OpenShift 4.17 Installation - Simple Command Guide

STEP 1: Prepare Your Bastion Computer

What this does: Gets your management computer ready with all the tools needed

Run these commands on your bastion server:

```
 # Update your system
sudo dnf update -y

# Install required tools
sudo dnf install -y git wget curl vim bind-utils telnet httpd-tools jq podman buildah skopeo firewalld dnsmasq httpd openssl

# Start firewall
sudo systemctl enable --now firewalld
```

STEP 2: Download OpenShift Files (Internet Required)

What this does: Downloads OpenShift software from Red Hat (do this on a computer with internet)

Run these commands on a connected computer:

```
 # Create download folder
mkdir -p ~/openshift-downloads
cd ~/openshift-downloads

# Download OpenShift tools
wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/4.17.0/openshift-client-linux.tar.gz
wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/4.17.0/openshift-install-linux.tar.gz
wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/4.17.0/oc-mirror.tar.gz

# Download operating system
wget https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.17/4.17.0/rhcos-4.17.0-x86_64-live.x86_64.iso

# Package everything
tar -czf openshift-tools.tar.gz *.tar.gz *.iso

# Transfer openshift-tools.tar.gz to your bastion computer
```

STEP 3: Install OpenShift Tools on Bastion

What this does: Installs the OpenShift management tools on your bastion computer

Run these commands on bastion:

```
# Extract tools
cd /tmp
tar -xzf openshift-tools.tar.gz
tar -xzf openshift-client-linux.tar.gz
tar -xzf openshift-install-linux.tar.gz
tar -xzf oc-mirror.tar.gz

# Install tools
sudo cp oc kubectl openshift-install oc-mirror /usr/local/bin/
sudo chmod +x /usr/local/bin/{oc,kubectl,openshift-install,oc-mirror}

# Check they work
oc version --client
openshift-install version
```

STEP 4: Create Your Local App Store (Container Registry)

What this does: Creates a local place to store OpenShift software since your cluster can't reach the internet

Run these commands on bastion:

```
# Create folders for registry
sudo mkdir -p /opt/registry/{data,certs,auth}

# Create security certificate
sudo openssl req -newkey rsa:4096 -nodes -sha256 \
  -keyout /opt/registry/certs/registry.key \
  -x509 -days 365 \
  -out /opt/registry/certs/registry.crt \
  -subj "/C=US/ST=State/L=City/O=Organization/CN=registry.example.com"

# Create login credentials
sudo htpasswd -bBc /opt/registry/auth/htpasswd admin password123

# Start the registry
sudo podman run -d --name local-registry \
  -p 5000:5000 \
  -v /opt/registry/data:/var/lib/registry:Z \
  -v /opt/registry/certs:/certs:Z \
  -v /opt/registry/auth:/auth:Z \
  -e REGISTRY_AUTH=htpasswd \
  -e REGISTRY_AUTH_HTPASSWD_REALM="Registry Realm" \
  -e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
  -e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/registry.crt \
  -e REGISTRY_HTTP_TLS_KEY=/certs/registry.key \
  docker.io/library/registry:2

# Make it start automatically
sudo systemctl enable podman
sudo podman generate systemd --name local-registry --files --new
sudo mv container-local-registry.service /etc/systemd/system/
sudo systemctl daemon-reload
sudo systemctl enable container-local-registry.service

# Trust the certificate
sudo cp /opt/registry/certs/registry.crt /etc/pki/ca-trust/source/anchors/
sudo update-ca-trust extract
```

STEP 5: Create Phone Book (DNS Setup)

What this does: Sets up name resolution so computers can find each other by name

Run these commands on bastion (replace IP addresses with your actual ones):

```
 # Configure DNS server
sudo tee /etc/dnsmasq.conf << 'EOF'
interface=*
bind-interfaces
domain=example.com
expand-hosts
no-resolv
no-poll

# Your server addresses (CHANGE THESE TO MATCH YOUR NETWORK)
address=/registry.example.com/192.168.1.100
address=/api.ocp4.example.com/192.168.1.101
address=/api-int.ocp4.example.com/192.168.1.101
address=/.apps.ocp4.example.com/192.168.1.102

cache-size=1000
log-queries
log-facility=/var/log/dnsmasq.log
EOF

# Create log file
sudo touch /var/log/dnsmasq.log
sudo chown dnsmasq:dnsmasq /var/log/dnsmasq.log

# Start DNS server
sudo systemctl enable --now dnsmasq

# Add computer names (CHANGE IP ADDRESSES TO MATCH YOUR SERVERS)
sudo tee -a /etc/hosts << 'EOF'
192.168.1.100 registry.example.com bastion.example.com
192.168.1.103 bootstrap.ocp4.example.com
192.168.1.104 master1.ocp4.example.com
192.168.1.105 master2.ocp4.example.com
192.168.1.106 master3.ocp4.example.com
192.168.1.107 worker1.ocp4.example.com
192.168.1.108 worker2.ocp4.example.com
192.168.1.101 api.ocp4.example.com api-int.ocp4.example.com
192.168.1.102 oauth-openshift.apps.ocp4.example.com console-openshift-console.apps.ocp4.example.com
EOF
```

STEP 6: Fill Your App Store (Copy OpenShift Software)

What this does: Downloads all OpenShift software and puts it in your local registry

Get your Red Hat credentials first:

1. Go to https://cloud.redhat.com/openshift/install/pull-secret
2. Download and save as `~/pull-secret.json`

Run these commands on bastion:

```
# Prepare credentials
REGISTRY_AUTH=$(echo -n 'admin:password123' | base64 -w0)
jq --arg auth "$REGISTRY_AUTH" \
  '.auths += {"registry.example.com:5000": {"auth": $auth}}' \
  ~/pull-secret.json > ~/combined-pull-secret.json

# Create mirror configuration
mkdir -p ~/mirror-config
cd ~/mirror-config
cat > imageset-config.yaml << 'EOF'
apiVersion: mirror.openshift.io/v1alpha2
kind: ImageSetConfiguration
metadata:
  name: openshift-4-17-mirror
mirror:
  platform:
    channels:
    - name: stable-4.17
      type: ocp
      minVersion: 4.17.0
      maxVersion: 4.17.0
    graph: true
  operators:
  - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.17
    packages:
    - name: local-storage-operator
  additionalImages:
  - name: registry.redhat.io/ubi8/ubi:latest
  helm: {}
EOF

# Login to registries
podman login --authfile ~/combined-pull-secret.json registry.redhat.io
podman login --authfile ~/combined-pull-secret.json quay.io
podman login --authfile ~/combined-pull-secret.json -u admin -p password123 registry.example.com:5000

# Copy all images (THIS TAKES 1-3 HOURS)
oc mirror --config=imageset-config.yaml \
  docker://registry.example.com:5000 \
  --dest-skip-tls \
  --continue-on-error
```

STEP 7: Create Door Keys (SSH Keys)

What this does: Creates keys so you can log into your servers if needed

Run these commands on bastion:

```
# Create SSH keys
ssh-keygen -t ed25519 -N '' -f ~/.ssh/openshift-key

# Add to SSH agent
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/openshift-key

# Show your public key (you'll need this later)
cat ~/.ssh/openshift-key.pub
```

STEP 8: Write Your Cluster Plan

What this does: Creates the blueprint for your OpenShift cluster

Run these commands on bastion:

```
 # Create installation folder
mkdir ~/openshift-install-dir
cd ~/openshift-install-dir

# Create cluster configuration (CHANGE NETWORK ADDRESSES TO MATCH YOURS)
cat > install-config.yaml << EOF
apiVersion: v1
baseDomain: example.com
metadata:
  name: ocp4
networking:
  networkType: OVNKubernetes
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork:
  - 172.30.0.0/16
  machineNetwork:
  - cidr: 192.168.1.0/24
compute:
- name: worker
  replicas: 2
controlPlane:
  name: master
  replicas: 3
platform:
  none: {}
pullSecret: |
$(cat ~/combined-pull-secret.json | jq -c .)
sshKey: |
$(cat ~/.ssh/openshift-key.pub)
imageContentSources:
- mirrors:
  - registry.example.com:5000/openshift/release-images
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - registry.example.com:5000/openshift/release
  source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
additionalTrustBundle: |
$(cat /opt/registry/certs/registry.crt | sed 's/^/  /')
EOF

# Backup your configuration
cp install-config.yaml install-config.yaml.backup
```

STEP 9: Create Detailed Instructions

What this does: Turns your plan into specific instructions for each server

Run these commands on bastion:

```
# Create detailed plans
cd ~/openshift-install-dir
openshift-install create manifests --dir .

# For production, make masters not run applications (optional)
sed -i 's/mastersSchedulable: true/mastersSchedulable: false/' manifests/cluster-scheduler-02-config.yml

# Create final instructions for each server
openshift-install create ignition-configs --dir .

# You should now see these files:
ls -la *.ign
# bootstrap.ign - instructions for bootstrap server
# master.ign - instructions for master servers
# worker.ign - instructions for worker servers
```

STEP 10: Set Up Instruction Delivery

What this does: Creates a web server to deliver instructions to the bootstrap server

Run these commands on bastion:

```
# Start web server
sudo systemctl enable --now httpd

# Open firewall
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https
sudo firewall-cmd --permanent --add-service=dns
sudo firewall-cmd --permanent --add-port=5000/tcp
sudo firewall-cmd --permanent --add-port=6443/tcp
sudo firewall-cmd --permanent --add-port=22623/tcp
sudo firewall-cmd --reload

# Copy bootstrap instructions to web server
sudo cp bootstrap.ign /var/www/html/
sudo chown apache:apache /var/www/html/bootstrap.ign
sudo chmod 644 /var/www/html/bootstrap.ign

# Test it works
curl -s http://$(hostname -I | awk '{print $1}')/bootstrap.ign | head -5
```

STEP 11: Install Operating System on Servers

What this does: Installs Red Hat CoreOS on all your physical servers

For each server, boot from the RHCOS ISO and run:

Bootstrap Server:

```
sudo coreos-installer install \
  --ignition-url=http://192.168.1.100/bootstrap.ign \
  /dev/sda
```

Master Servers (run on each of 3 master servers):

```
sudo coreos-installer install \
  --ignition-url=data:text/plain;charset=utf-8;base64,$(base64 -w0 /path/to/master.ign) \
  /dev/sda
```

Worker Servers (run on each worker server):

```
sudo coreos-installer install \
  --ignition-url=data:text/plain;charset=utf-8;base64,$(base64 -w0 /path/to/worker.ign) \
  /dev/sda
```

STEP 12: Watch Your Cluster Build Itself

What this does: Monitors the automatic installation process

Run these commands on bastion:

```
 # Set up access to your cluster
cd ~/openshift-install-dir
export KUBECONFIG=~/openshift-install-dir/auth/kubeconfig

# Wait for bootstrap to finish (20-30 minutes)
openshift-install wait-for bootstrap-complete --log-level=debug

# Watch the progress
tail -f .openshift_install.log
```

STEP 13: Let Workers Join the Team

What this does: Gives permission for worker servers to join your cluster

Run these commands on bastion:

```
 # Check for workers waiting to join
oc get csr

# Give them permission (run this several times as new requests appear)
oc get csr -o name | xargs oc adm certificate approve

# Watch for new requests
watch 'oc get csr | grep Pending'
```

STEP 14: Wait for Everything to Finish

What this does: Waits for all cluster services to start properly

Run these commands on bastion:

```
 # Wait for installation to complete (30-60 minutes total)
openshift-install wait-for install-complete --log-level=debug

# Check everything is working
oc get nodes
oc get clusteroperators
```

STEP 15: Set Up Internal Storage

What this does: Configures OpenShift's internal image storage

Run these commands on bastion:

```
# Configure storage for images
oc patch configs.imageregistry.operator.openshift.io cluster \
  --type merge --patch '{"spec":{"storage":{"emptyDir":{}}}}'

# Enable the registry
oc patch configs.imageregistry.operator.openshift.io cluster \
  --type merge --patch '{"spec":{"managementState":"Managed"}}'
```

STEP 16: Test Everything Works

What this does: Verifies your cluster is healthy and ready to use

Run these commands on bastion:

```
# Check cluster health
oc get nodes
oc get clusteroperators
oc get pods --all-namespaces | grep -v Running | grep -v Completed

# Get web console access
echo "Web Console: https://$(oc get routes console -n openshift-console -o jsonpath='{.spec.host}')"
echo "Username: kubeadmin"
echo "Password: $(cat ~/openshift-install-dir/auth/kubeadmin-password)"
```

Summary

Your OpenShift cluster is now ready! You can:

- Access the web console using the URL and credentials from Step 16
- Use `oc` commands to manage your cluster
- Deploy applications to your cluster

Important files to keep safe:

- `~/openshift-install-dir/auth/kubeconfig` - Access to your cluster
- `~/openshift-install-dir/auth/kubeadmin-password` - Admin password
- `~/.ssh/openshift-key` - SSH access to servers
- `/opt/registry/certs/registry.crt` - Registry certificate