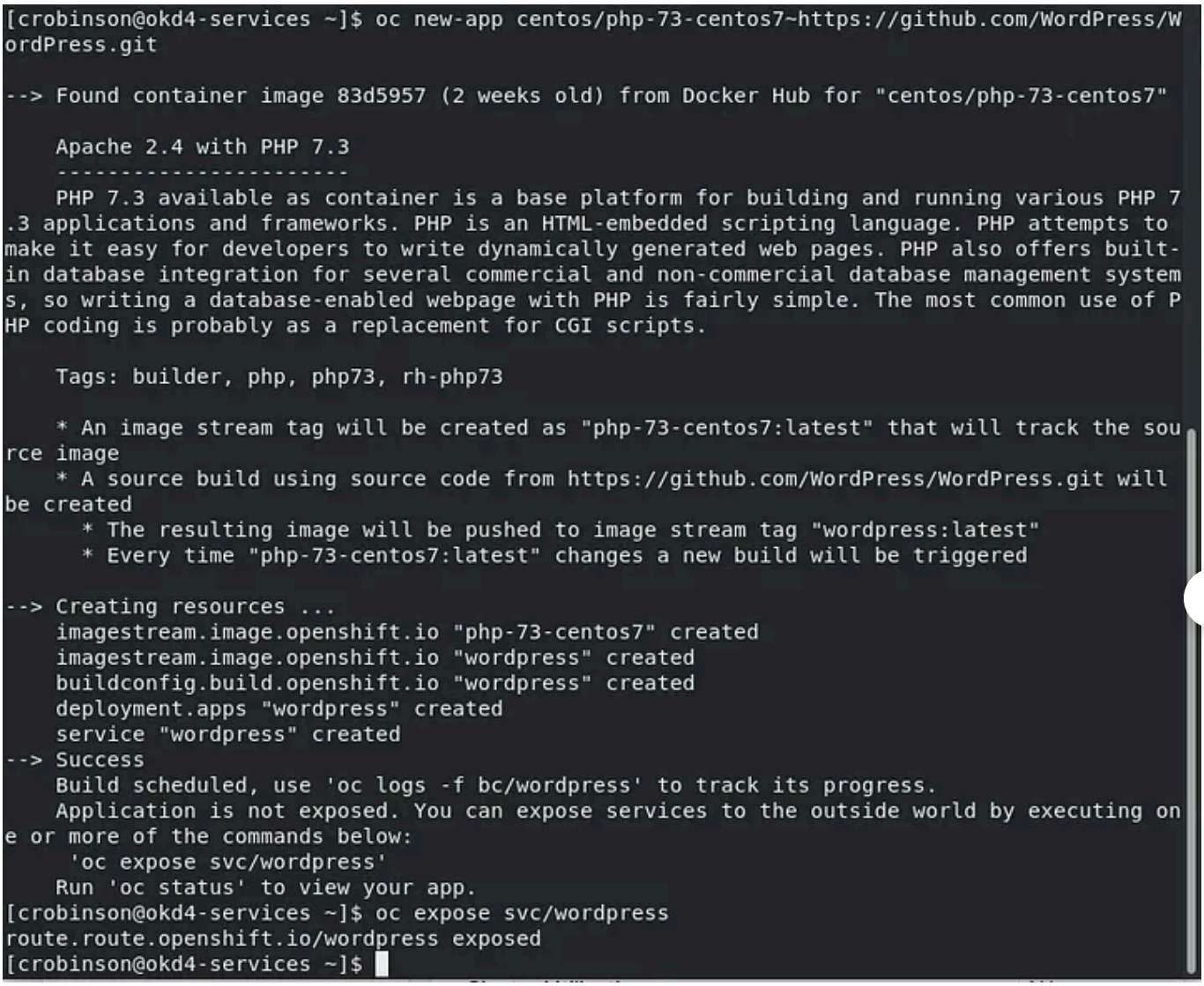
Create a new OKD project.

```
oc new-project wordpress-test
```

```
[crobinson@okd4-services ~]$ oc new-project wordpress-test
Now using project "wordpress-test" on server "https://api.lab.okd.local:6443".

You can add applications to this project with the 'new-app' command. For example, try:

    oc new-app ruby~https://github.com/sclorg/ruby-ex.git

to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes app
lication:

    kubectl create deployment hello-node --image=gcr.io/hello-minikube-zero-install/hello-no
de

[crobinson@okd4-services ~]$
```

Create a new app using the centos php73 s2i image from docker hub and use the WordPress GitHub repo for the source. Expose the service to create a route.

```
oc new-app centos/php-73-
centos7~https://github.com/WordPress/WordPress.git
oc expose svc/wordpress
```

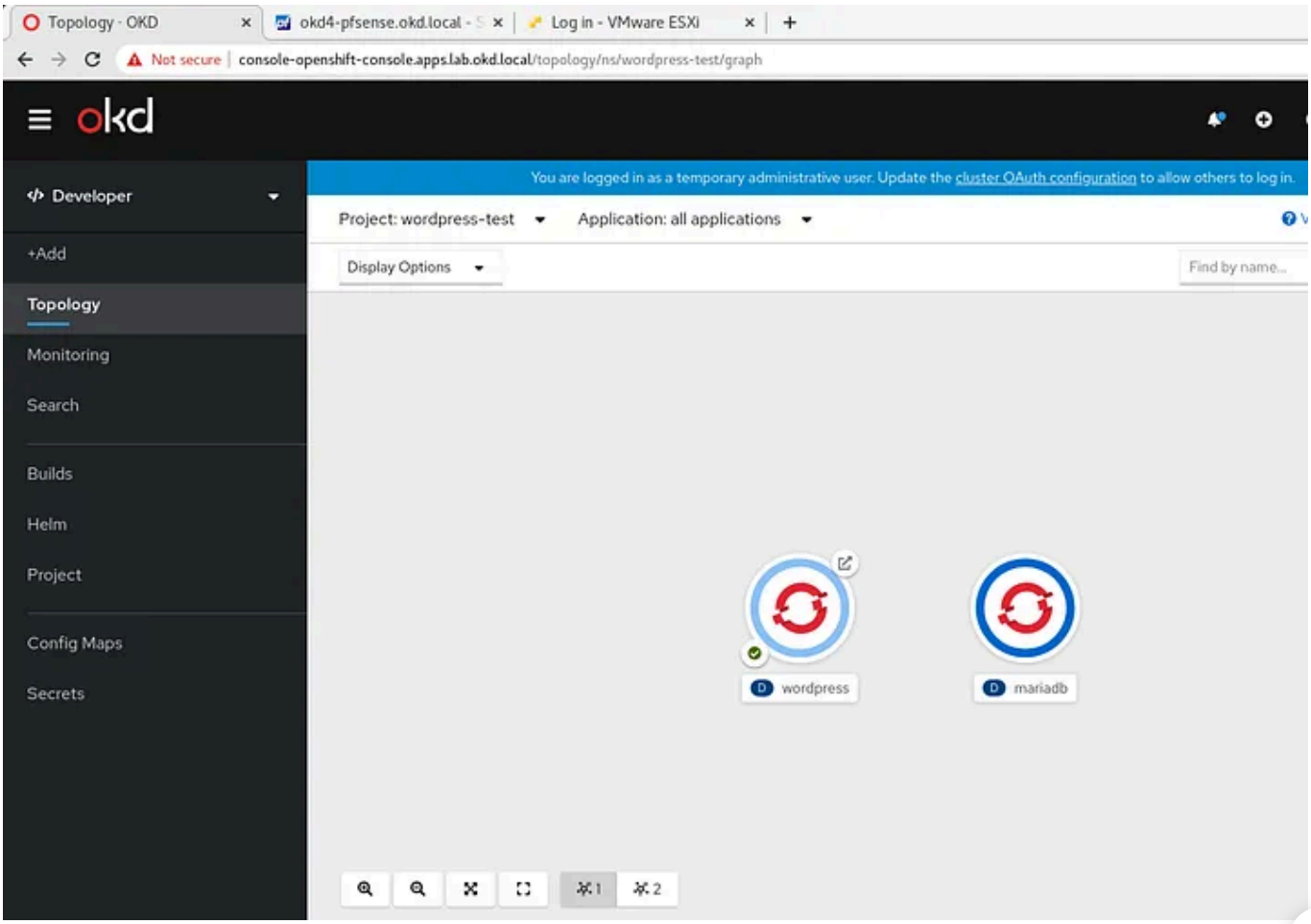


Create a new app using the centos7 MariaDB image with some environment variables:

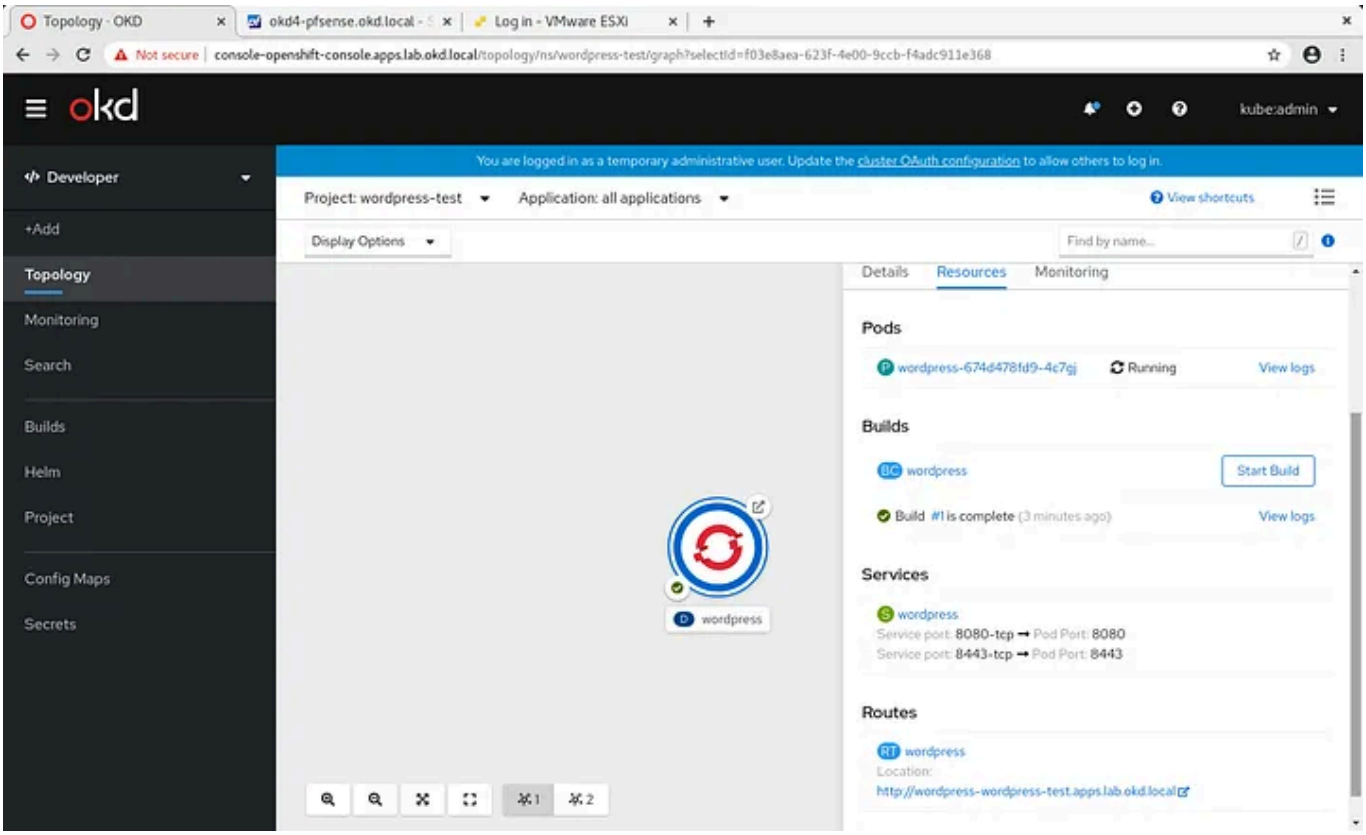
```
oc new-app centos/mariadb-103-centos7 --name mariadb --env
MYSQL_DATABASE=wordpress --env MYSQL_USER=wordpress --env
MYSQL_PASSWORD=wordpress
```

```
[crobinson@okd4-services ~]$ oc new-app centos/mariadb-103-centos7 --name mariadb --env MYSQ
L_DATABASE=wordpress --env MYSQL_USER=wordpress --env MYSQL_PASSWORD=wordpress
--> Found container image 000778a (4 days old) from Docker Hub for "centos/mariadb-103-cento
s7"

    MariaDB 10.3
    ------------
    MariaDB is a multi-user, multi-threaded SQL database server. The container image provide
s a containerized packaging of the MariaDB mysqld daemon and client application. The mysqld
server daemon accepts connections from clients and provides access to content from MariaDB d
atabases on behalf of the clients.

    Tags: database, mysql, mariadb, mariadb103, rh-mariadb103, galera

    * An image stream tag will be created as "mariadb:latest" that will track this image

--> Creating resources ...
    imagestream.image.openshift.io "mariadb" created
    deployment.apps "mariadb" created
    service "mariadb" created
--> Success
    Application is not exposed. You can expose services to the outside world by executing on
e or more of the commands below:
     'oc expose svc/mariadb'
    Run 'oc status' to view your app.
[crobinson@okd4-services ~]$
```

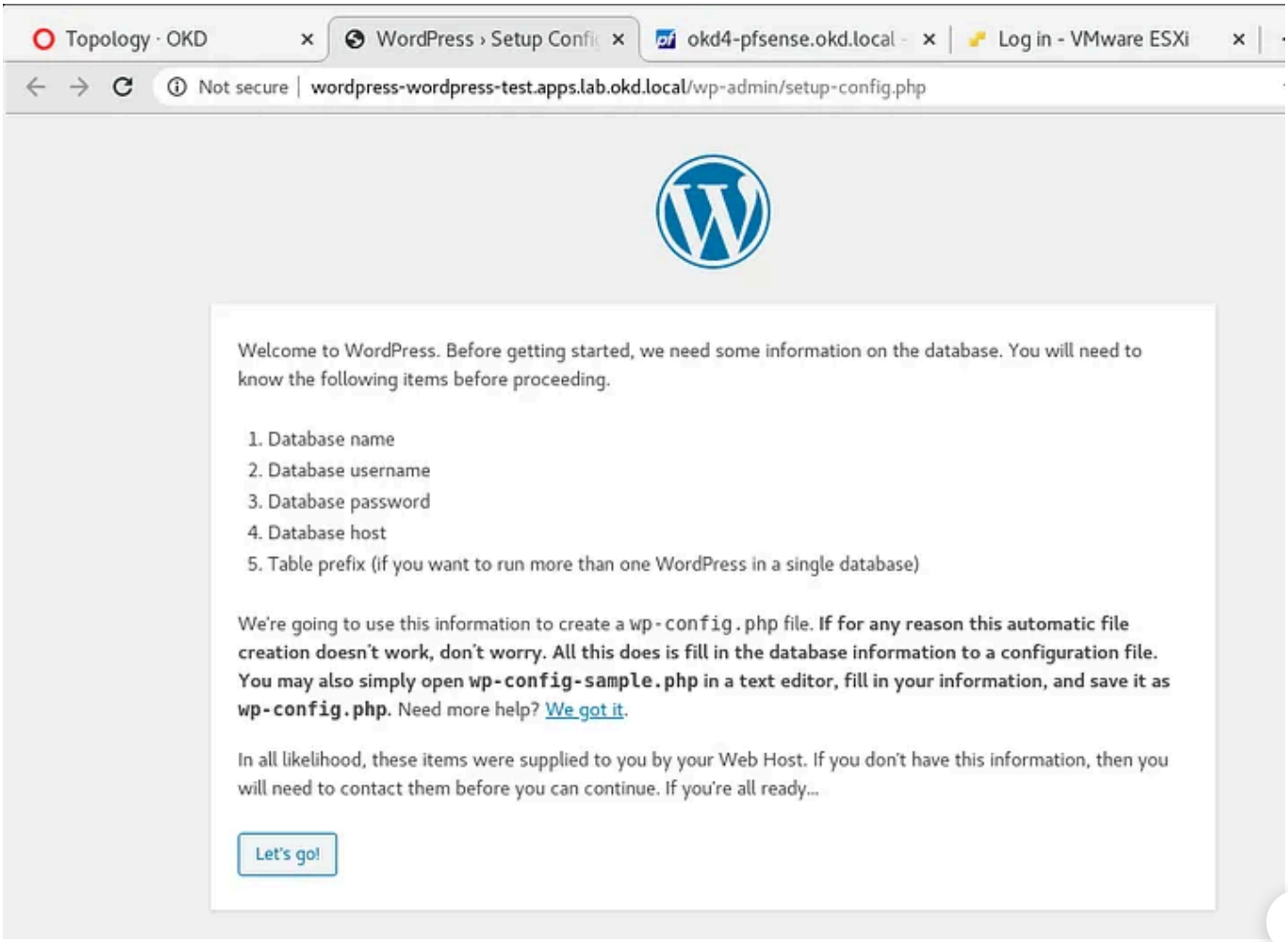
Open the OpenShift console and browse to the WordPress-test project. Once the WordPress image is built and ready, it will be dark blue like the MariaDB instance show here:



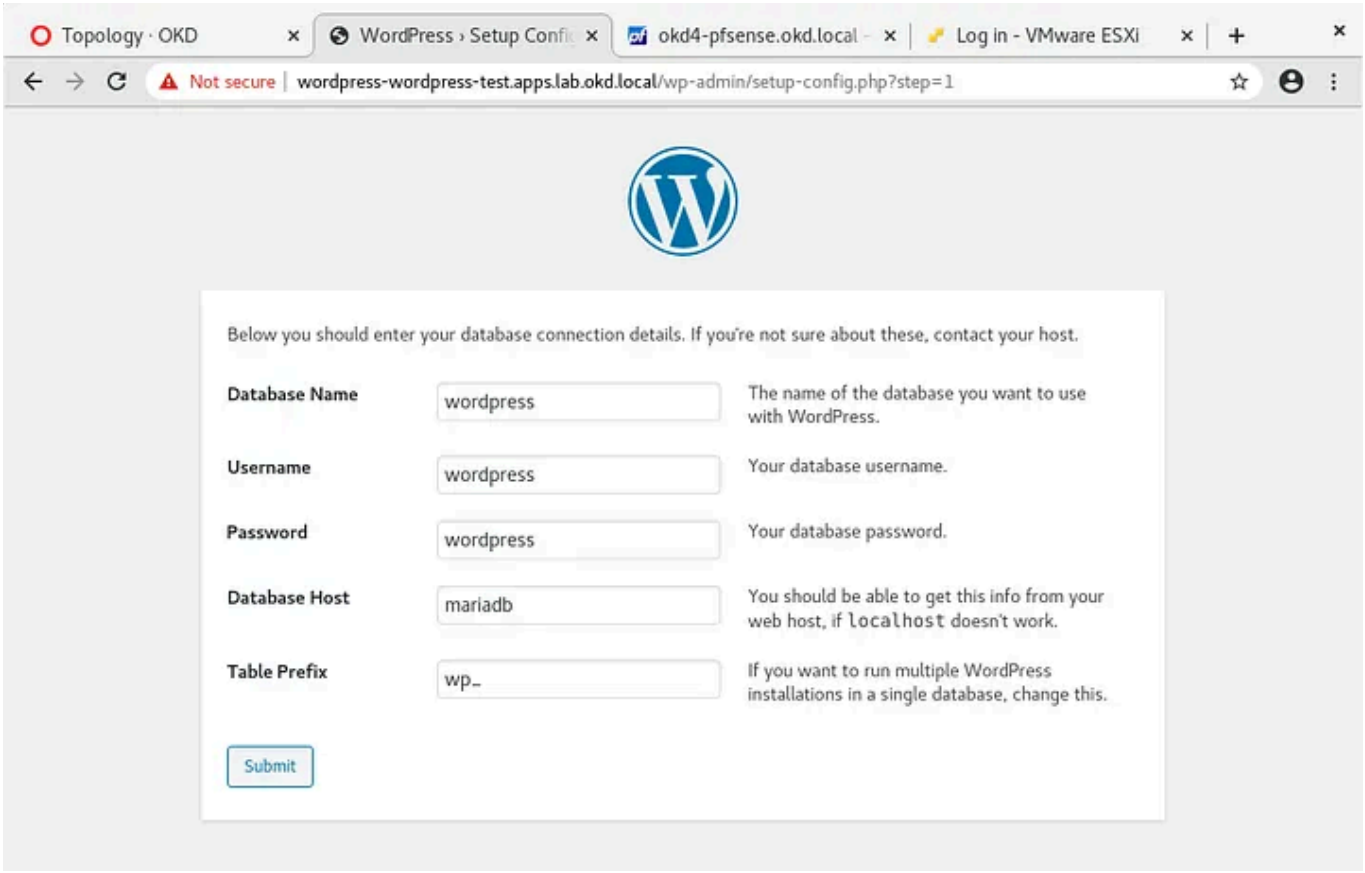
Click on the WordPress object and click on the route to open it in your web browser:



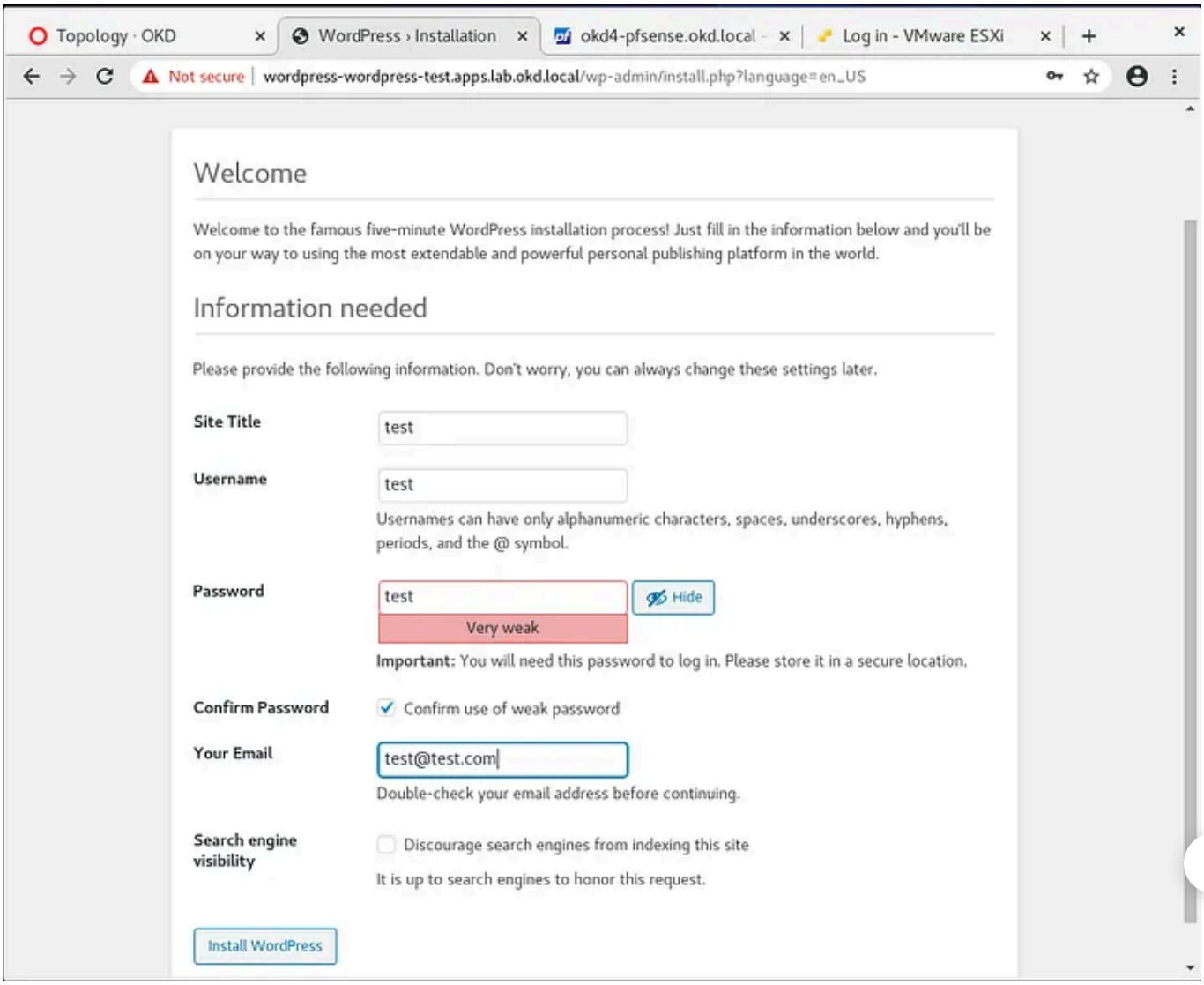
You should see the WordPress setup config, click Let's Go.



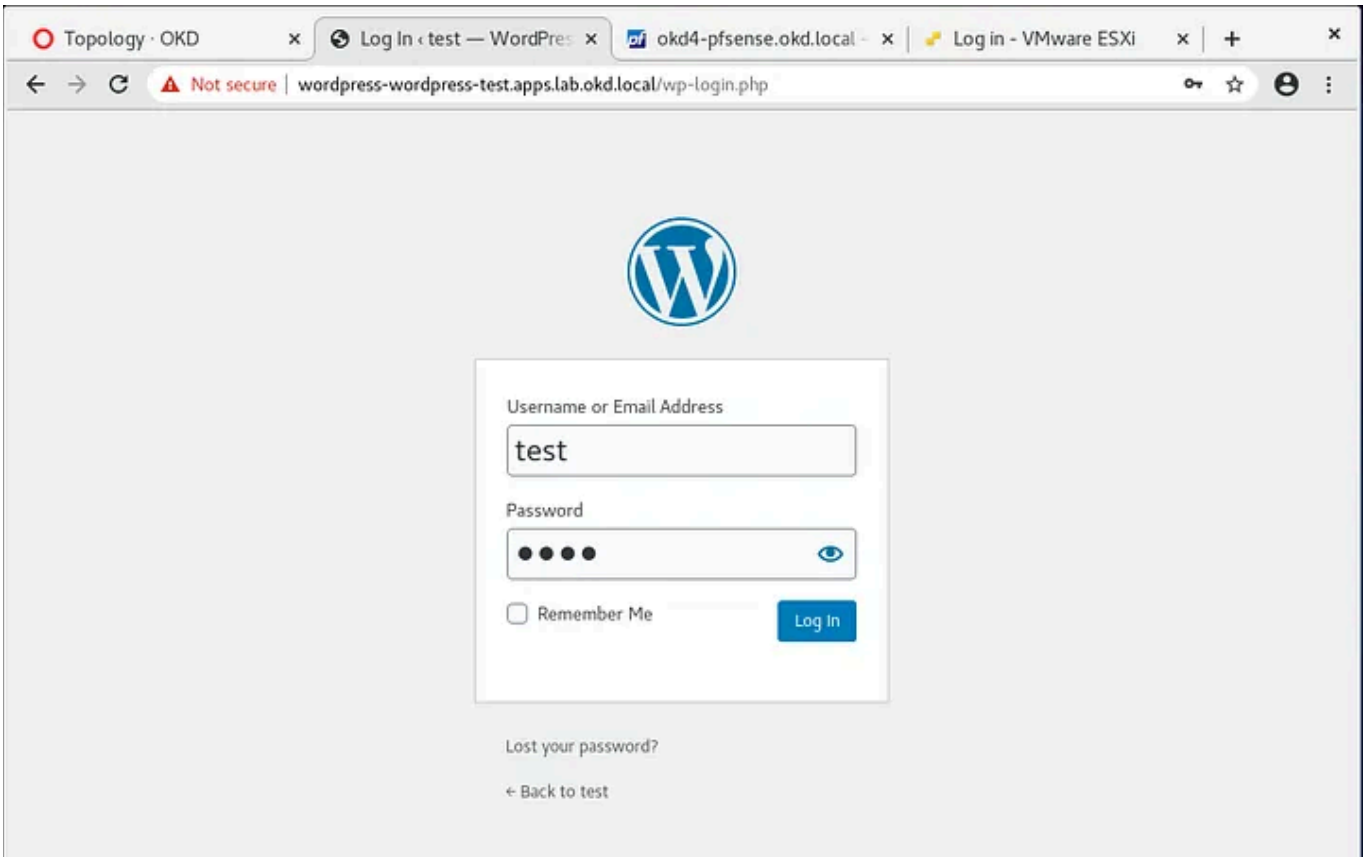
Fill in the database, username, password, and database host as pictured and run the installation:

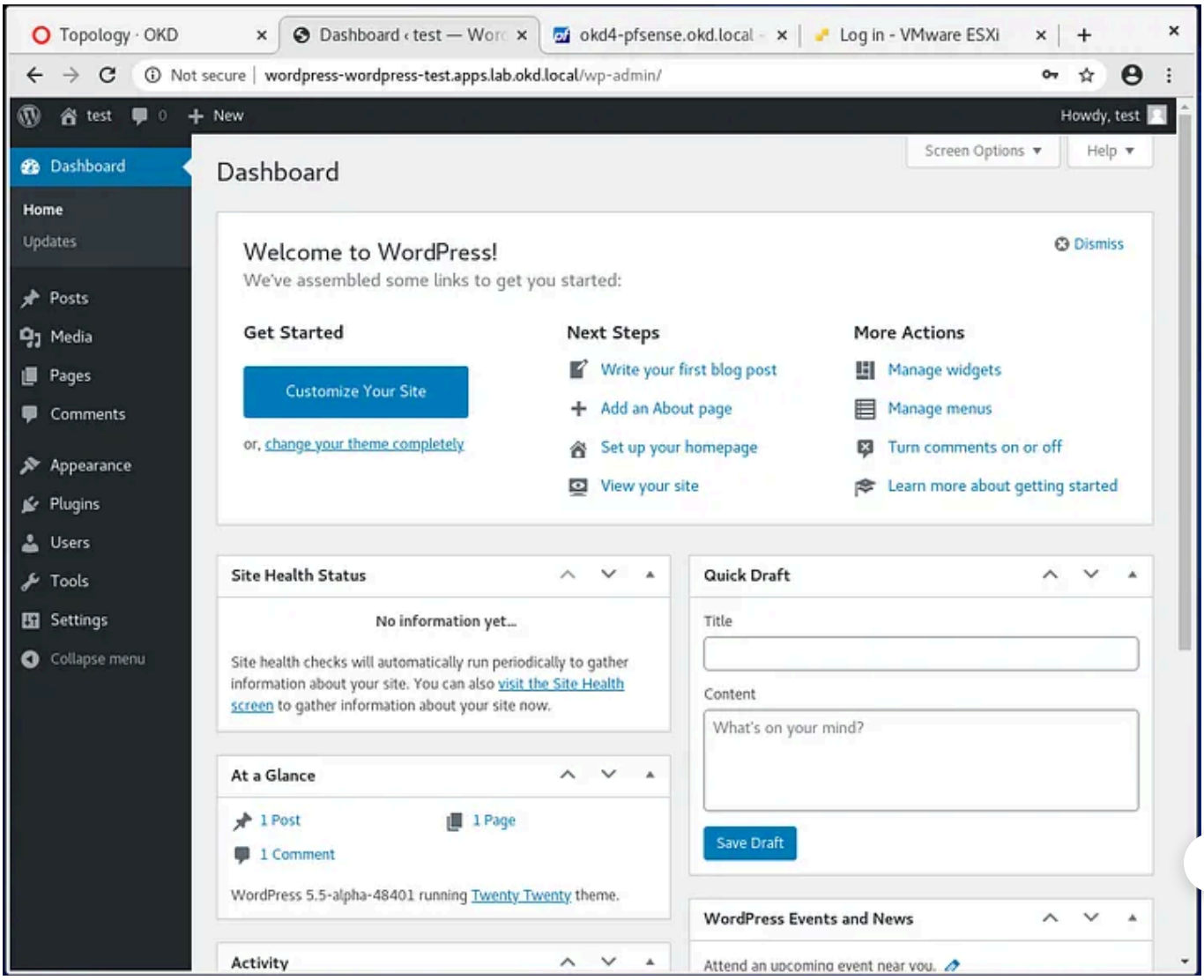


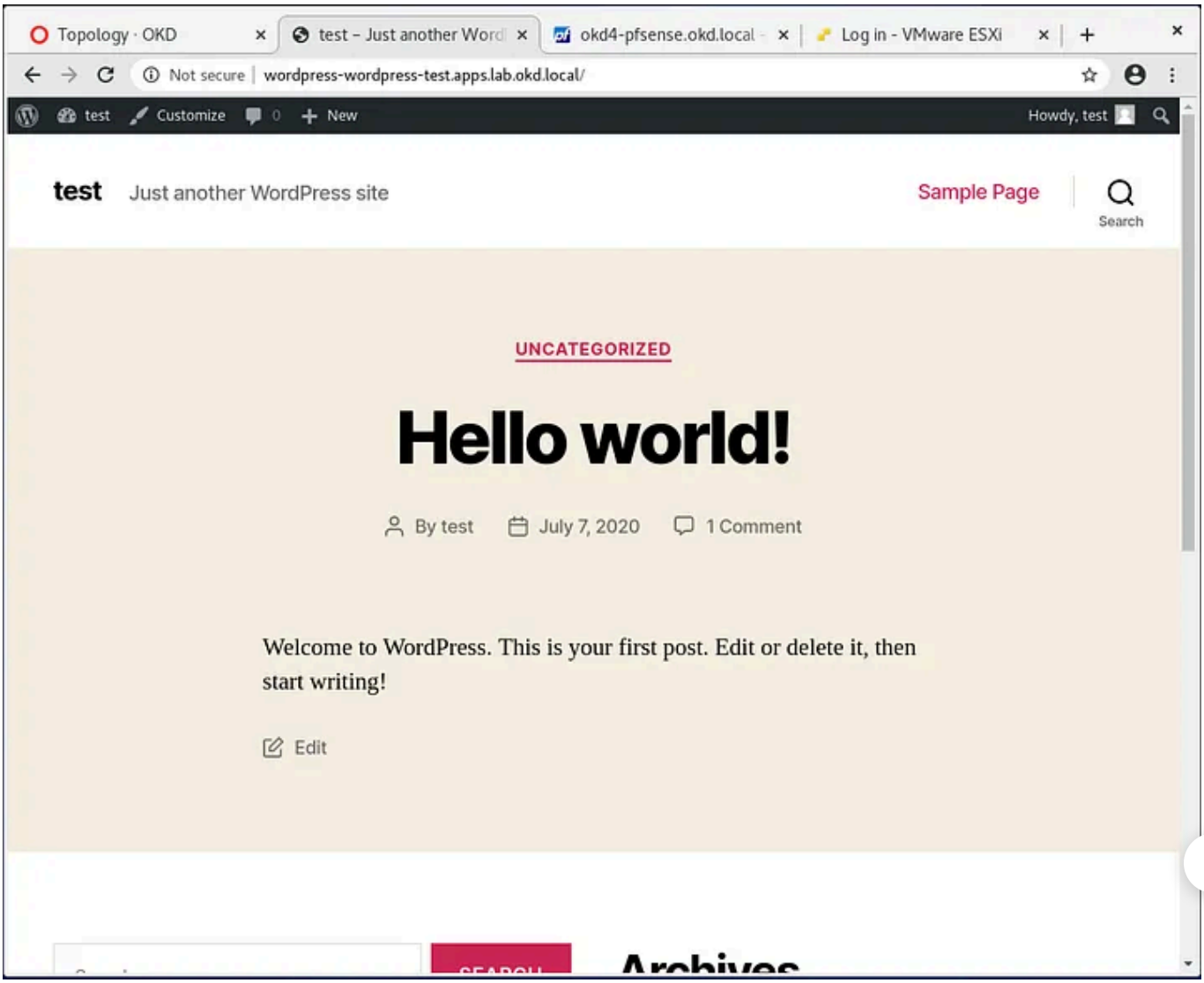
Fill out the welcome information and click Install WordPress.



Log in, and you should have a working WordPress installation:


Topology · OKD
Log In ‹ test — WordPres
okd4-pfsense.okd.local
Log in - VMware ESXi
Not secure | wordpress-wordpress-test.apps.lab.okd.local/wp-login.php
Username or Email Address
test
Password
Remember Me
Log In
Lost your password?
← Back to test


Topology · OKD
Dashboard ‹ test — Wor
okd4-pfsense.okd.local
Log in - VMware ESXi
Not secure | wordpress-wordpress-test.apps.lab.okd.local/wp-admin/
test
0
New
Howdy, test
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Screen Options
Help
Welcome to WordPress!
We've assembled some links to get you started:
Dismiss
Get Started
Customize Your Site
or, change your theme completely
Next Steps
Write your first blog post
Add an About page
Set up your homepage
View your site
More Actions
Manage widgets
Manage menus
Turn comments on or off
Learn more about getting started
Site Health Status
No information yet...
Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information about your site now.
At a Glance
1 Post
1 Page
1 Comment
WordPress 5.5-alpha-48401 running Twenty Twenty theme.
Activity
Quick Draft
Title
Content
What's on your mind?
Save Draft
WordPress Events and News
Attend an upcoming event near you.



Check the size of your NFS export on the okd4-services VM. It should be around 300MB in size.

```
du -sh /var/nfsshare/registry/
```

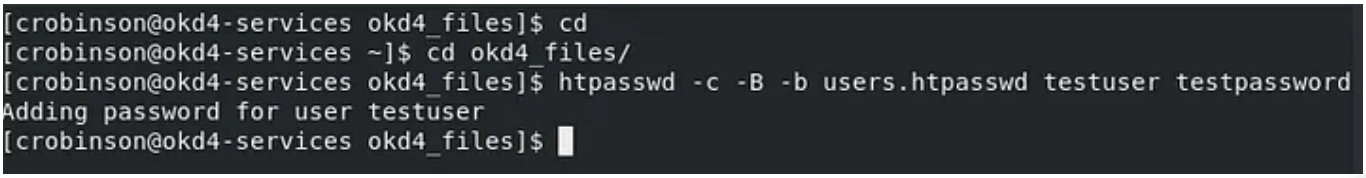
```
[crobinson@okd4-services ~]$ du -sh /var/nfsshare/registry/
272M    /var/nfsshare/registry/
[crobinson@okd4-services ~]$
```

You have just verified your persistent volume is working.

HTPasswd Setup:

The kubeadmin is a temporary user. The easiest way to set up a local user is with htpasswd.

```
cd
cd okd4_files
htpasswd -c -B -b users.htpasswd testuser testpassword
```



Create a secret in the openshift-config project using the users.htpasswd file you generated:

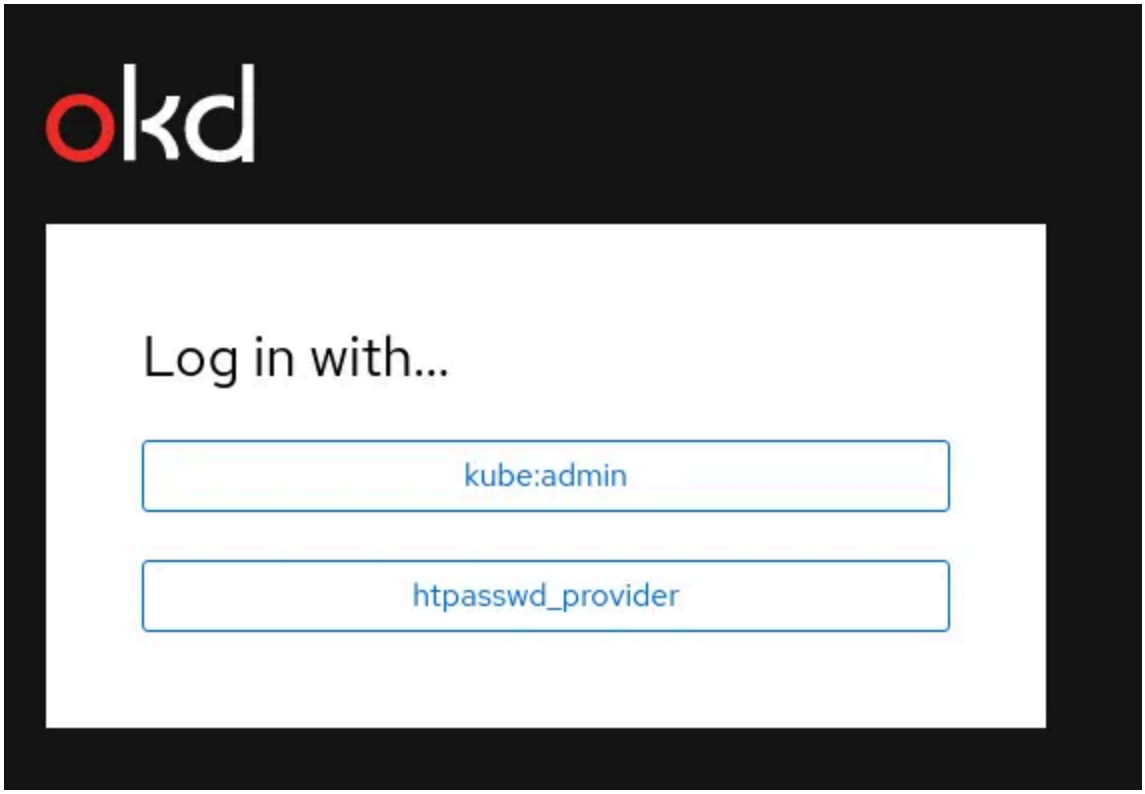
```
oc create secret generic htpass-secret --from-
file=htpasswd=users.htpasswd -n openshift-config
```

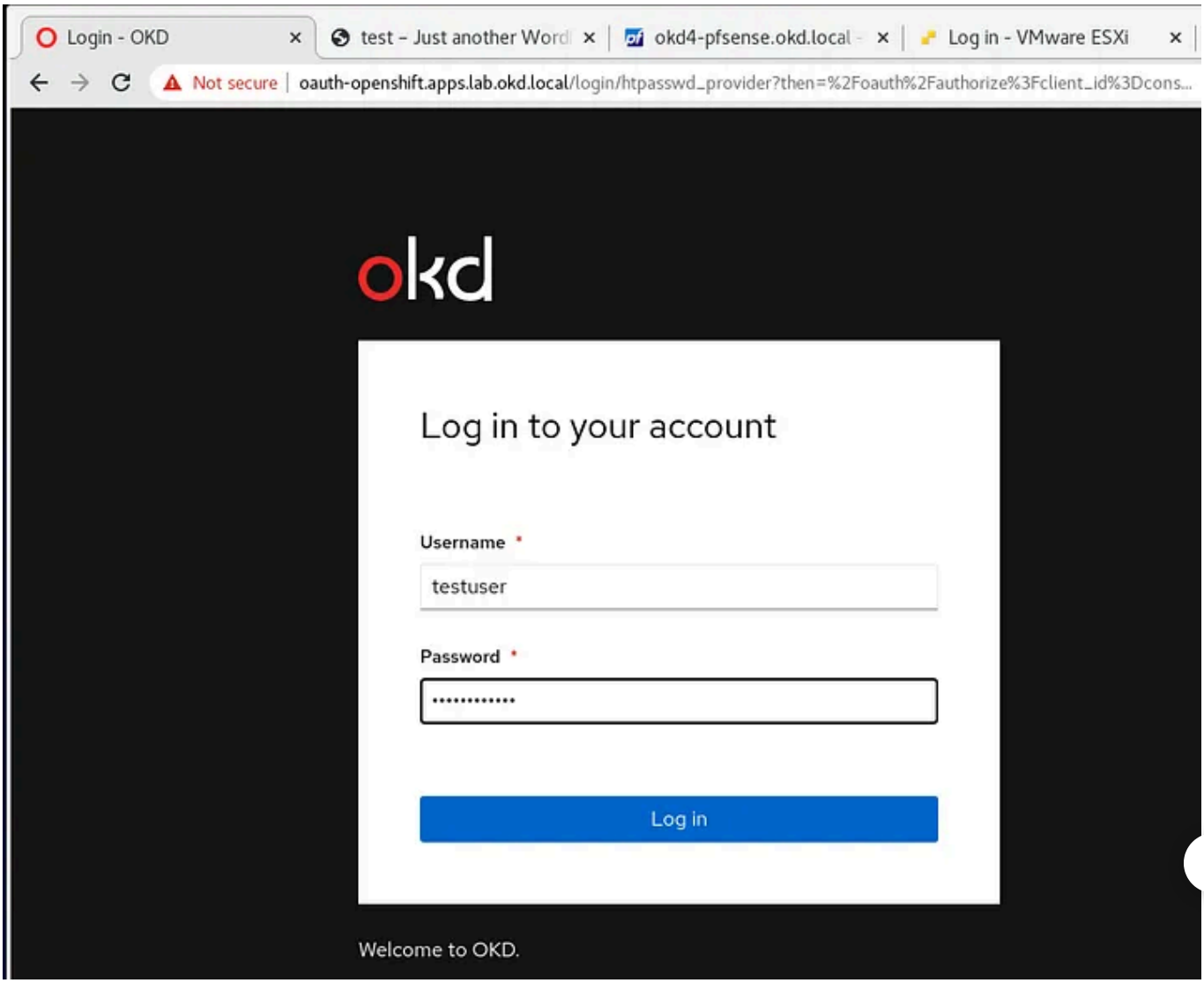
Add the identity provider.

```
oc apply -f htpasswd_provider.yaml
```

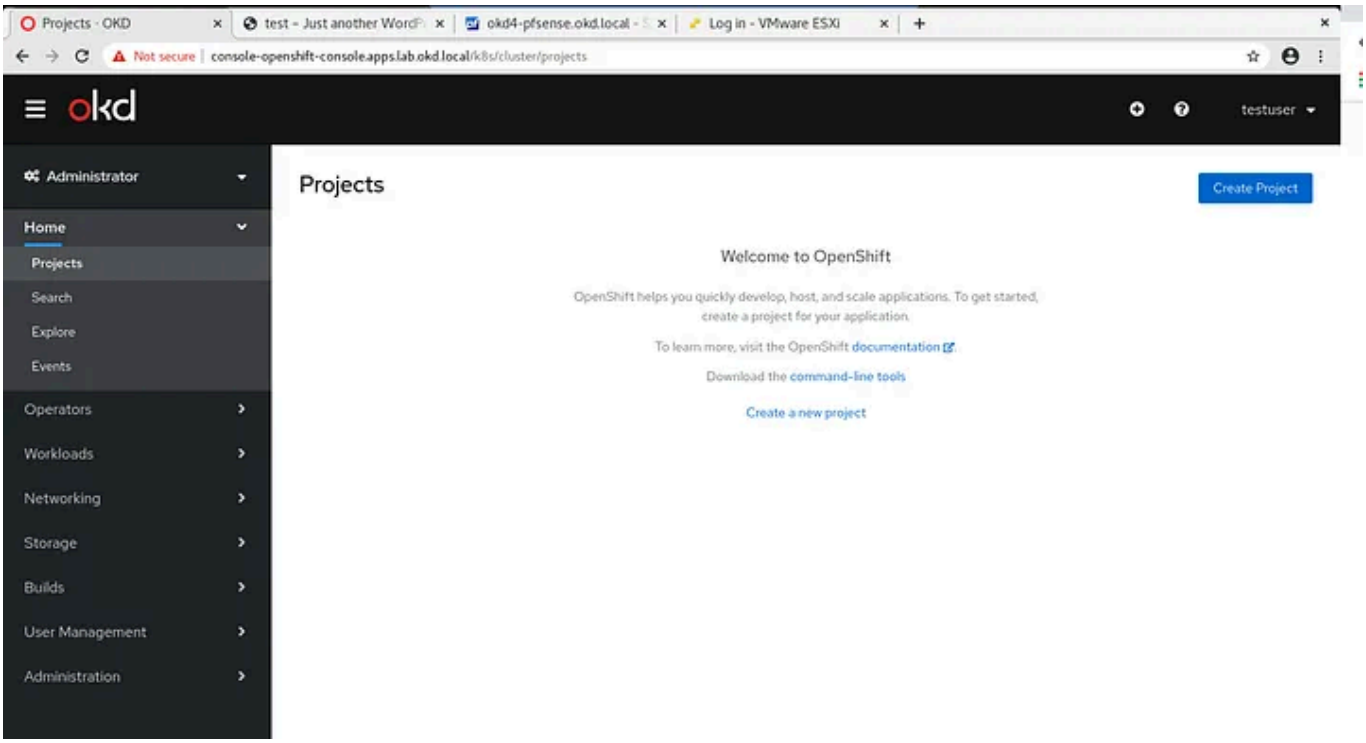
```
[crobinson@okd4-services okd4_files]$ oc apply -f htpasswd_provider.yaml
Warning: oc apply should be used on resource created by either oc create --save-config or oc
 apply
oauth.config.openshift.io/cluster configured
[crobinson@okd4-services okd4_files]$
```

Logout of the OpenShift Console. Then select htpasswd_provider and login with testuser and testpassword credentials.





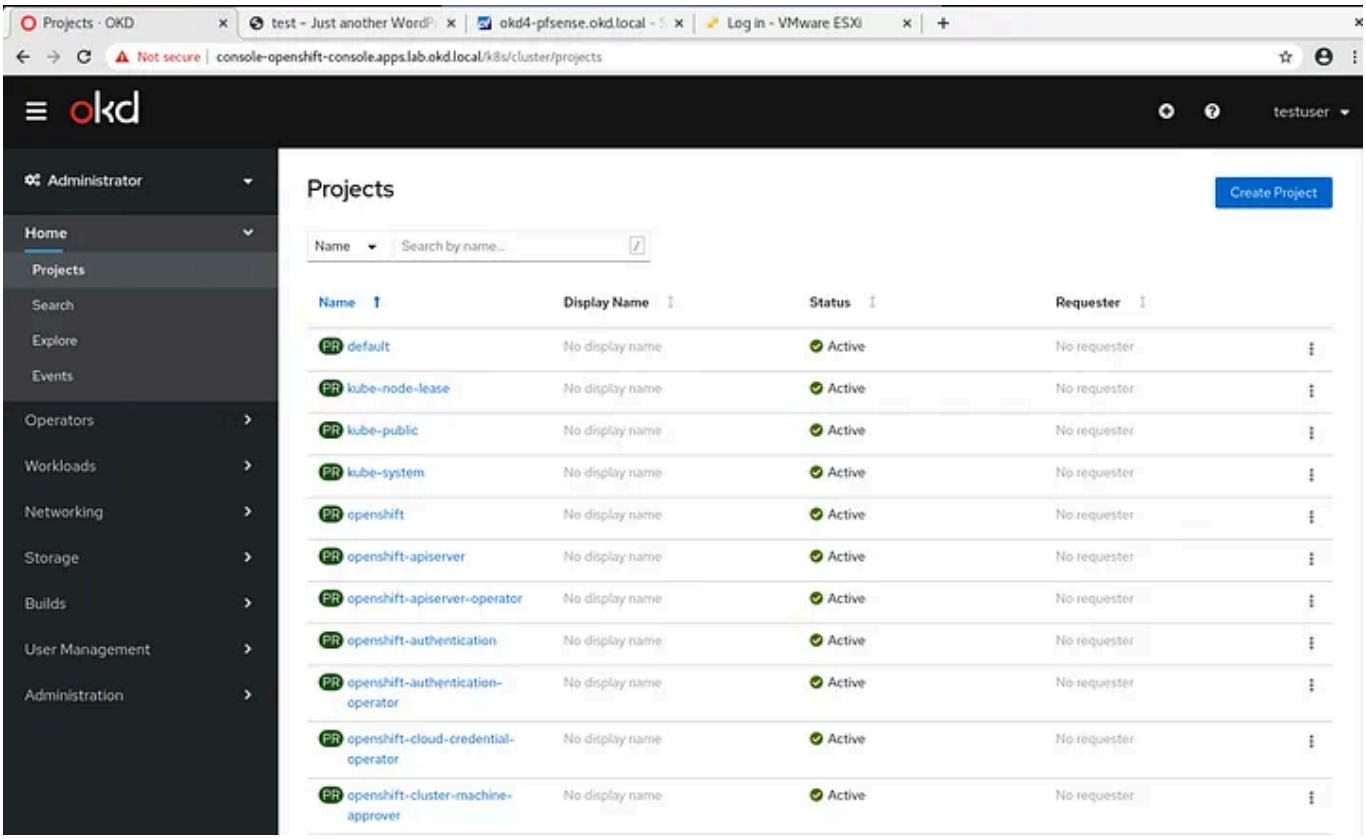
If you visit the Administrator page you should see no projects:



Give yourself cluster-admin access, and the projects should immediately populate:

```
oc adm policy add-cluster-role-to-user cluster-admin testuser
```

Your user should now have cluster-admin level access:



Congrats! You have created an OKD Single Node Cluster!

Hopefully, you have created an OKD single node cluster and learned a few things along the way. At this point, you should have a decent basis to tinker with OKD and continue to learn.

Here are some resources available to help you along your journey:

To report issues, use the OKD Github Repo: https://github.com/openshift/okd

For support check out the #openshift-users channel on k8s Slack

The OKD Working Group meets bi-weekly to discuss the development and next steps. The meeting schedule and location are tracked in the openshift/community repo.

Google group for okd-wg: https://groups.google.com/forum/#!forum/okd-wg

Openshift Okd Kubernetes WordPress Cluster

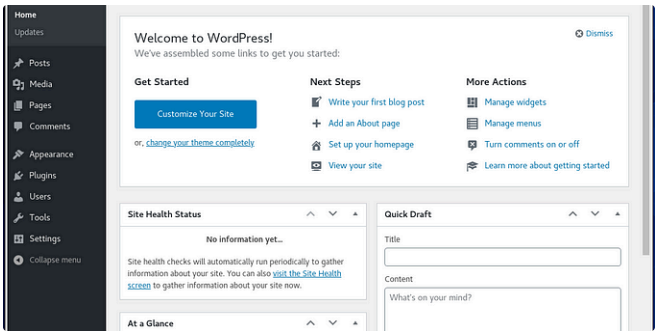
Written by Craig Robinson

Follow

517 Followers · Writer for The Startup

OpenShift Enthusiast

More from Craig Robinson and The Startup



Craig Robinson in ITNEXT

Guide: Installing an OKD 4.5 Cluster

Updated 7/7/2020

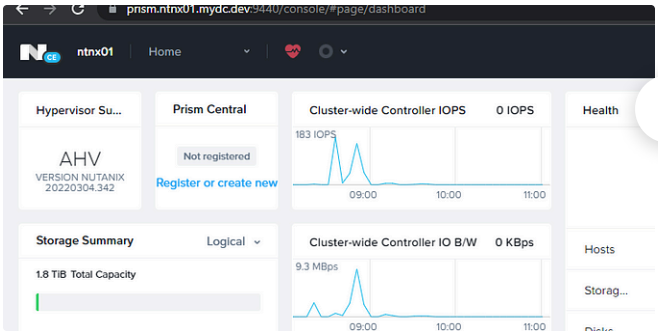
Jul 8, 2020 507 53

Onyedikachukwu Czar in The Startup

I Asked Google Gemini To Come Up With Business Ideas Not Found...

How to use AI to iterate ideas and come up with what stands out.

Jul 12 2.1K 42



Greyson Ferguson in The Startup

Countries You Can Move To Indefinitely Without a Visa

Why deal with all of that visa paperwork when you could just move without filling out a sing...

Jul 4 4.8K 56

Craig Robinson

Installing Prism Central on Nutanix Community Edition 2.0

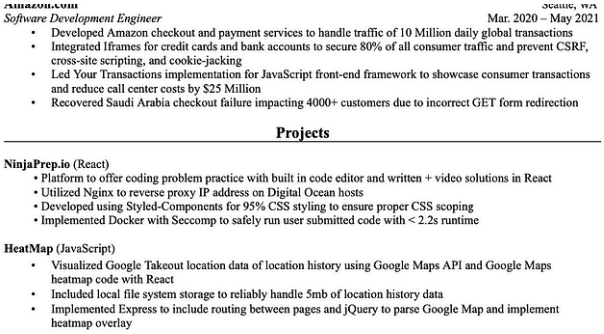
Installing Prism Central

Mar 7, 2023 2

See all from Craig Robinson

See all from The Startup

Recommended from Medium



Software Development Engineer Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by $25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects

NinjaPrep.io (React)
- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)
- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay

Alexander Nguyen in Level Up Coding

The resume that got a software engineer a $300,000 job at Google.

1-page. Well-formatted.

Jun 1 16.1K 252

Michelle Teheux in Minds Without Borders

We Could Learn a Lot About Sex From the Dutch

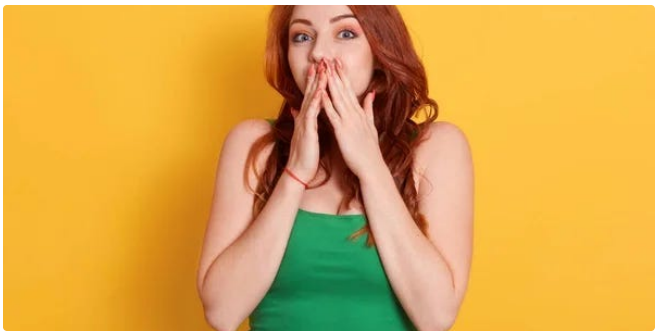
My Dutch relative's views shocked me, but I immediately realized she was right

Jul 18 25K 299

Lists

Natural Language Processing

1618 stories · 1187 saves



Samanta Writes in ILLUMINATION

12 Signs of a Highly Unintelligent Person According to Science

3. Nodding Your Head

Jul 19 8.8K 236

Unbecoming

10 Seconds That Ended My 20 Year Marriage

It's August in Northern Virginia, hot and humid. I still haven't showered from my...

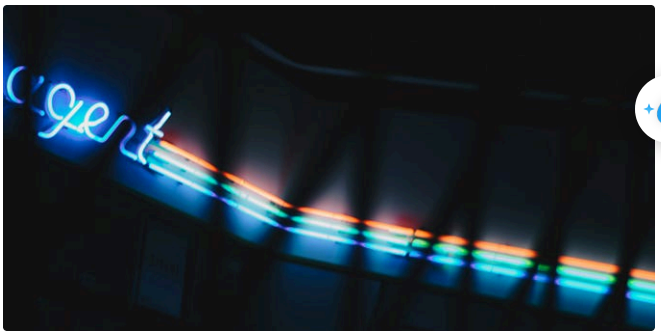
Feb 16, 2022 83K 1139

Derek Johnson

I'm Unemployed for Over Two Years (as a software engineer)

In 2022, I worked on a contract as a software engineer at Apple. Apple dissolved our entir...

Jun 1 5.7K 143



kapil rajyaguru

Agent-Based Red Hat OpenShift Cluster Install

With OpenShift 4.11, Red Hat introduced a new agent-based installer for OpenShift to the...

Apr 25

See more recommendations