

You have **2** free stories left this month. [Sign up and get an extra one!](#)



Войдите в приложение "medium.com" с аккаунтом Google X



Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

[Продолжить как Sviatoslav](#)

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".

OKD 4.4 Bare Metal Install VMWare Home Lab



Craig Robinson [Follow](#)
Mar 3 · 15 min read ★

Updated: 5/26/2020

This guide is deprecated. Please use the new guide for OKD 4.5 at https://medium.com/@craig_robinson/guide-installing-an-okd-4-5-cluster-508a2631cbbe

OKD is the upstream and community-supported version of the Red Hat OpenShift Container Platform (OCP). OpenShift expands vanilla Kubernetes into an application platform designed for enterprise use at scale. Starting with the release of OpenShift 4, the default operating system is Red Hat CoreOS, which provides an immutable infrastructure and automated updates. Fedora CoreOS, like OKD, is the upstream version of Red Hat CoreOS.

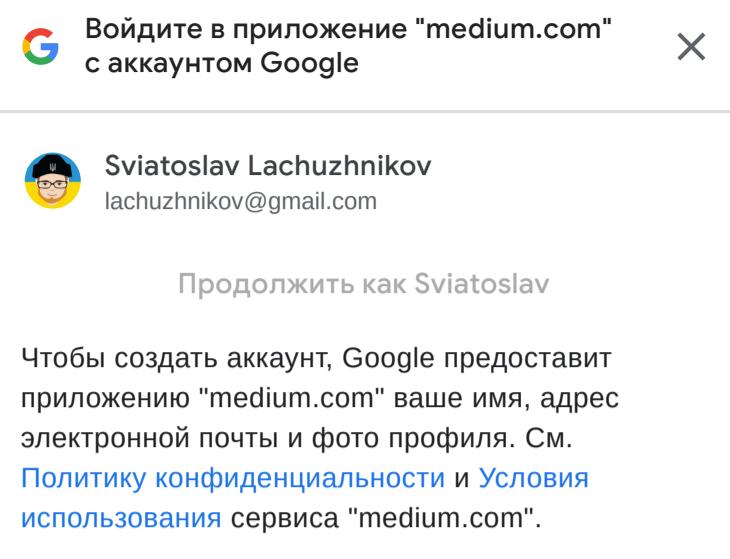
For those of you who have a Home Lab, you can gain some experience with OpenShift by using the open-source upstream combination of OKD and FCOS (Fedora CoreOS) to build your own cluster.

Experience is an excellent way to learn new technologies. Used hardware for a home lab that could run an OKD cluster is relatively inexpensive these days (\$250–\$350), especially when compared to a cloud-hosted solution costing over \$250 per month.

The purpose of this guide is to help you successfully build an OKD 4.4 cluster at home that you can take for a test drive. VMWare is the hypervisor used in this guide, but you could use Hyper-V, VirtualBox, bare metal, or other platforms.

This guide assumes you have a virtualization platform, access to Linux, and the ability to Google.

Note: In this tutorial, I use a 192.168.1.0/24 network. If you already use 192.168.1.0/24 be sure to use a different range in your OKD environment. An alternate network would be 192.168.100.0/24 for example.



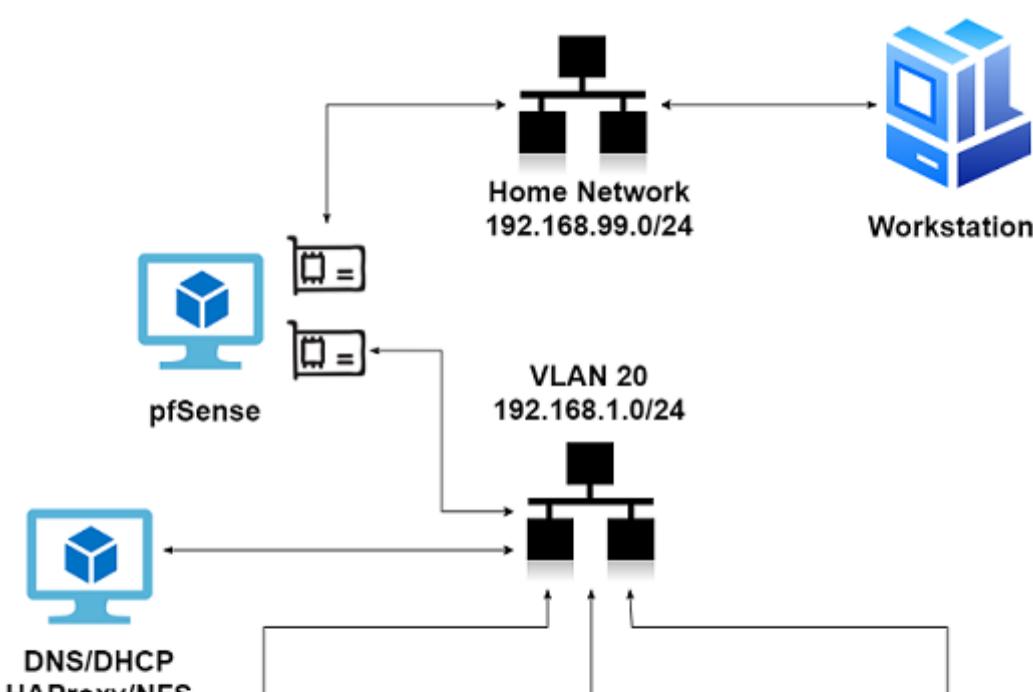
VM Overview:

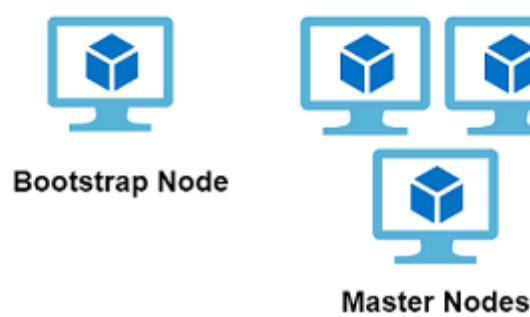
For my installation, I used an ESXi 6.5 host with 96GB of RAM and a separate VLAN configured for OKD. Here is a breakdown of the virtual machines:

Machine	Type	OS	vCPU	RAM	Storage	IP Address
okd4-bootstrap	Bootstrap	Fedora CoreOS	4	16	120	192.168.1.200
okd4-control-plane-1	Master	Fedora CoreOS	4	16	120	192.168.1.201
okd4-control-plane-2	Master	Fedora CoreOS	4	16	120	192.168.1.202
okd4-control-plane-3	Master	Fedora CoreOS	4	16	120	192.168.1.203
okd4-compute-1	Worker	Fedora CoreOS	4	16	120	192.168.1.204
okd4-compute-2	Worker	Fedora CoreOS	4	16	120	192.168.1.205
okd4-services	DNS/LB/Web/NFS	CentOS 7	2	2	20 & 100	192.168.1.210
okd4-pfsense	Router/DHCP	FreeBSD	1	1	8	192.168.1.1

Note: “There are terraform scripts available to setup vSphere-enabled kubelet” -/r/roignac. I chose not to use the scripts as it added an extra layer of complexity, but they are available.

Network Layout





Войдите в приложение "medium.com" с аккаунтом Google X

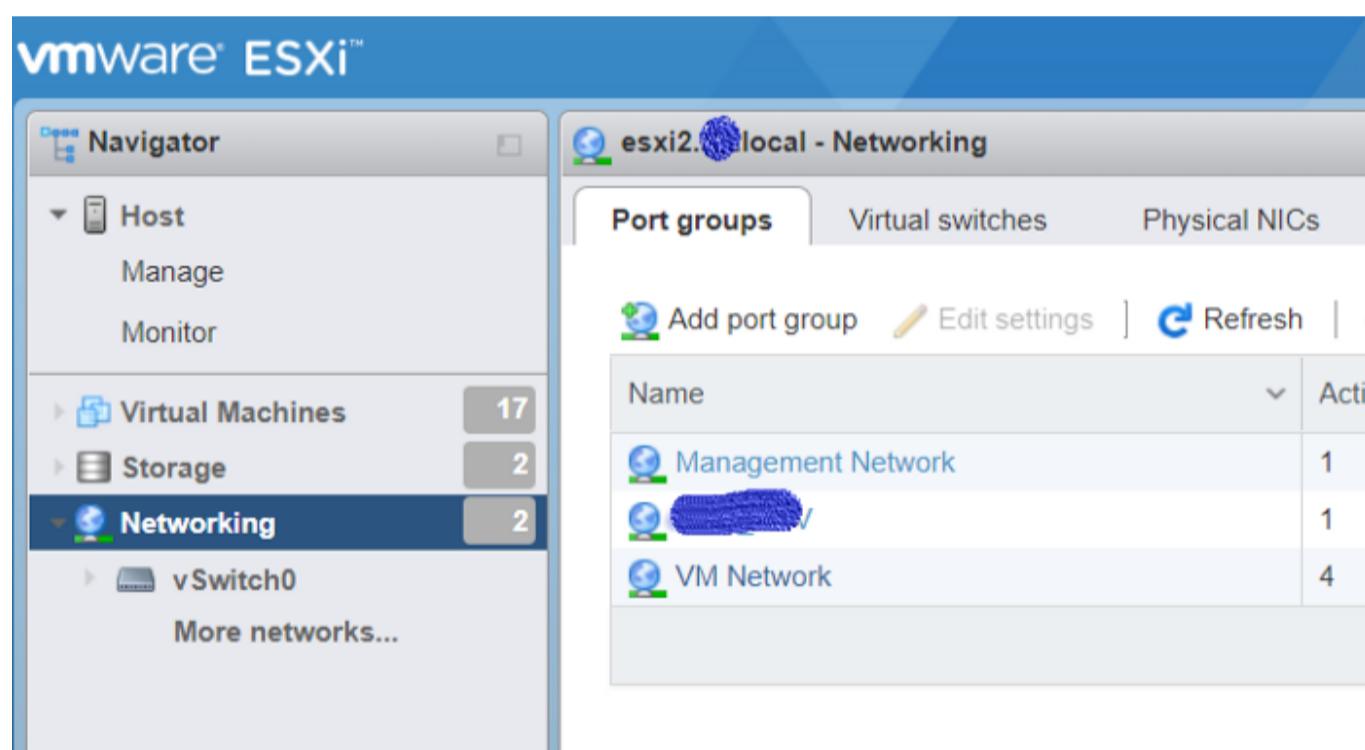
Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".

Create a new network in VMWare for OKD:

Login to your VMWare Host. Select Networking → Port Groups → Add port group. Setup an OKD network on an unused VLAN, in my instance, VLAN 20.



Name	OKD
VLAN ID	20
Virtual switch	vSwitch0
Security Click to expand	

Name your Group and set your VLAN ID.

Create a pfSense VM:

Download the pfSense ISO and upload it to your ESXi host's datastore.

X

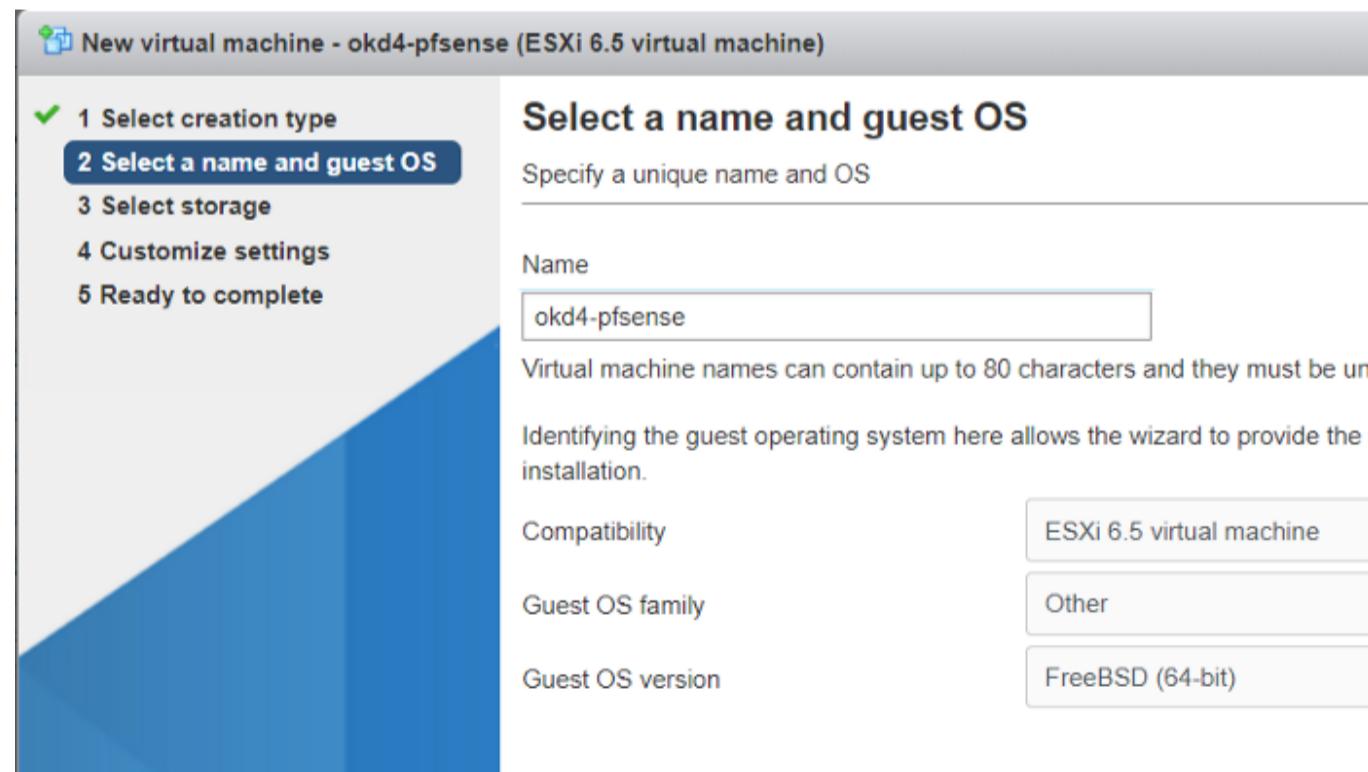
Войдите в приложение "medium.com" с аккаунтом Google

Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

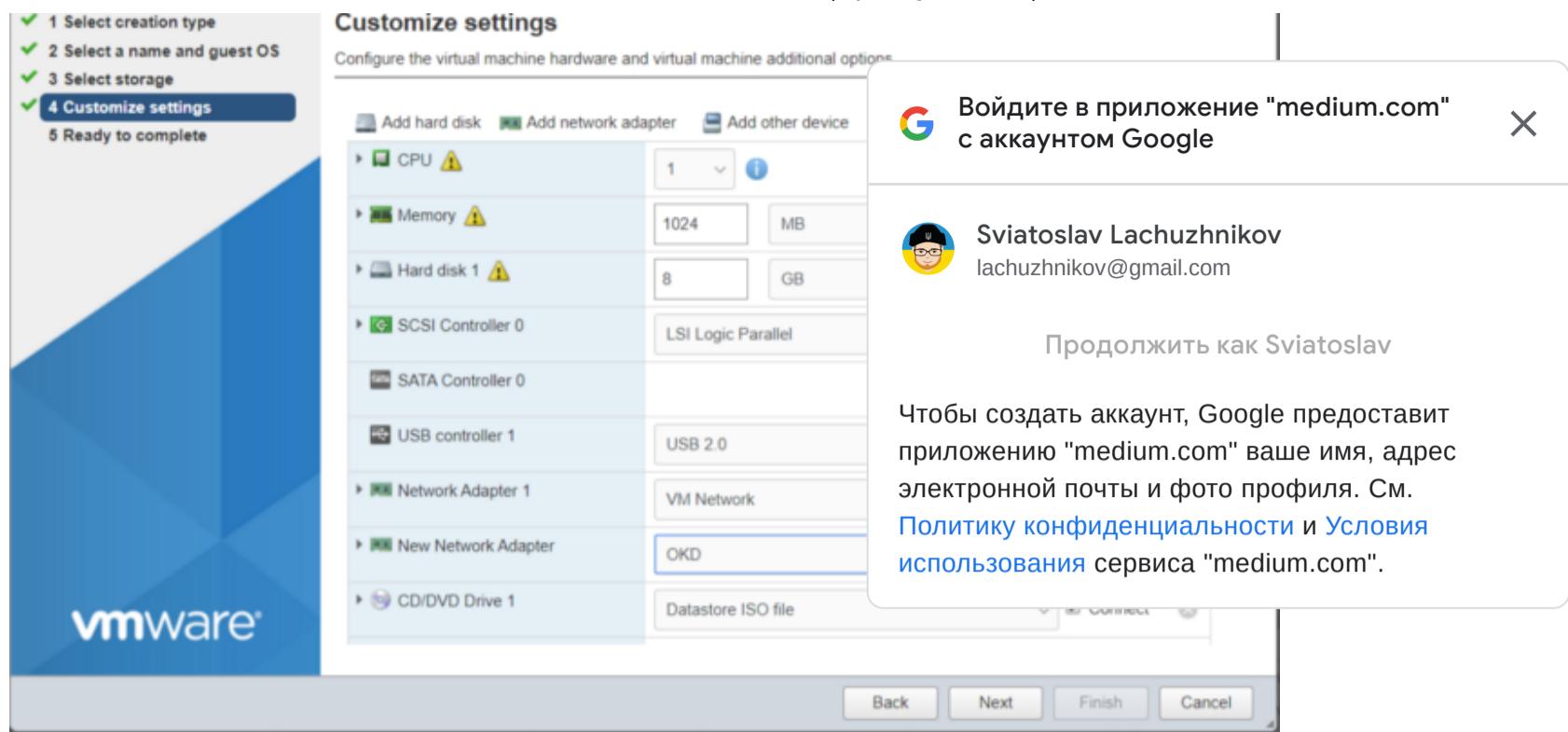
Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. Политику конфиденциальности и Условия использования сервиса "medium.com".

Create a new Virtual Machine. Choose Guest OS as *Other* and Select FreeBSD 64-bit.



Use the default template settings for resources. I prefer to thin provision my disks for non-production use.

Select your home network for Network Adapter 1, and add a new network adapter using the OKD network.



Setup pfSense:

Power on your pfSense VM and run through the installation using all the default values. After completion your VM console should look like this:

```

okd4-pfsense
Starting syslog...done.
Starting CRON... done.
pfSense 2.4.4-RELEASE (Patch 3) amd64 Wed May 15 18:53:44 EDT 2019
Bootup complete

FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

VMware Virtual Machine - Netgate Device ID: 7edea533ffe02c

*** Welcome to pfSense 2.4.4-RELEASE-p3 (amd64) on pfSense ***

WAN (wan)      -> em0          -> v4/DHCP4: 192.168.1.244/24
LAN (lan)      -> em1          -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults 13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                15) Restore recent configuration
7) Ping host                  16) Restart PHP-FPM
8) Shell

Enter an option: 

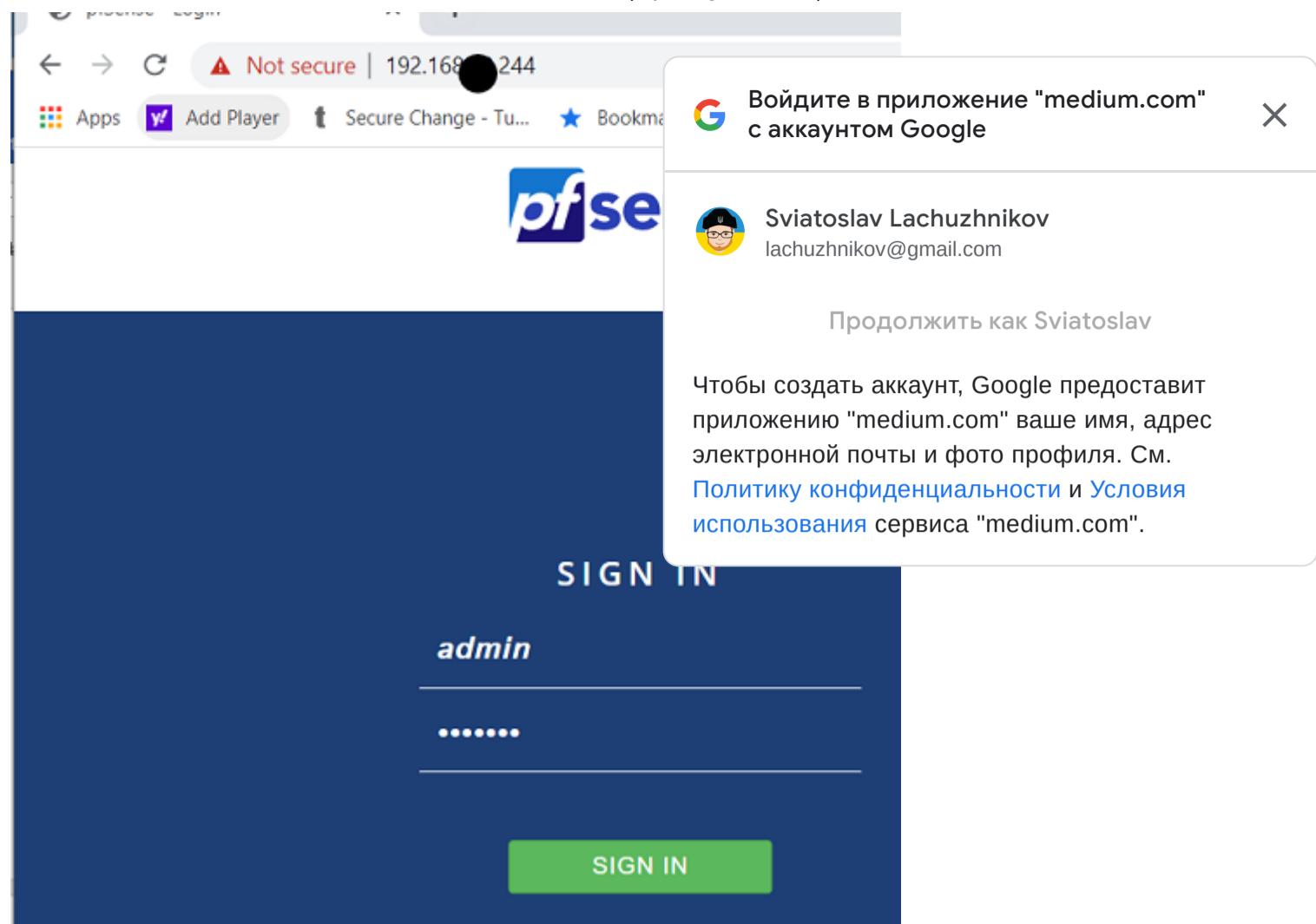
```

pfSense Console

Using the VMWare console on the pfSense VM, select option 8 to access the shell. Add a firewall rule by typing “`easymrule pass wan tcp any any 443`”. Next, type “`pfctl -d`” to turn off the packet filter temporarily. This rule allows you to access the pfSense web GUI from your workstation.

Source: <https://docs.j7k6.org/pfsense-wan-webinterface/>

Login to pfSense via your web-browser. The default username is “admin” and the password is “pfsense”.



After logging in, click next and use “okd4-pfsense” for hostname and “okd.local” for the domain. Also, enable DNS Server Override and Disable DNS Forwarder. Next.

System / General Setup

System

Hostname: okd4-pfSense
Name of the firewall host, without domain part

Domain: okd.local
Do not use '.local' as the final part of the domain (TLD). The '.local' domain is widely used by Bonjour/Rendezvous/Airprint/Airplay), and some Windows systems and networked devices. Alternatives such as '.local.lan' or '.mylocal' are safe.

DNS Server Settings

DNS Servers: DNS Server, DNS Hostname, none

Add DNS Server: + Add DNS Server

DNS Server Override: Allow DNS server list to be overridden by DHCP/PPP on WAN
If this option is set, pfSense will use DNS servers assigned by a DHCP/PPP server on WAN (e.g. Router/ISP). However, they will not be assigned to DHCP clients.

Disable DNS Forwarder: Do not use the DNS Forwarder/DNS Resolver as a DNS server for the firewall
By default localhost (127.0.0.1) will be used as the first DNS server where the DNS Forwarder is disabled so system can use the local DNS service to perform lookups. Checking this box omits localhost from the list of DNS servers.

Select your Timezone. Next.

Use Defaults for WAN Configuration. Uncheck “Block RFC1918 Private Networks” since your home network is the “WAN” in this setup. Next.

RFC1918 Networks

Block RFC1918 Private Networks

- Block private networks from entering via WAN

When set, this option blocks traffic from IP address (10/8, 172.16/12, 192.168/16) as well as loopback on, unless the WAN network lies in such a private a

Block bogon networks

Block bogon networks

- Block non-Internet routed networks from enterin

When set, this option blocks traffic from IP address IANA. Bogons are prefixes that should never appear in the Internet routin as the source address in any packets received.

» Next



Войдите в приложение "medium.com" с аккаунтом Google X



Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. Политику конфиденциальности и Условия использования сервиса "medium.com".

Create bootstrap, master, and worker 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 31.20200505.3.0

Stream: stable [\(JSON\)](#)—7 hours ago

Cloud Launchable

Bare Metal & Virtualized

Bare Metal

Verify your download using the detached signature after importing [Fedora's GPG signing keys](#). The detached signature is for the released artifact itself. If there is a good signature from one of the Fedora keys, and the SHA256 checksum matches, then the download is valid.

ISO

(iso)

31.20200505.3.0 stable

[Download](#)

[Verify signature & SHA256](#)

Virtualized

Verify your download us [Fedora's GPG signing ke](#) artifact itself. If there is a and the SHA256 checks!

OpenStack

(qcow2.xz)

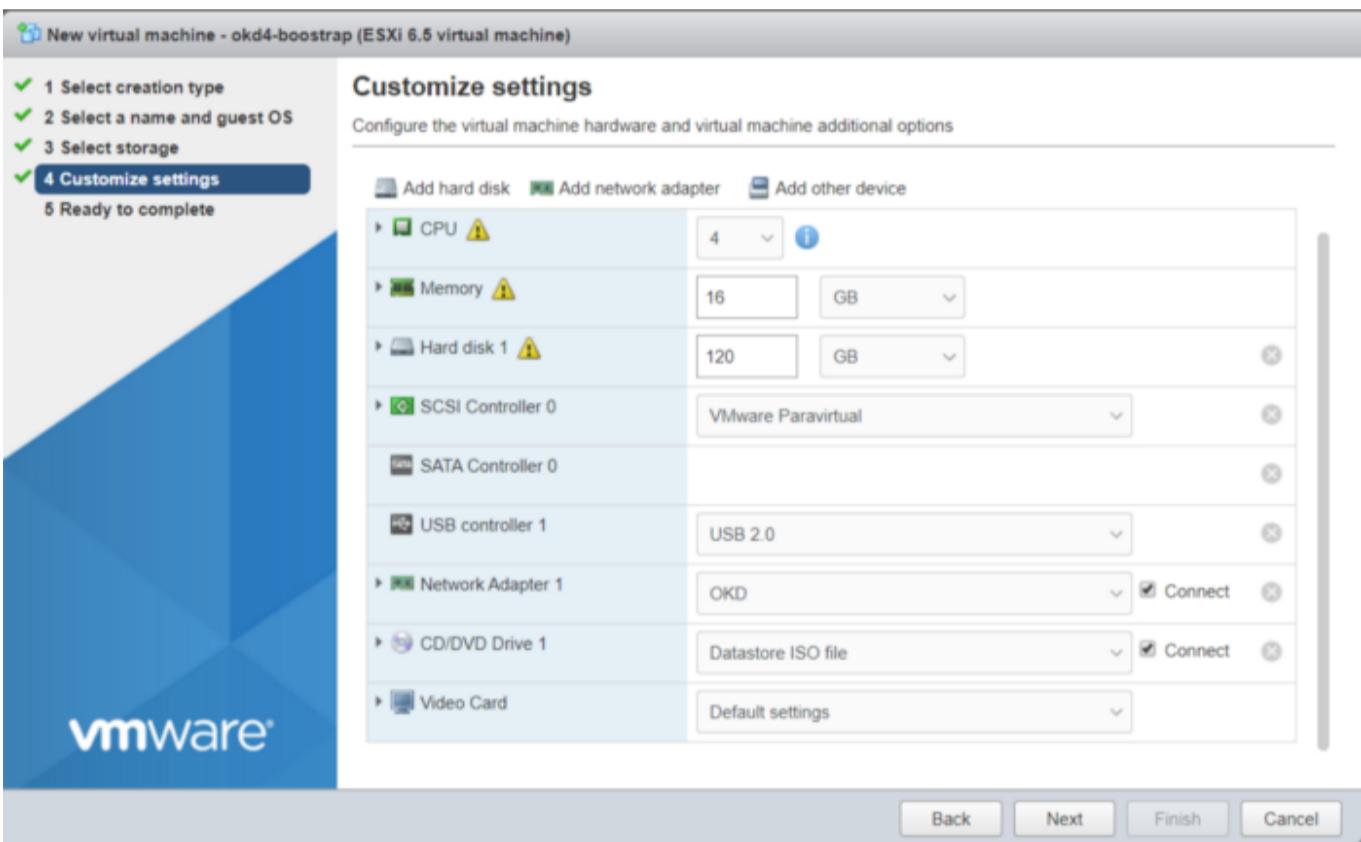
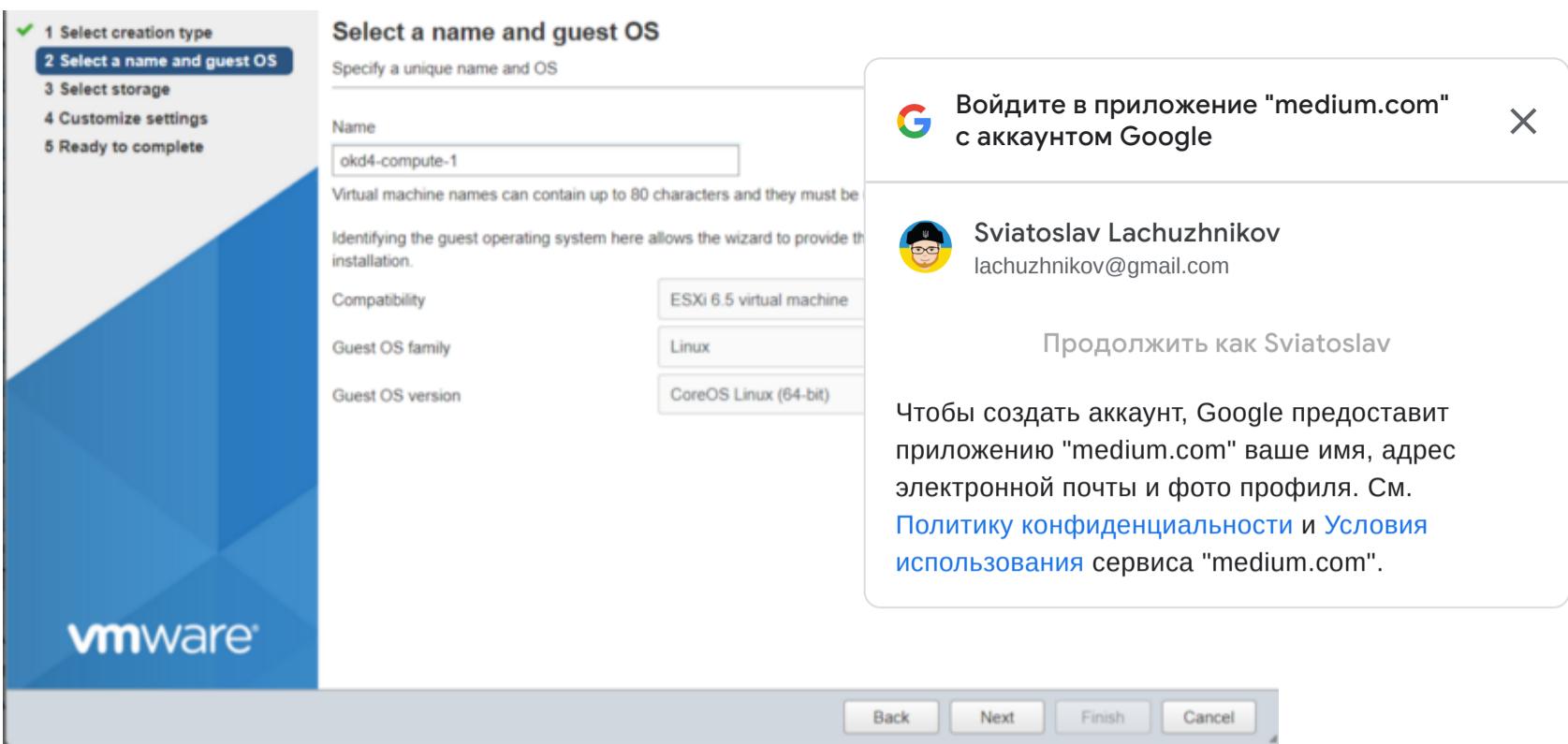
31.20200505.3.0 s

[Download](#)

[Verify signature & S](#)

Create the six ODK nodes (bootstrap, master, worker) on your ESXi host using the values in the spreadsheet:

Note: I had to increase my compute nodes to 4 vCPU to deploy some image-pruner pods. YMMV.



You should end up with the following VMs:

	Virtual machine	Status	Guest OS	Used space	H
	okd4-bootstrap	Normal	CoreOS Linux (64-bit)	Unknown	U
	okd4-compute-1	Normal	CoreOS Linux (64-bit)	Unknown	U
	okd4-compute-2	Normal	CoreOS Linux (64-bit)	Unknown	U
	okd4-control-plane-1	Normal	CoreOS Linux (64-bit)	Unknown	U
	okd4-control-plane-2	Normal	CoreOS Linux (64-bit)	Unknown	U
	okd4-control-plane-3	Normal	CoreOS Linux (64-bit)	Unknown	U
	okd4-pfsense	Normal	FreeBSD (64-bit)	9.11 GB	U

Setup DHCP reservations:

Compile a list of the OKD nodes MAC addresses by viewing the hardware configuration of your VMs.

Hardware Configuration

CPU	4 vCPUs
Memory	16 GB
Hard disk 1	120 GB
USB controller	USB 2.0
Network adapter 1	
Network	OKD (Connected)
Connected	Yes
MAC address	00:0c:29:[REDACTED]
Pass-through (Direct-path I/O)	Yes
Video card	16 MB
CD/DVD drive 1	ISO [REDACTED] D] ISOs/fedora-coreos-31.20200210.3.0-live.x86_64.iso Select disc image
Others	
Additional Hardware	

Войдите в приложение "medium.com" с аккаунтом Google

Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. Политику конфиденциальности и Условия использования сервиса "medium.com".

Login into pfSense. Go to Services → DHCP Server and change your ending range IP to 192.168.1.99, then click Save.

Services / DHCP Server / LAN

LAN

General Options

Enable	<input checked="" type="checkbox"/> Enable DHCP server on LAN interface
BOOTP	<input type="checkbox"/> Ignore BOOTP queries
Deny unknown clients	<input type="checkbox"/> Only the clients defined below will get DHCP leases from this server.
Ignore denied clients	<input type="checkbox"/> Denied clients will be ignored rather than rejected. This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is assigned.
Ignore client identifiers	<input type="checkbox"/> If a client includes a unique identifier in its DHCP request, that UID will not be recorded in its lease. This option may be useful when a client can dual boot using different client identifiers but the same address. Note that the resulting server behavior violates the official DHCP specification.
Subnet	192.168.1.0
Subnet mask	255.255.255.0
Available range	192.168.1.1 - 192.168.1.254
Range	192.168.1.10 - 192.168.1.100

On the DHCP Server, page click Add at the bottom.

DHCP Static Mappings for this Interface

Static ARP	MAC address	IP address	Hostname	Description

+ Add

Fill in the MAC Address, IP Address, and Hostname, then click save. Do this for each ODK VM. Click Apply Changes at the top of the page when

complete.

DHCP Static Mappings for this Interface			
Static ARP	MAC address	IP address	Hostname
00:50:56:	[REDACTED]	192.168.1.200	okd4-bootstrap
00:50:56:	[REDACTED]	192.168.1.201	okd4-control-pla
00:50:56:	[REDACTED]	192.168.1.202	okd4-control-pla
00:50:56:	[REDACTED]	192.168.1.203	okd4-control-pla
00:50:56:	[REDACTED]	192.168.1.204	okd4-compute-1
00:50:56:	[REDACTED]	192.168.1.205	okd4-compute-2

Войдите в приложение "medium.com" с аккаунтом Google X

Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".

+ Add

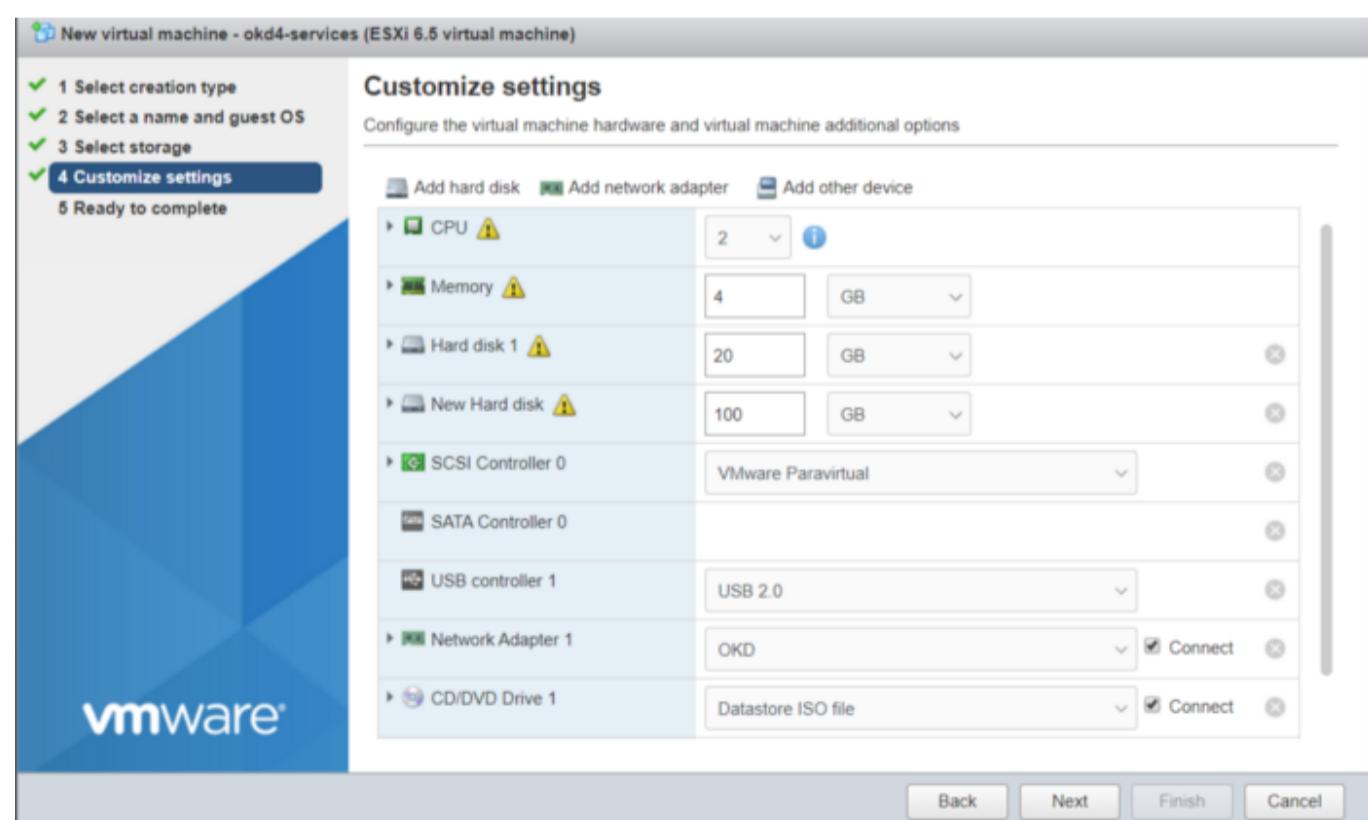
Create and set up VM to host various services:

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

Download CentOS 7 minimal ISO (no GUI) and upload it to your ESXi datastore. Example: CentOS-7-x86_64-Minimal-1908.iso

Note Other users have reported issues with CentOS 8, I recommend using CentOS 7.

Create your okd4-services virtual machine. The 20GB drive is for the OS. The 100GB drive is for NFS and persistent storage for the OpenShift registry.



After creating the VM, copy the MAC address on the VM Hardware configuration page and set up a DHCP Reservation for this VM using the IP

address 192.168.1.210. Hit “Apply Changes” at the top of the DHCP page when completed.

DHCP Static Mappings for this Interface		
Static ARP	MAC address	IP address
	00:50:56:XX:XX:XX	192.168.1.200
	00:50:56:XX:XX:XX	192.168.1.201
	00:50:56:XX:XX:XX	192.168.1.202
	00:50:56:XX:XX:XX	192.168.1.203
	00:50:56:XX:XX:XX	192.168.1.204
	00:50:56:XX:XX:XX	192.168.1.205
	00:50:56:XX:XX:XX	192.168.1.210

Войдите в приложение "medium.com" с аккаунтом Google

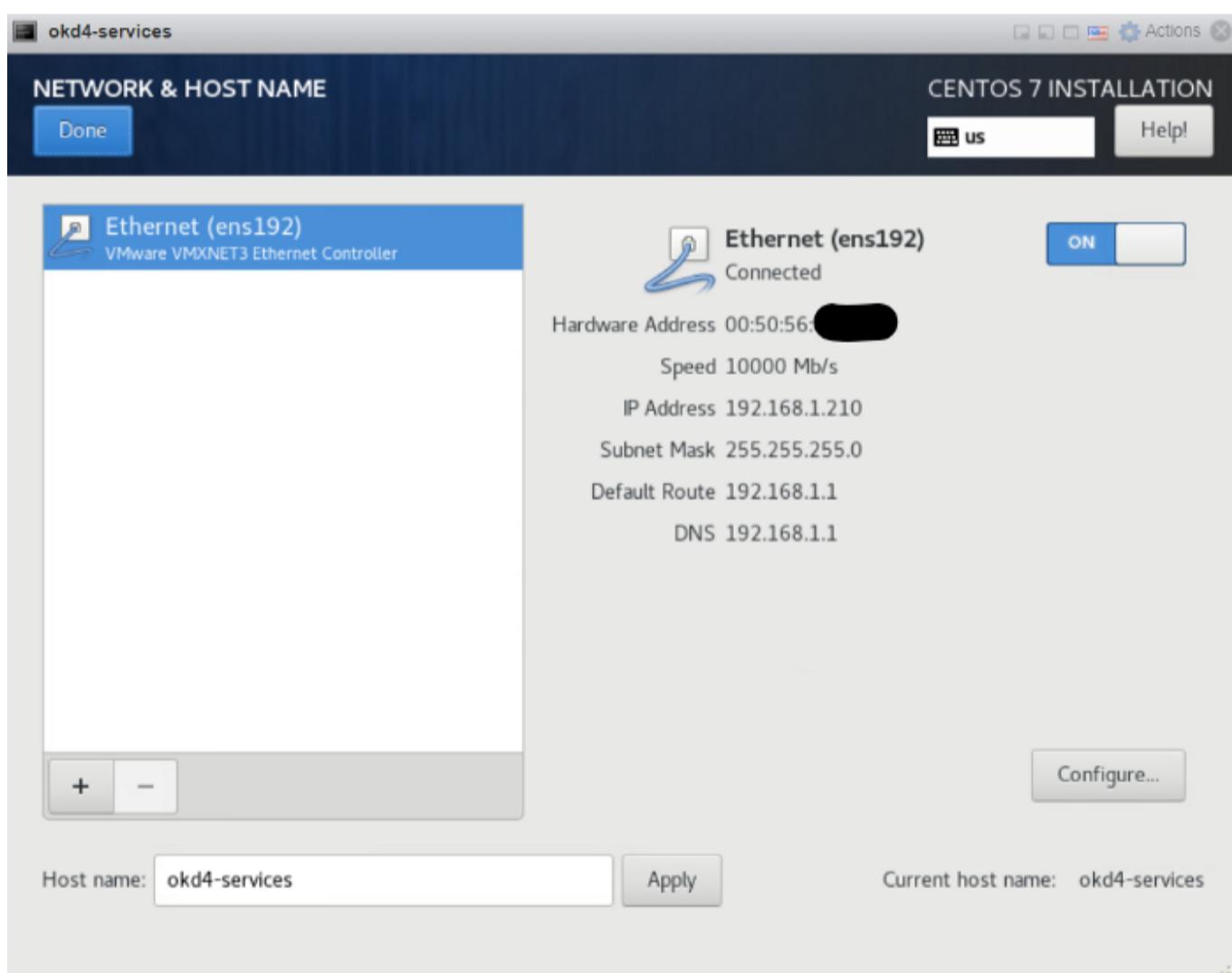
Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. Политику конфиденциальности и Условия использования сервиса "medium.com".

Power on the okd4-services VM.

Run through a typical CentOS 7 install, use the 20GB drive to install the OS. On the networking page, it should pull the 192.168.1.210 IP address from DHCP. Set a root password and an admin user.



While CentOS is installing, go to the pfSense GUI and setup a port forward to the okd4-services for remote ssh/scp access. This rule allows you to ssh from your workstation to copy and paste bash commands easily. Go to Firewall → NAT → Port Forward → Add

Destination port range “22” to “22”; redirect target IP “192.168.1.210”; redirect target port “22”

Click Save, then Apply Changes.

The screenshot shows the 'Edit Redirect Entry' configuration page. Key settings include:

- Disabled:** Disable this rule
- No RDR (NOT):** Disable redirection for traffic matching this rule
- Interface:** WAN
- Protocol:** TCP
- Source:**
- Destination:** Invert match. Type:
- Destination port range:** From port: To port: Custom
- Redirect target IP:**
- Redirect target port:** Port: Custom

Reboot the okd4-services VM after installation and ssh to the WAN IP of your pfSense VM that you port forwarded to the okd4-services server.

```
ssh user@pfSense_ip
```

Install open-vm-tools, vim, wget, and git packages. Also, yum update the OS and reboot after completion.

```
sudo yum install -y open-vm-tools vim wget git && sudo yum update -y
&& sudo init 6
```

```
[redacted]@okd4-services:~$ ssh [redacted]@192.168.1.244
[redacted]@192.168.1.244's password:
Last failed login: Mon Feb 24 22:19:29 EST 2020 from 192.168.1.231 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Mon Feb 24 22:18:54 2020 from 192.168.1.231
[redacted]@okd4-services ~]$ sudo yum install -y open-vm-tools && sudo yum update -y
[redacted]
```

After updated and a restart. Use git to download the DNS, HAProxy, and install-conf.yaml example files via git:

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

...

Install bind (DNS)

```
sudo yum -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
```

 Войдите в приложение "medium.com" с аккаунтом Google X

 Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".

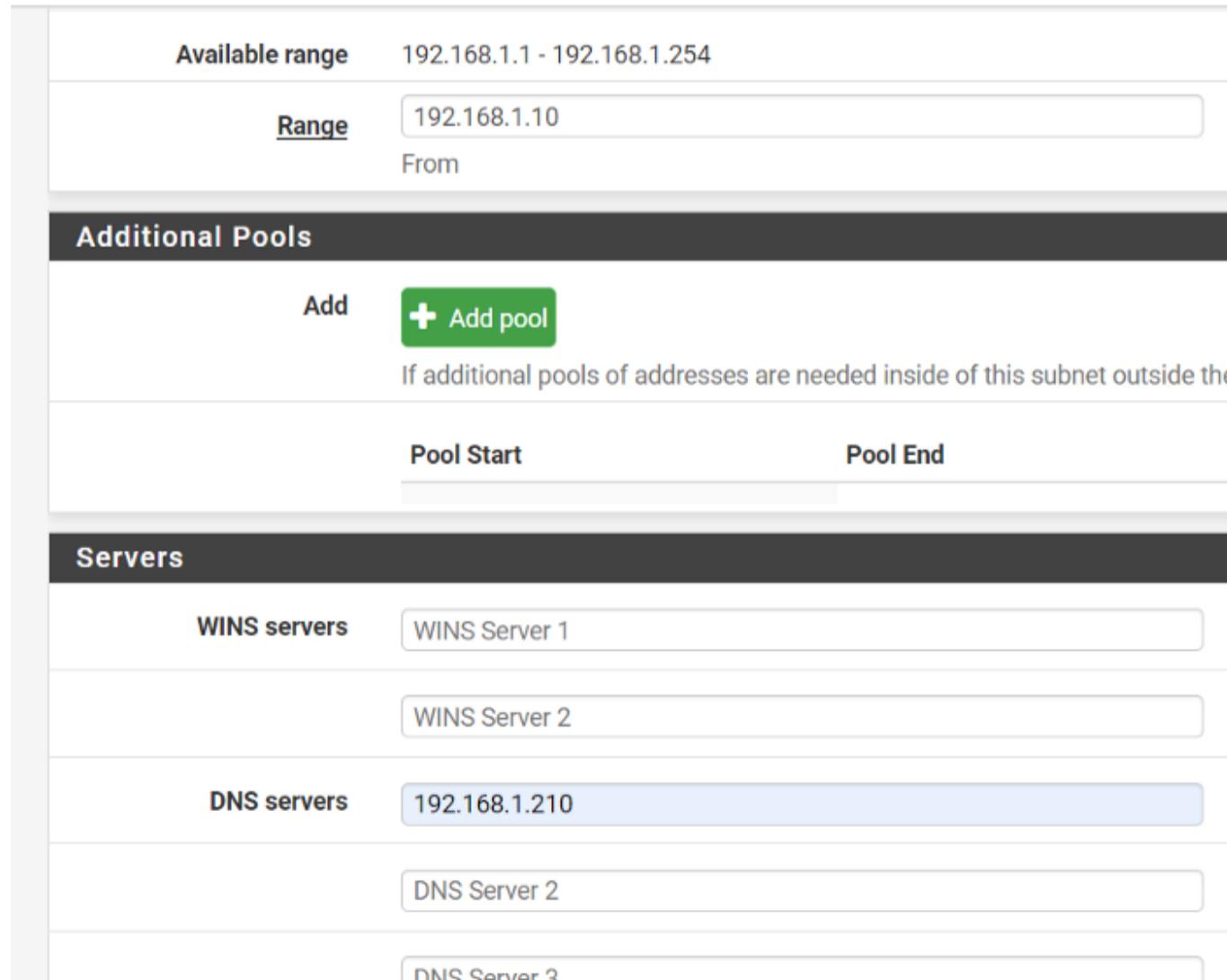
Enable and start named:

```
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
```

Set the DNS server IP to 192.168.1.210 in the pfSense DHCP server settings. Services → DHCP → Servers



Available range	
Range	192.168.1.10
From	

Additional Pools

Add	+ Add pool
If additional pools of addresses are needed inside of this subnet outside the	

Pool Start	Pool End

Servers

WINS servers	WINS Server 1
	WINS Server 2

DNS servers	192.168.1.210
	DNS Server 2
	DNS Server 3

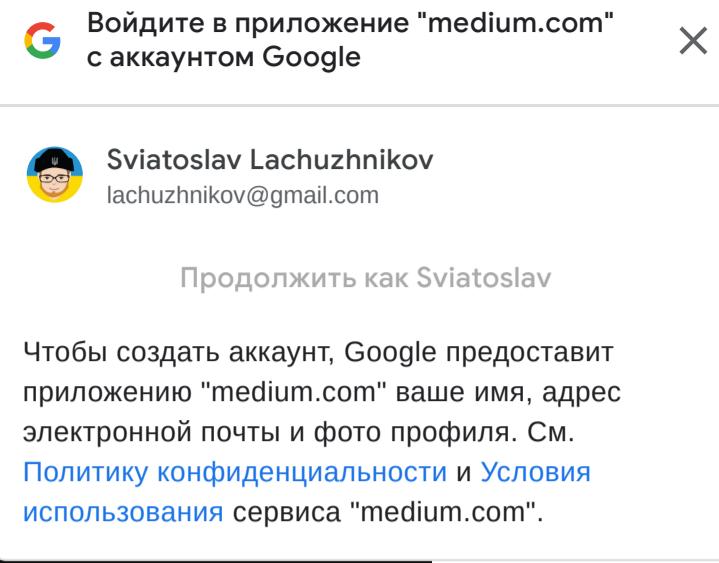
Restart the network services on the okd4-services VM:

```
sudo systemctl restart network.service
```

Test DNS on the okd4-services

```
dig okd.local
dig -x 192.168.1.210
```

```
[root@okd4-services ~]$ dig okd.local
; <>> DiG 9.11.4-P2-RedHat-9.11.4-9.P2.el7 <>> 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: 62467
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;okd.local.           IN      A
;
;; AUTHORITY SECTION:
okd.local.        604800  IN      SOA     okd4-services.okd.local. admin.okd.local. 1 604800 86400 2419200 604800
;
;; Query time: 0 msec
;; SERVER: 192.168.1.210#53(192.168.1.210)
;; WHEN: Fri Feb 28 00:29:16 EST 2020
;; MSG SIZE rcvd: 94
```



Install HAProxy:

```
sudo yum install haproxy
```

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 yum install -y httpd
```

Change httpd to listen port to 8080:

```
sudo vim /etc/httpd/conf/httpd.conf
```

 Войдите в приложение "medium.com" с аккаунтом Google X

 Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".

Change “Listen 80” to “Listen 8080” around line 42:

```
# ServerRoot: The top of the directory tree under which
# configuration, error, and log files are kept
#
# Do not add a slash at the end of the directory path.
# ServerRoot at a non-local disk, be sure to specify a
# full path name. "None" indicates /usr/local/apache2
# Mutex directive, if file-based mutexes are used.
# same ServerRoot for multiple httpd daemons,
# least PidFile.
#
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific
# ports, instead of the default. See also the
# directive.
#
# Change this to Listen on specific IP address as shown
# prevent Apache from glomming onto all bound
# ports.
#
#Listen 12.34.56.78:80
Listen 8080
```

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
```

```
<div class="container">
  <div class="row">
    <div class="col-sm-6">
      <h2>Important note:</h2>
      <p class="lead">The CentOS Project has nothing to do with this website or its content,
      it just provides the software that makes the website run.</p>
      <p>If you have issues with the content of this site, contact the owner of the domain, not the CentOS project.</p>
```

```
Unless you intended to visit CentOS.org, the CentOS Project does not have anything to do with this website,
the content or the lack of it.</p>
<p>For example, if this website is www.example.com the example.com domain at the following WHOIS server:</p>
<p><a href="http://www.internic.net/whois.html">http://www.internic.net/whois.html</a></p>
</div>
<div class="col-sm-6">
<h2>The CentOS Project</h2>
<p>The CentOS Linux distribution is a stable, predictable platform derived from
the sources of Red Hat Enterprise Linux (RHEL).<br/>
Additionally to being a popular choice for web
a rich platform for open source communities to build upon. For
please visit the <a href="http://www.centos.org">CentOS.org</a></p>
</div>
</div>
</div>
</div>
</body></html>
[    @okd4-services ~]$
```

A successful curl should look like this.

Войдите в приложение "medium.com" с аккаунтом Google X

Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".

pfSense Tweaks

To access OKD on port 443 from your home network, we need to change the management port on the pfSense like so:

Add 4433 to Firewall rule on PFSense; Firewall → Rules → WAN → Add

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
×	0 /8 KIB	*	Reserved Not assigned by IANA	*	*	*	*	*	*	Block bogon networks	⚙️
✓	0 /0 B	IPv4 TCP	*	*	*	4433	*	none		change default pfsense port to 4433	🔗 📝 ✖️ trash
✓	1 /2.08 MiB	IPv4 TCP	*	*	443	(HTTPS)	*	none		Easy Rule: Passed from Firewall Log View	🔗 📝 ✖️ trash
✓	1 /1.08 MiB	IPv4 TCP	*	*	192.168.1.210	22 (SSH)	*	none		NAT	🔗 📝 ✖️ trash
✓	0 /0 B	IPv4 TCP	*	*	192.168.1.210	6443	*	none		NAT	🔗 📝 ✖️ trash

Change PFSense default port to 4433; System → Advanced → TCP Port
Change to 4433

Admin Access	Firewall & NAT	Networking	Miscellaneous	System Tunables	Notifications
--------------	----------------	------------	---------------	-----------------	---------------

The screenshot shows the 'webConfigurator' interface with the following configuration:

- Protocol:** HTTP
- SSL Certificate:** webConfigurator default (5e547895d5f0e)
- TCP port:** 4433
Enter a custom port number for the webConfigurator above after save.
- Max Processes:** 2
Enter the number of webConfigurator processes to run. To concurrently.

A Google sign-in overlay is present, asking to log in to "medium.com" with a Google account. It shows the user's profile picture, name "Sviatoslav Lachuzhnikov", email "lachuzhnikov@gmail.com", and a "Продолжить как Sviatoslav" (Continue as Sviatoslav) button. Below it, text explains that Google will provide the app with the user's name, email, and profile photo.

Add 6443, 80, 443 port-forward rules to okd4-service home network.

The screenshot shows the 'Port Forward' tab of the Firewall/NAT configuration. A message at the top states: "The NAT configuration has been changed. The changes must be applied for them to take effect." A green 'Apply Changes' button is visible. The table below lists four port-forward rules:

Index	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
1	WAN	TCP	*	*	WAN address	443 (HTTPS)	192.168.1.210	443 (HTTPS)		
2	WAN	TCP	*	*	WAN address	80 (HTTP)	192.168.1.210	80 (HTTP)		
3	WAN	TCP	*	*	WAN address	6443	192.168.1.210	6443		
4	WAN	TCP	*	*	WAN address	22 (SSH)	192.168.1.210	22 (SSH)		

Buttons at the bottom include: Add, Save, and Separator.

Congratulations, You Are Half Way There!

Congrats! You should now have a separate home lab environment setup and ready for OKD. Now we can start the install.

... . . .

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

```
wget https://github.com/openshift/okd/releases/download/4.4.0-0.okd-2020-05-23-055148-beta5/openshift-client-linux-4.4.0-0.okd-2020-05-23-055148-beta5.tar.gz
```

```
wget https://github.com/openshift/okd/releases/download/4.4.0-0.okd-2020-05-23-055148-beta5/openshift-install-linux-4.4.0-0.okd-2020-05-23-055148-beta5.tar.gz
```

Extract the okd version of the oc client and openshift-install:

```
tar -zxvf openshift-client-linux-4.4.0-0.okd-2021-06-14-19-15-00-beta5.tar.gz
```

```
tar -zxvf openshift-install-linux-4.4.0-0.okd-2021-06-14-19-15-00-beta5.tar.gz
```

Войдите в приложение "medium.com" с аккаунтом Google X



Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".

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 apply a bypass fix. For the bypass use “{“auths”:{“fake”:{“auth”: “bar”}}}" as the pull-secret.

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

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, insert your pull secret and ssh key, and backup the install-config.yaml as it will be deleted in the next step:

```
sudo 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/
```

Modify the cluster-scheduler-02-config.yaml manifest from being scheduled on the control plane machine

```
vim install_dir/manifests/cluster-scheduler-02-config.yaml
```

Locate the mastersSchedulable parameter and change the value to false :

```
apiVersion: config.openshift.io/v1
kind: Scheduler
metadata:
  creationTimestamp: null
  name: cluster
spec:
  mastersSchedulable: False
  policy:
    name: ""
status: {}
```

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
shorten the file names:

```
cd /var/www/html/okd4/
```

```
sudo wget https://builds.coreos.fedoraproject.org/prod/streams/stable/builds/31.20200505.3.0/x86_64/fedora-coreos-31.20200505.3.0-metal.x86_64.raw.xz
```

```
sudo wget https://builds.coreos.fedoraproject.org/prod/streams/stable/builds/31.20200505.3.0/x86_64/fedora-coreos-31.20200505.3.0-metal.x86_64.raw.xz.sig
```

```
sudo mv fedora-coreos-31.20200505.3.0-metal.x86_64.raw.xz fcos.raw.xz
```

```
sudo mv fedora-coreos-31.20200505.3.0-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:

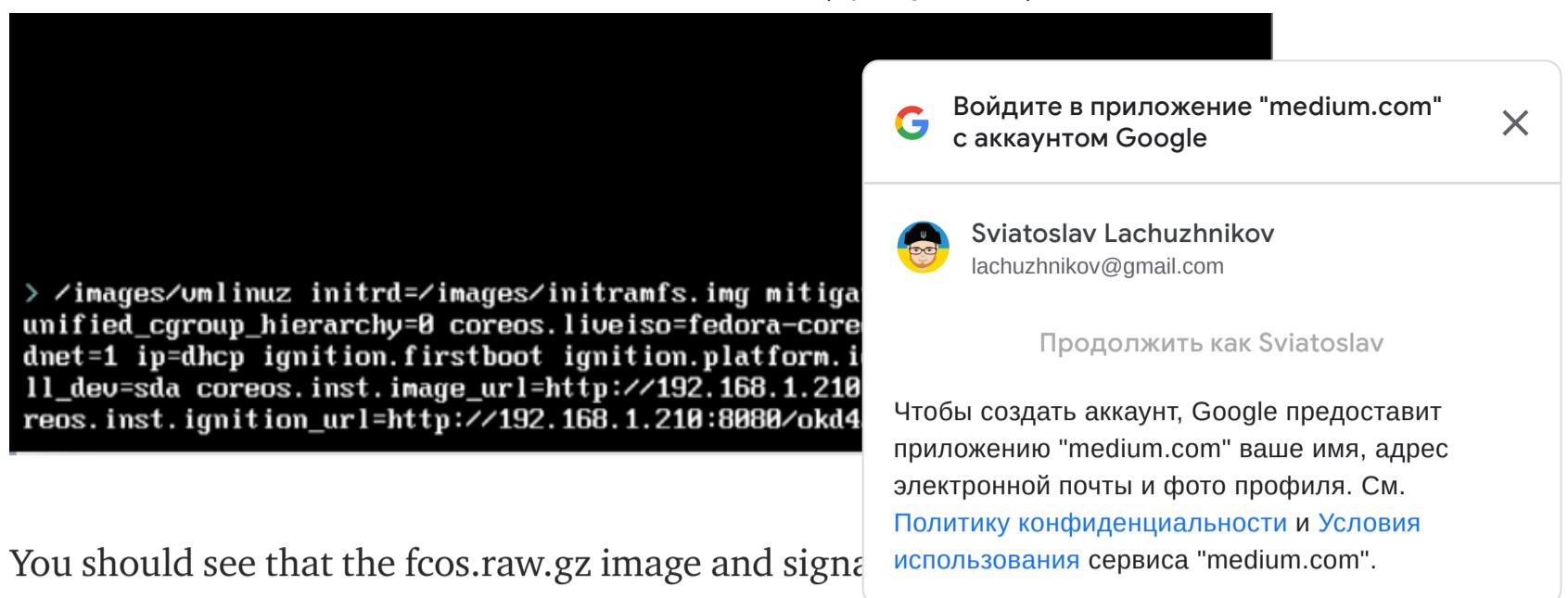
Power on odk4-bootstrap VM. Press the TAB key to edit the kernel boot options and add the following:

```
coreos.inst.install_dev=/dev/sda
```

```
coreos.inst.image_url=http://192.168.1.210:8080/okd4/fcos.raw.xz
```

```
coreos.inst.ignition_url=http://192.168.1.210:8080/okd4/bootstrap.ign
```





```
okd4-bootstrap
[ OK ] Started Update is Completed.
[ OK ] Reached target System Initialization.
[ OK ] Started Daily rotation of log files.
[ OK ] Started Daily Cleanup of Temporary Directories.
[ OK ] Reached target Timers.
[ OK ] Listening on D-Bus System Message Bus Socket.
      Starting Docker Socket for the API...
[ OK ] Listening on Docker Socket for the API.
[ OK ] Reached target Sockets.
[ OK ] Reached target Basic System.
      Starting Network Manager...
      Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
[ OK ] Started Network Manager.
[ OK ] Reached target Network.
      Starting Network Manager Wait Online...
      Starting Hostname Service...
[ OK ] Started Hostname Service.
[ OK ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.
      Starting Network Manager Script Dispatcher Service...
[ OK ] Started Network Manager Script Dispatcher Service.
[ OK ] Started Network Manager Wait Online.
[ OK ] Reached target Network is Online.
      Starting CoreOS Installer...
#####
[ 13.390565] coreos-installer-service[1121]: coreos-installer install /dev/sda --ignition /tmp/coreos-installer-yGio8s --firstboot-args rd.neednet=1 ip=dhcp --image-url http://192.168.1.210:8080/okd4/fcos.raw.xz
[ 13.409749] coreos-installer-service[1121]: Downloading image from http://192.168.1.210:8080/okd4/fcos.raw.xz
[ 13.411227] coreos-installer-service[1121]: Downloading signature from http://192.168.1.210:8080/okd4/fcos.raw.xz.sig
[ 24.030849] kauditd_printk_skb: 7 callbacks suppressed
[ 24.030851] audit: type=1131 audit(1582995305.750:89): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=NetworkManager-dispatcher comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
```

After installation of the image, the VM restarts to the login screen and begins to show the following (do not panic, this is normal):

```
Fedora CoreOS 31.20200210.3.0
Kernel 5.4.17-200.fc31.x86_64 on an x86_64 (tty1)

SSH host key: SHA256:0b1jITsrXPL5HCuaNxOYLynBPhjkhK3zUQhuCJMBnw (ED25519)
SSH host key: SHA256:diP4EHM2FulgullJH82drRSIjbGLA6kzu9tfDoneuI (ECDSA)
SSH host key: SHA256:uBSmaYtYYApGxtrzqtEQhmMbqcg0N7tU+7SdUSInrY8 (RSA)
eth0: 192.168.1.200 fe80::250:56ff:fe86:dae3
okd4-bootstrap login: [ 28.497805] kauditd_printk_skb: 41 callbacks suppressed
[ 28.497806] audit: type=1130 audit(1582995392.892:135): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=kubelet comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
[ 28.498596] audit: type=1131 audit(1582995392.892:136): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=kubelet comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
[ 28.510041] audit: type=1130 audit(1582995392.904:137): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=kubelet comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=failed'
[ 30.310913] audit: type=1131 audit(1582995394.678:138): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=NetworkManager-dispatcher comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
```

The console window shows many messages during the install and restarts several times. It's time to start the control plane and compute nodes.

Starting the control plane nodes:

When powering on the control nodes press TAB to enter the following values (note the master.ign at the end):

```
coreos.inst.install_dev=/dev/sda
```

```
coreos.inst.image_url=http://192.168.1.210:8080/okd4/fcos.raw.xz
```

```
coreos.inst.ignition_url=http://192.168.1.210:8080/okd4/master.ign
```

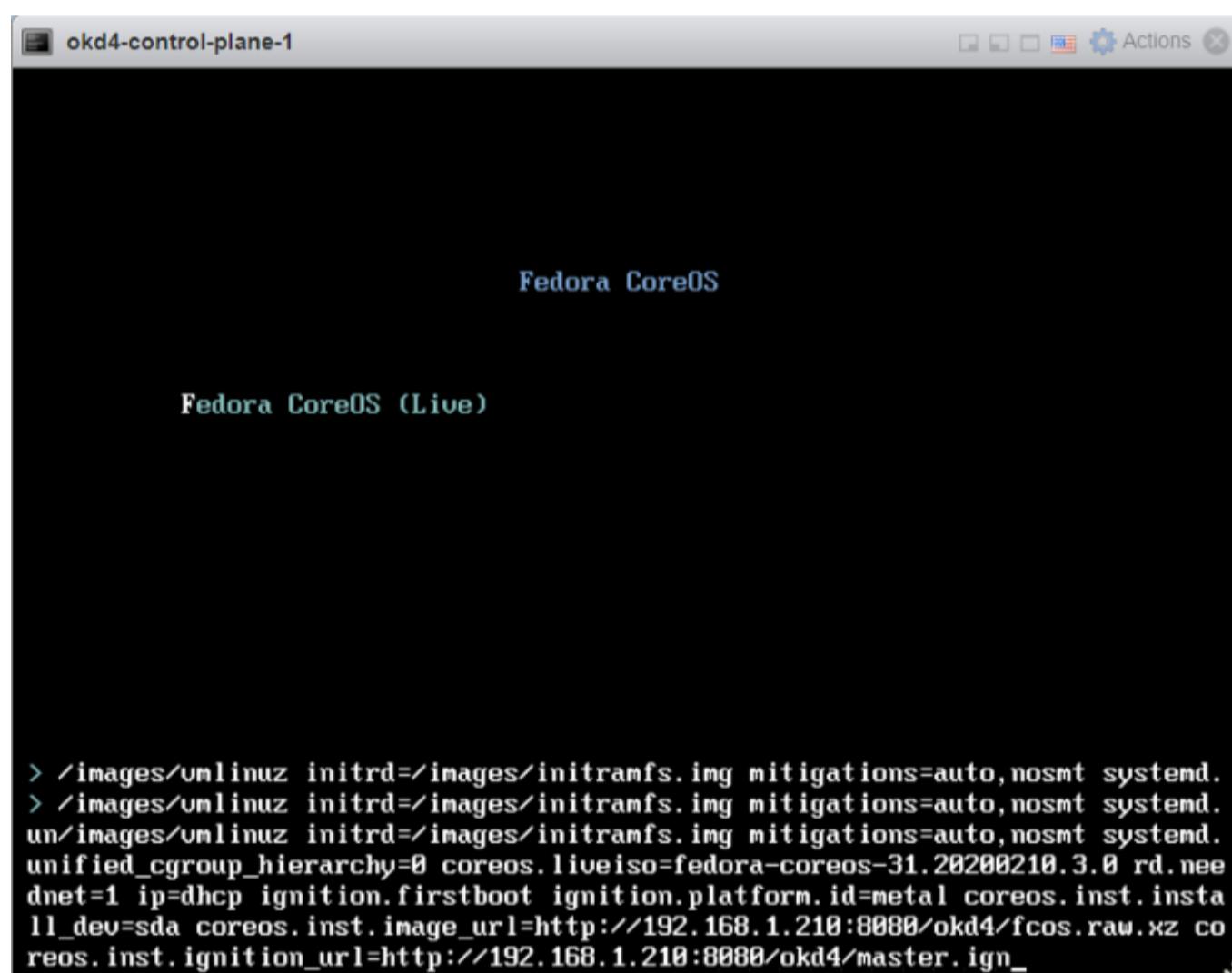
 Войдите в приложение "medium.com" с аккаунтом Google X



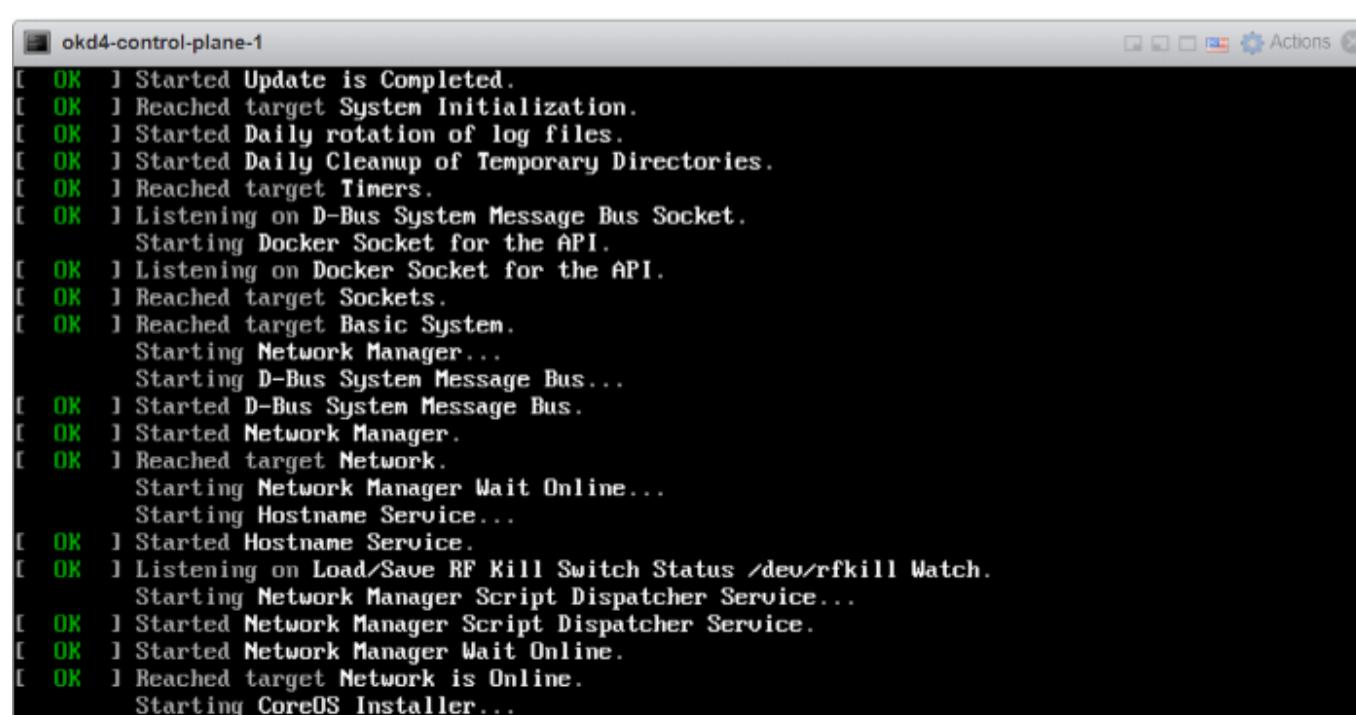
Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

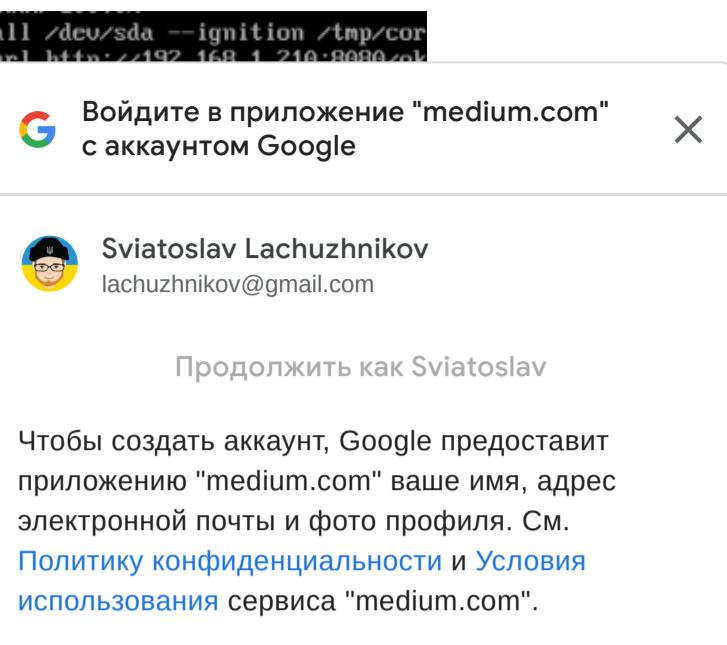
Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".



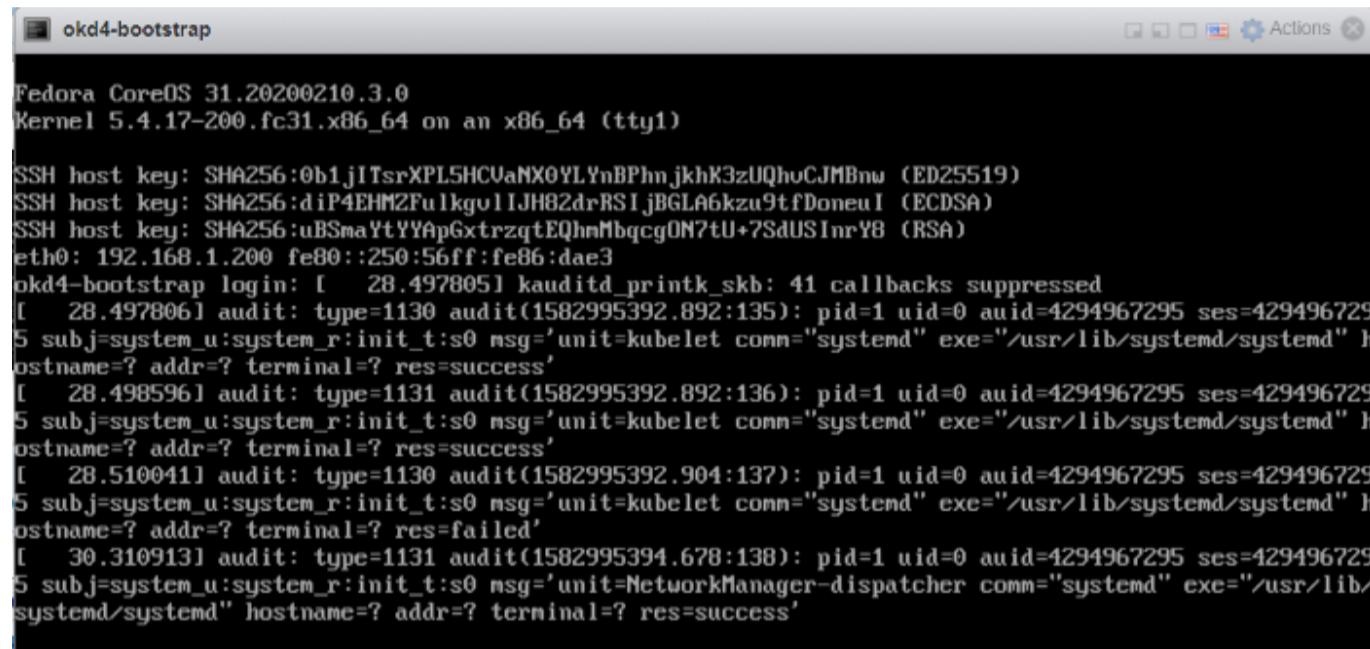
Just like the bootstrap node, you should see that the fcos.raw.gz image and signature have been downloaded:



```
[ 14.261958] coreos-installer-service[1134]: coreos-installer install /dev/sda --ignition /tmp/coreos-installer-fc8KQs --firstboot-args rd.neednet=1 ip=dhcp --image-url http://192.168.1.210:8080/okd4/fcos.raw.xz
[ 14.285485] coreos-installer-service[1134]: Downloading image from http://192.168.1.210:8080/okd4/fcos.raw.xz
[ 14.287340] coreos-installer-service[1134]: Downloading signature for http://192.168.1.210:8080/okd4/fcos.raw.xz.sig
[ 25.029501] kauditd_printk_skb: 7 callbacks suppressed
[ 25.029502] audit: type=1131 audit(1582997797.886:89): pid=1 uid=0 auid=1000 subj=system_u:system_r:init_t:s0 msg='unit=NetworkManager-dispatcher comm="systemd" hostname=? addr=? terminal=? res=success'
-
```



After installing the image is installed, the VM restarts to begin to update ostree, and then reboot once completed to become a bootstrap node:



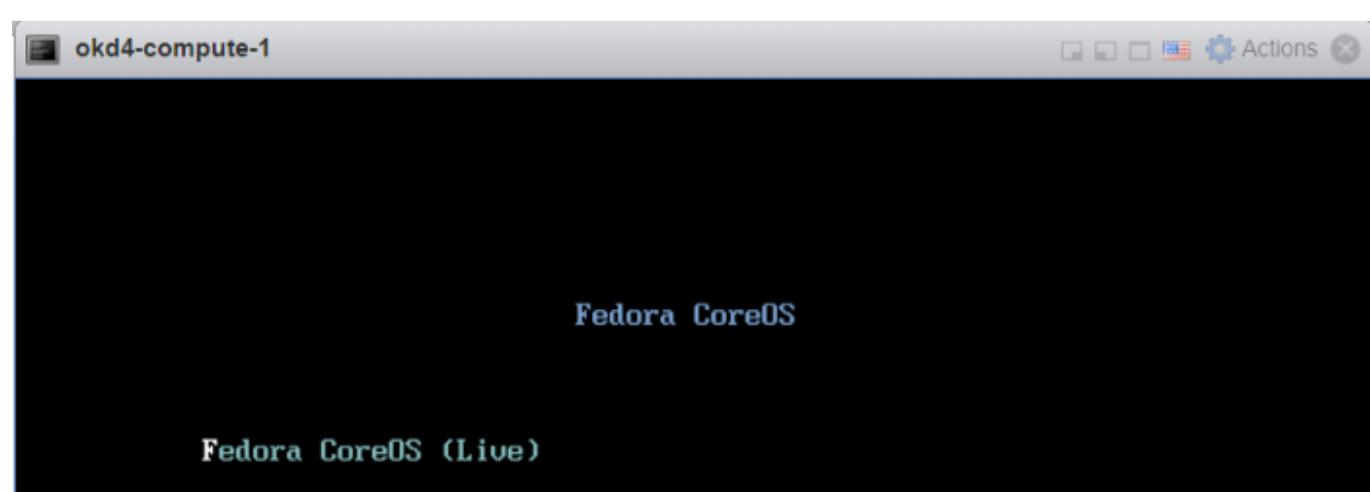
Now repeat the same process for the other two control nodes and compute nodes.

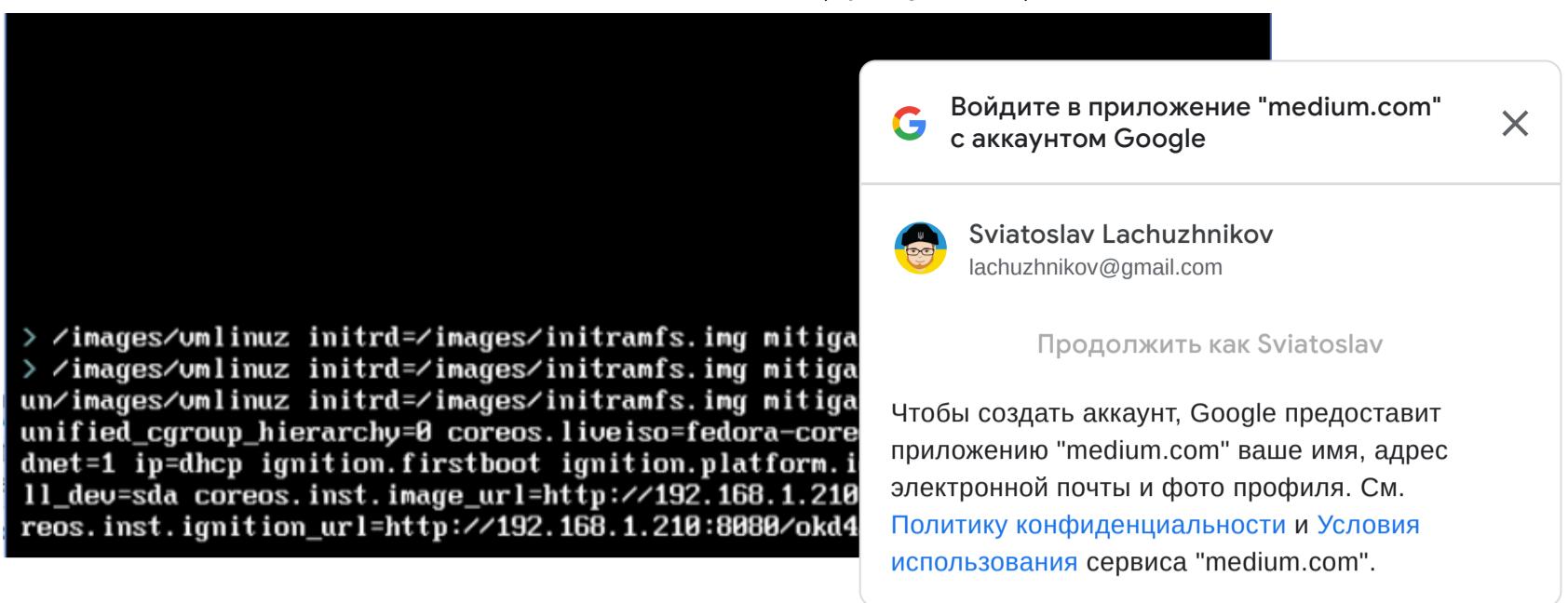
... . . .

Starting the compute nodes:

When powering on the compute nodes press TAB to edit like before and use the following values (note the worker.ign at the end):

```
coreos.inst.install_dev=/dev/sda
coreos.inst.image_url=http://192.168.1.210:8080/okd4/fcos.raw.xz
coreos.inst.ignition_url=http://192.168.1.210:8080/okd4/worker.ign
```





Just like the other nodes, you should see that the fcos.raw.gz image and signature have been downloaded:

```
[ ok ] Started Update is Completed.
[ ok ] Reached target System Initialization.
[ ok ] Started Daily rotation of log files.
[ ok ] Started Daily Cleanup of Temporary Directories.
[ ok ] Reached target Timers.
[ ok ] Listening on D-Bus System Message Bus Socket.
      Starting Docker Socket for the API...
[ ok ] Listening on Docker Socket for the API.
[ ok ] Reached target Sockets.
[ ok ] Reached target Basic System.
      Starting Network Manager...
      Starting D-Bus System Message Bus...
[ ok ] Started D-Bus System Message Bus.
[ ok ] Started Network Manager.
[ ok ] Reached target Network.
      Starting Network Manager Wait Online...
      Starting Hostname Service...
[ ok ] Started Hostname Service.
[ ok ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.
      Starting Network Manager Script Dispatcher Service...
[ ok ] Started Network Manager Script Dispatcher Service.
[ ok ] Started Network Manager Wait Online.
[ ok ] Reached target Network is Online.
      Starting CoreOS Installer...
=====
[ 14.261958] coreos-installer-service[1134]: coreos-installer install /dev/sda --ignition /tmp/cor
eos-installer-fc8KQs --firstboot-args rd.neednet=1 ip=dhcp --image-url http://192.168.1.210:8080/ok
d4/fcos.raw.xz
[ 14.285485] coreos-installer-service[1134]: Downloading image from http://192.168.1.210:8080/okd4
/fcos.raw.xz
[ 14.287340] coreos-installer-service[1134]: Downloading signature from http://192.168.1.210:8080/
okd4/fcos.raw.xz.sig
[ 25.029501] kauditd_printk_skb: 7 callbacks suppressed
[ 25.029502] audit: type=1131 audit(1582997797.886:89): pid=1 uid=0 auid=4294967295 ses=4294967295
  subj=system_u:system_r:init_t:s0 msg='unit=NetworkManager-dispatcher comm="systemd" exe="/usr/lib/s
ystemd/systemd" hostname=? addr=? terminal=? res=success'
```

After installing the image, the VM restarts to the login screen, begin to update ostree, and then reboot once completed just like the other nodes:

```
Fedora CoreOS 31.20200210.3.0
Kernel 5.4.17-200.fc31.x86_64 on an x86_64 (tty1)

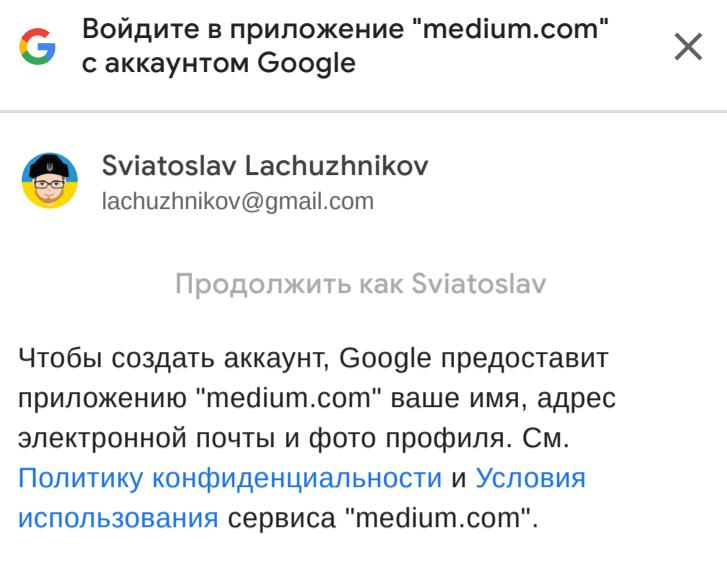
SSH host key: SHA256:0b1jITsrXPL5HCUaNX0YLynBPhnjkhK3zUQhuCJMBnw (ED25519)
SSH host key: SHA256:dIP4EHM2FulgulIJH82drRSIjbGLA6kzu9tfDoneuI (ECDSA)
SSH host key: SHA256:uBSmaYtYYApGxtrzqtEQhmMbqcg0N7tU+7SdUSInrY8 (RSA)
eth0: 192.168.1.200 fe80::250:56ff:fe86:dae3
okd4-bootstrap login: [ 28.497805] kauditd_printk_skb: 41 callbacks suppressed
[ 28.497806] audit: type=1130 audit(1582995392.892:135): pid=1 uid=0 auid=4294967295 ses=429496729
5 subj=system_u:system_r:init_t:s0 msg='unit=kubelet comm="systemd" exe="/usr/lib/systemd/systemd" h
ostname=? addr=? terminal=? res=success'
[ 28.498596] audit: type=1130 audit(1582995392.892:136): pid=1 uid=0 auid=4294967295 ses=429496729
5 subj=system_u:system_r:init_t:s0 msg='unit=kubelet comm="systemd" exe="/usr/lib/systemd/systemd" h
ostname=? addr=? terminal=? res=success'
[ 28.510041] audit: type=1130 audit(1582995392.904:137): pid=1 uid=0 auid=4294967295 ses=429496729
5 subj=system_u:system_r:init_t:s0 msg='unit=kubelet comm="systemd" exe="/usr/lib/systemd/systemd" h
ostname=? addr=? terminal=? res=failed'
[ 30.310913] audit: type=1131 audit(1582995394.678:138): pid=1 uid=0 auid=4294967295 ses=429496729
5 subj=system_u:system_r:init_t:s0 msg='unit=NetworkManager-dispatcher comm="systemd" exe="/usr/lib/
systemd/systemd" hostname=? addr=? terminal=? res=success'
```

Now repeat the same process for the other worker node.

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
```



```
[ @okd4-services ~]$ openshift-install --dir=install_dir/ wait-for bootstrap-complete --log-level=info
INFO Waiting up to 20m0s for the Kubernetes API at https://api.lab.okd.local:6443...
INFO API v1.17.1 up
INFO Waiting up to 40m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
[ @okd4-services ~]$ .
```

Note: At this point, in the image above, you can shutdown your bootstrap node. Now is a good time to edit the /etc/haproxy/haproxy.cfg, comment out the bootstrap node, and restart the haproxy service.

```
sudo vi /etc/haproxy/haproxy.cfg
sudo systemctl restart haproxy
```

...

Login to the cluster and approve CSRs:

Now that the masters are 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:

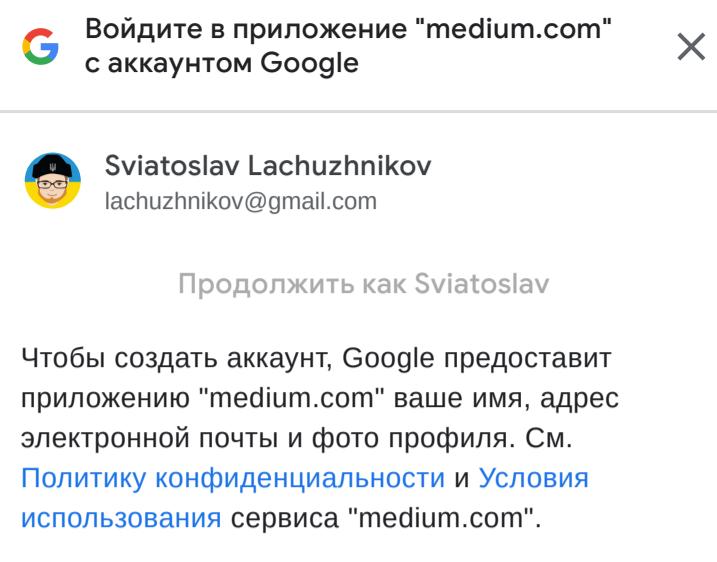
```
export KUBECONFIG=~/install_dir/auth/kubeconfig
oc whoami
oc get nodes
oc get csr
```

```
[ @okd4-services ~]$ export KUBECONFIG=~/install_dir/auth/kubeconfig
[ @okd4-services ~]$ oc whoami
system:admin
[ @okd4-services ~]$ oc get nodes
NAME           STATUS   ROLES    AGE    VERSION
okd4-control-plane-1 Ready    master   20m   v1.17.1
okd4-control-plane-2 Ready    master   19m   v1.17.1
okd4-control-plane-3 Ready    master   20m   v1.17.1
[ @okd4-services ~]$ oc get csr
NAME      AGE     REQUESTOR                                     CONDITION
csr-2b4dt 20m    system:serviceaccount:openshift-machine-config-operator:node-bootstrapper Approved,Issued
csr-2gh6t  8s     system:serviceaccount:openshift-machine-config-operator:node-bootstrapper Pending
csr-799qm  20m    system:node:okd4-control-plane-3          Approved,Issued
csr-92ddt  20m    system:node:okd4-control-plane-1          Approved,Issued
csr-mpnmm  20m    system:serviceaccount:openshift-machine-config-operator:node-bootstrapper Approved,Issued
csr-shdcg  19m    system:node:okd4-control-plane-2          Approved,Issued
csr-tsw7z  6s     system:serviceaccount:openshift-machine-config-operator:node-bootstrapper Pending
csr-xns4d  20m    system:serviceaccount:openshift-machine-config-operator:node-bootstrapper Approved,Issued
[ @okd4-services ~]$ .
```

You should only see the master nodes and several CSR's waiting for approval. Install the jq package to assist with approving them once time.

```
wget -O jq https://github.com/stedolan/jq/releases/download/v1.6/jq-linux64
```

```
chmod +x jq
sudo mv jq /usr/local/bin/
jq --version
```



```
[root@okd4-services ~]$ wget -O jq https://github.com/stedolan/jq/releases/download/v1.6/jq-linux64
--2020-02-29 19:38:07-- https://github.com/stedolan/jq/releases/download/v1.6/jq-linux64
Resolving github.com (github.com)... 192.30.253.112
Connecting to github.com (github.com)|192.30.253.112|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/2020/03/12/193807/Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200301T00362Z&actor_id=0&response-content-disposition=attachment%3B%20filename%3Djq-1.6
--2020-02-29 19:38:07-- https://github-production-release-asset-2e65be.s3.amazonaws.com/2020/03/12/193807/Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200301T00362Z&actor_id=0&response-content-disposition=attachment%3B%20filename%3Djq-1.6
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 192.30.253.112
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|192.30.253.112|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3953824 (3.8M) [application/octet-stream]
Saving to: 'jq'

100%[=====] 3953824 / 3953824 [2020-02-29 19:38:10] (1.55 MB/s) - 'jq' saved [3953824/3953824]

[root@okd4-services ~]$ chmod +x jq
[root@okd4-services ~]$ sudo mv jq /usr/local/bin/
[root@okd4-services ~]$ jq --version
jq-1.6
[root@okd4-services ~]$
```

Approve all the pending certs and check your nodes:

```
oc get csr -ojson | jq -r '.items[] | select(.status == {} ) | .metadata.name' | xargs oc adm certificate approve
```

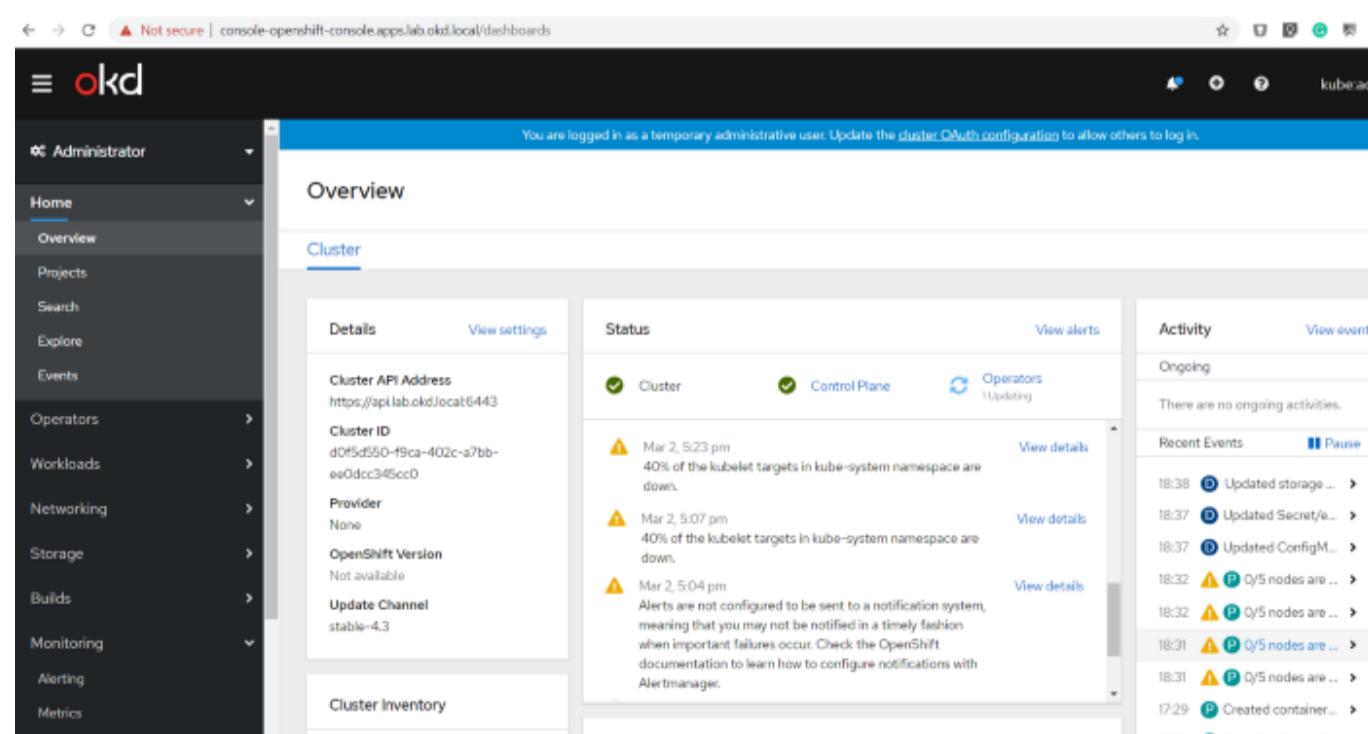
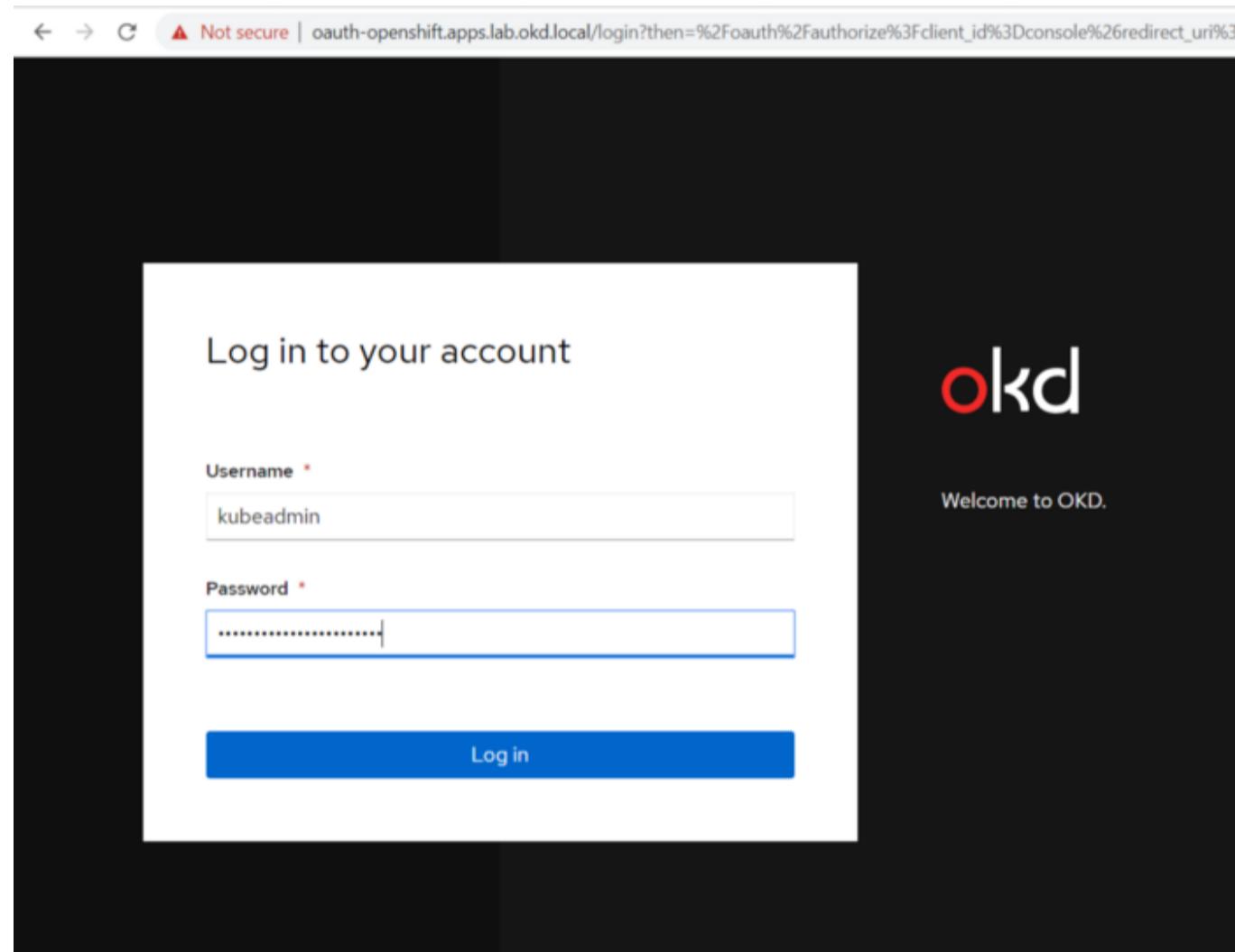
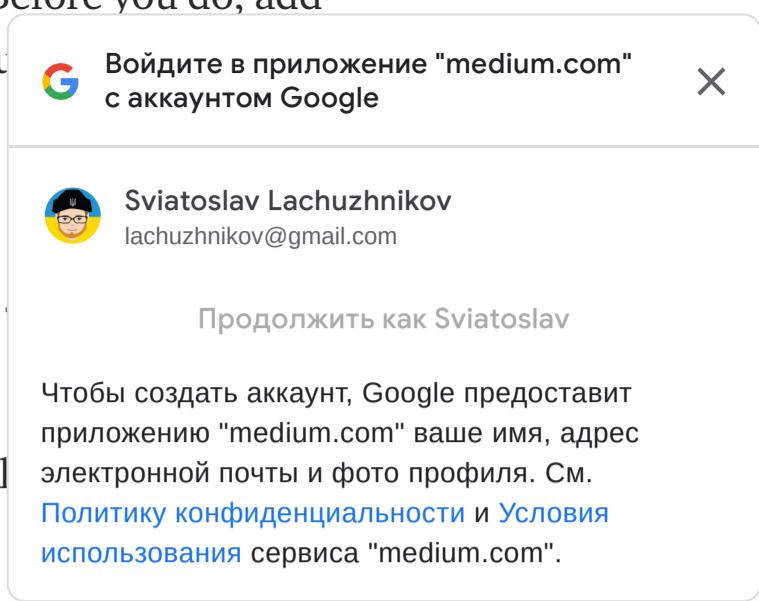
```
[crobinson@okd4-services ~]$ oc get nodes
NAME           STATUS    ROLES      AGE     VERSION
okd4-compute-1 NotReady  worker    77s    v1.17.1
okd4-compute-2 NotReady  worker    81s    v1.17.1
okd4-control-plane-1 Ready     master   24m    v1.17.1
okd4-control-plane-2 Ready     master   24m    v1.17.1
okd4-control-plane-3 Ready     master   25m    v1.17.1
[crobinson@okd4-services ~]$ oc get nodes
NAME           STATUS    ROLES      AGE     VERSION
okd4-compute-1 Ready     worker   2m17s   v1.17.1
okd4-compute-2 Ready     worker   2m21s   v1.17.1
okd4-control-plane-1 Ready     master   25m    v1.17.1
okd4-control-plane-2 Ready     master   25m    v1.17.1
okd4-control-plane-3 Ready     master   26m    v1.17.1
[crobinson@okd4-services ~]$
```

Log in to the web console using your workstation. Before you do, add `console-openshift-console.apps.lab.okd.local` to your hosts file :

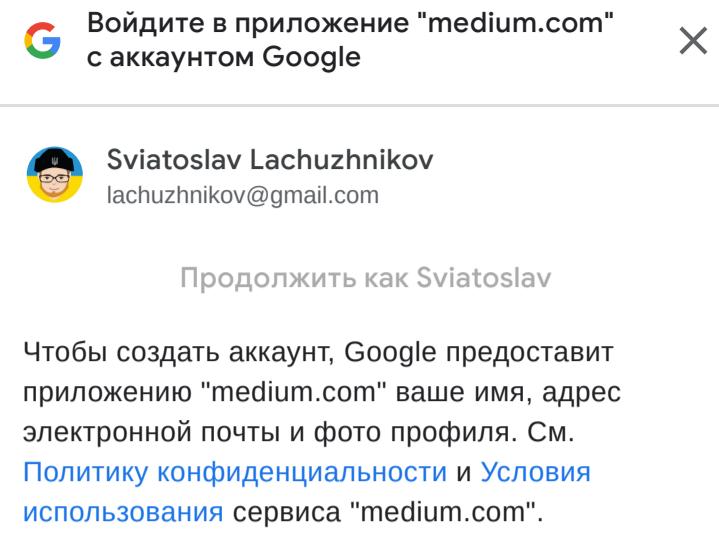
```
<workstation ip addr> console-openshift-console.apps.lab.okd.local
```

Also, get your kubeadmin password from the install directory and login to the web console:

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



The only items remaining are the operators, persistent storage, and configuration of the registry.



Войдите в приложение "medium.com" с аккаунтом Google X

 Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".

Checking Cluster Operators:

Even though the GUI is up, some operators may still be coming online. To see a list of those operators and their status, run the following command:

```
oc get clusteroperators
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.4.0-0.okd-2020-03-02-123048	True	False	False	82m
cloud-credential	4.4.0-0.okd-2020-03-02-123048	True	False	False	124m
cluster-autoscaler	4.4.0-0.okd-2020-03-02-123048	True	False	False	96m
console	4.4.0-0.okd-2020-03-02-123048	True	False	False	85m
csi-snapshot-controller	4.4.0-0.okd-2020-03-02-123048	True	False	False	84m
dns	4.4.0-0.okd-2020-03-02-123048	True	False	False	110m
etcd	4.4.0-0.okd-2020-03-02-123048	True	False	False	107m
image-registry	4.4.0-0.okd-2020-03-02-123048	False	True	False	67m
ingress	4.4.0-0.okd-2020-03-02-123048	True	False	False	90m
insights	4.4.0-0.okd-2020-03-02-123048	True	False	False	112m
kube-apiserver	4.4.0-0.okd-2020-03-02-123048	True	False	False	106m
kube-controller-manager	4.4.0-0.okd-2020-03-02-123048	True	False	False	106m
kube-scheduler	4.4.0-0.okd-2020-03-02-123048	True	False	False	106m
kube-storage-version-migrator	4.4.0-0.okd-2020-03-02-123048	False	False	False	86m
machine-api	4.4.0-0.okd-2020-03-02-123048	True	False	False	109m
machine-config	4.4.0-0.okd-2020-03-02-123048	True	False	False	87m
marketplace	4.4.0-0.okd-2020-03-02-123048	True	False	False	98m
monitoring	4.4.0-0.okd-2020-03-02-123048	True	False	False	84m
network	4.4.0-0.okd-2020-03-02-123048	True	False	False	114m
node-tuning	4.4.0-0.okd-2020-03-02-123048	True	False	False	112m
openshift-apiserver	4.4.0-0.okd-2020-03-02-123048	True	False	False	86m
openshift-controller-manager	4.4.0-0.okd-2020-03-02-123048	True	False	False	97m
openshift-samples	4.4.0-0.okd-2020-03-02-123048	True	False	False	95m
operator-lifecycle-manager	4.4.0-0.okd-2020-03-02-123048	True	False	False	109m
operator-lifecycle-manager-catalog	4.4.0-0.okd-2020-03-02-123048	True	False	False	109m
operator-lifecycle-manager-packageserver	4.4.0-0.okd-2020-03-02-123048	True	False	False	101m
service-ca	4.4.0-0.okd-2020-03-02-123048	True	False	False	112m
service-catalog-apiserver	4.4.0-0.okd-2020-03-02-123048	True	False	False	113m
service-catalog-controller-manager	4.4.0-0.okd-2020-03-02-123048	True	False	False	112m
storage	4.4.0-0.okd-2020-03-02-123048	True	False	False	98m
support	4.4.0-0.okd-2020-03-02-123048	True	False	False	112m

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.

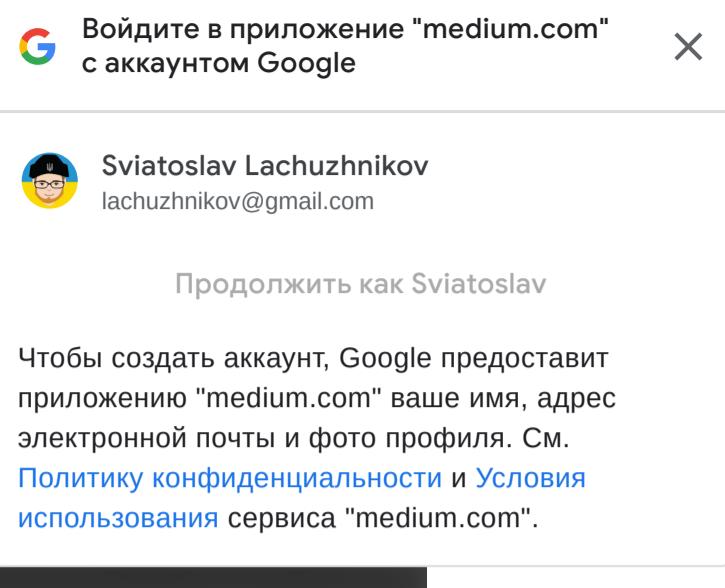
Login to your okd4-services VM and begin to set up an NFS server. The following commands install the necessary packages, enable services, and configure file and folder permissions.

```
sudo yum install -y nfs-utils
sudo systemctl enable nfs-server rpcbind
sudo systemctl start nfs-server rpcbind nfs-lock-nfs-idmap
sudo mkdir -p /var/nfsshare/registry
sudo chmod -R 777 /var/nfsshare
sudo chown -R nfsnobody:nfsnobody /var/nfsshare
```

Create an NFS Export

```
sudo vi /etc/exports
```

Add this line in the new /etc/exports file “/var/nfss
192.168.1.0/24(rw,sync,no_root_squash,no_all_sq



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
```

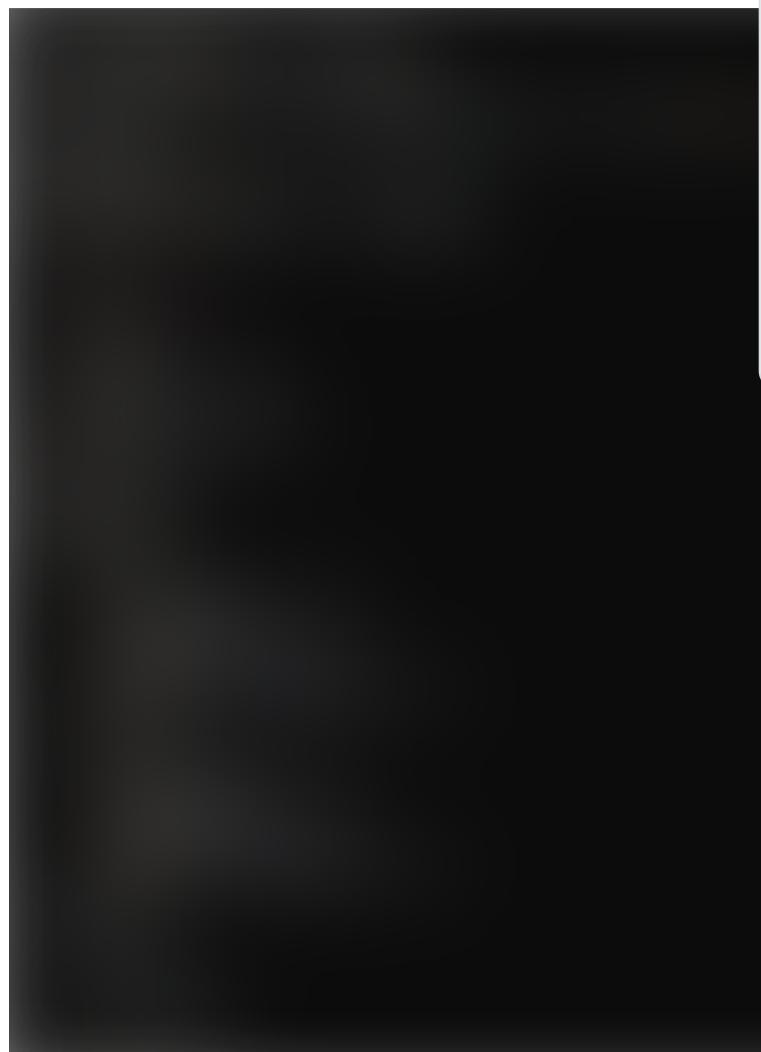
Edit the image-registry operator:

```
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:

```
managementStage: Managed
```

```
storage:  
pvc:  
claim:
```



Войдите в приложение "medium.com" с аккаунтом Google X

Sviatoslav Lachuzhnikov
lachuzhnikov@gmail.com

Продолжить как Sviatoslav

Чтобы создать аккаунт, Google предоставит приложению "medium.com" ваше имя, адрес электронной почты и фото профиля. См. [Политику конфиденциальности](#) и [Условия использования](#) сервиса "medium.com".

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

```
oc get pv
```

To test the registry and underlying storage, create a new php project in OKD and use the example repository. Check your NFS export on okd4-services for data:



...

Create Wordpress Project:

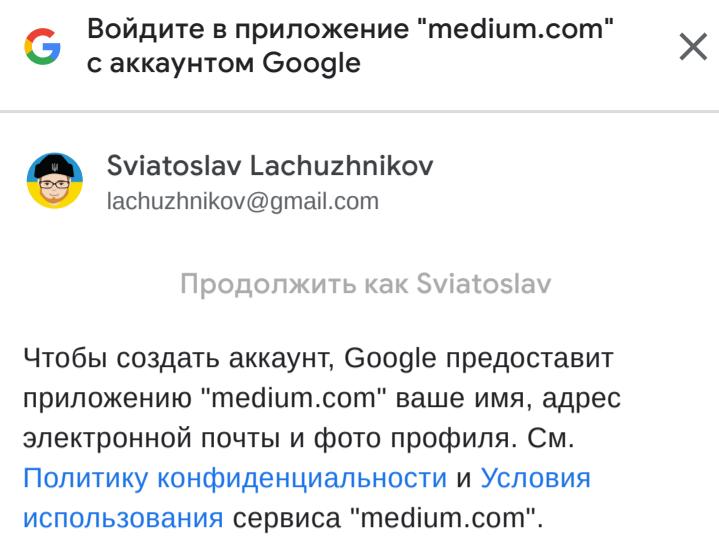
Create a new project.

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

Create a new app using the centos php73 s2i image from dockerhub and use the WordPress GitHub repo for the source. Also expose 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.

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

Open the route url and setup WordPress using mariadb (the service name) as the database host.

... . . .

HTPasswd Setup:

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

```
htpasswd -c -B -b </path/to/users.htpasswd> <user_name> <password>
```

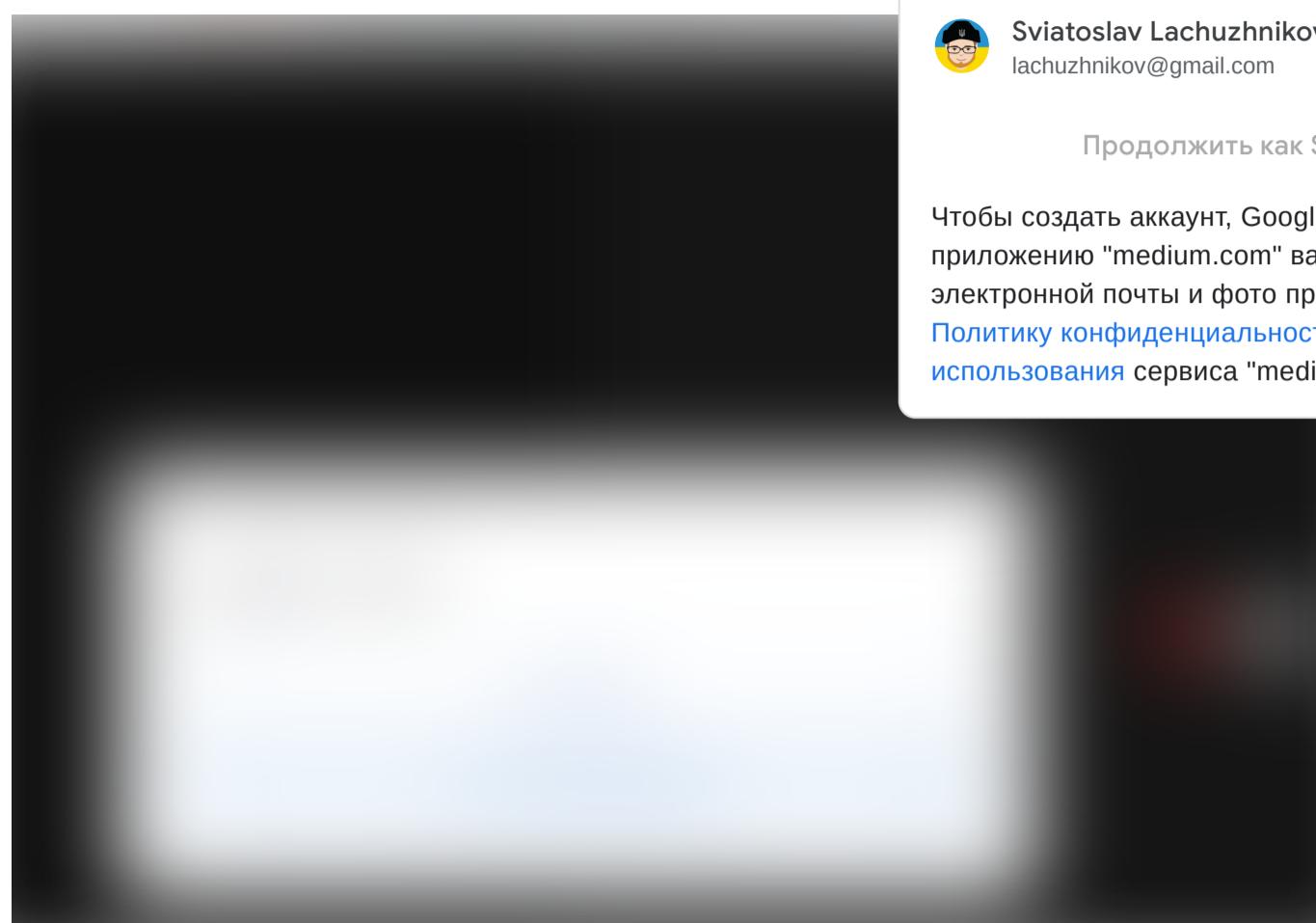
Create a secret in the openshift-config project using the users.htpasswd file you generated:

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

Add the identity provider.

```
oc apply -f okd4_files/htpasswd_provider.yaml
```

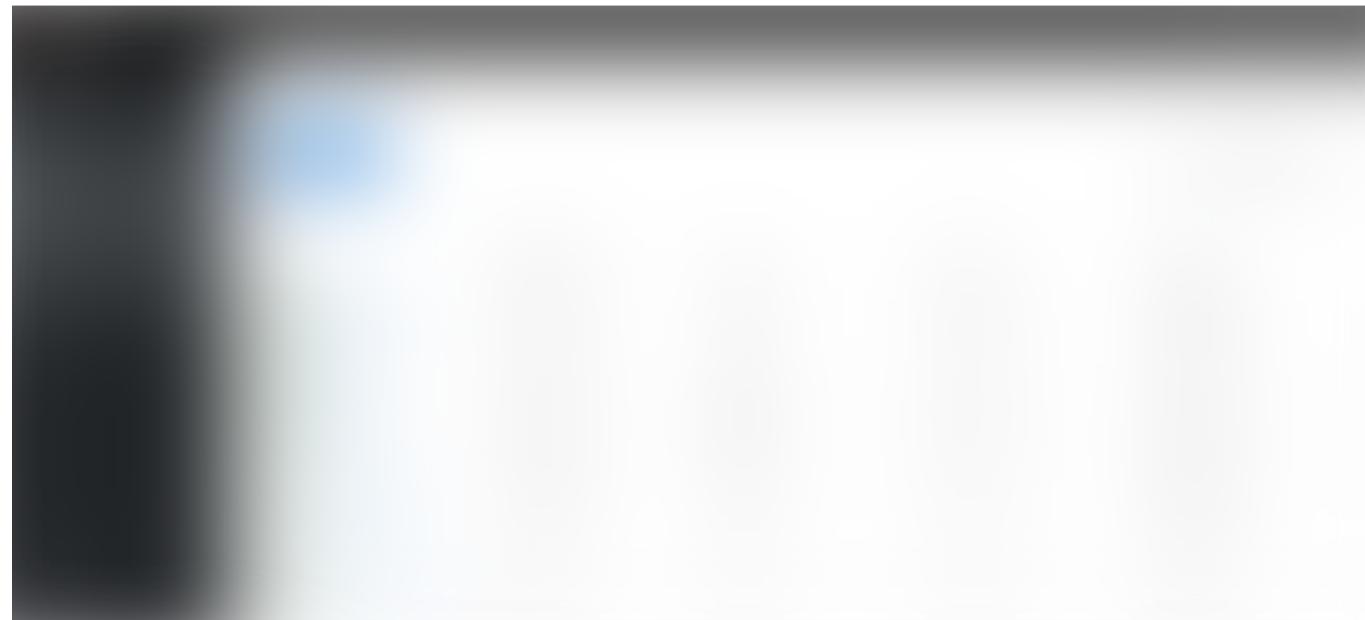
Login with my _htpasswd_provider using the username and password you created:



Bind the cluster-admin role to your user:

```
oc adm policy add-cluster-role-to-user cluster-admin <username>
```

Your user should now have cluster-admin level access:



...

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:

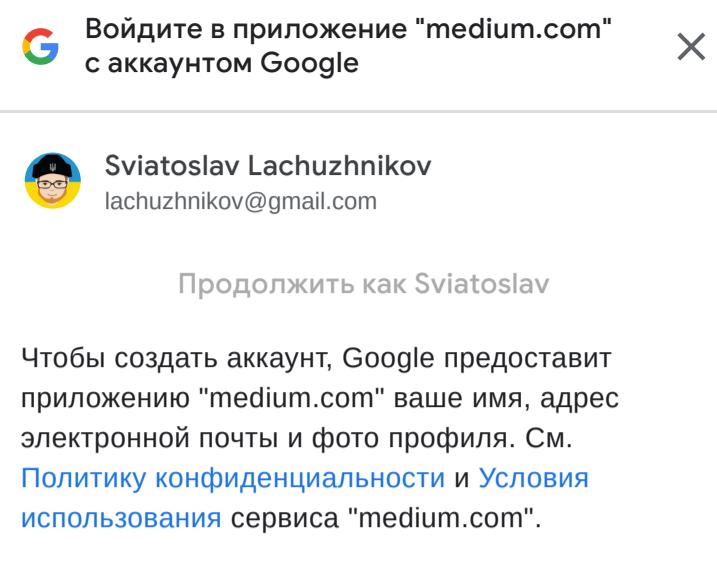
<https://groups.google.com/forum/#!forum/okd-wg>

• • •

This guide should get you up and started in the right direction. Good luck on your journey with OpenShift!

No rights reserved by the author. [\(\)](#)

[Openshift](#) [Okd](#) [Kubernetes](#) [Coreos](#) [Homelab](#)



Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. Watch

Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. Explore

Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just \$5/month. Upgrade

[About](#) [Help](#) [Legal](#)