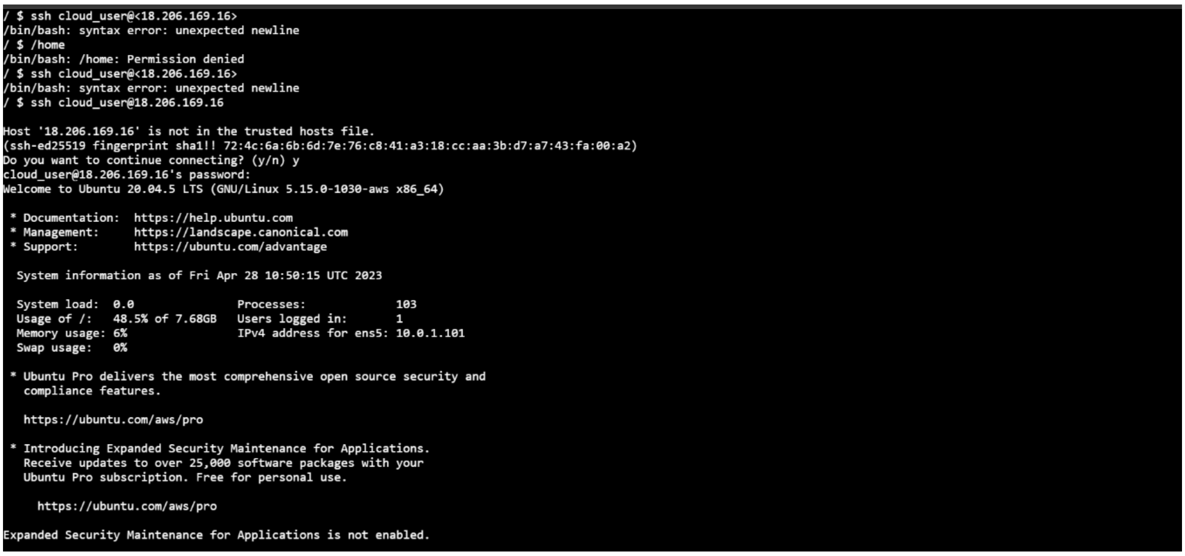
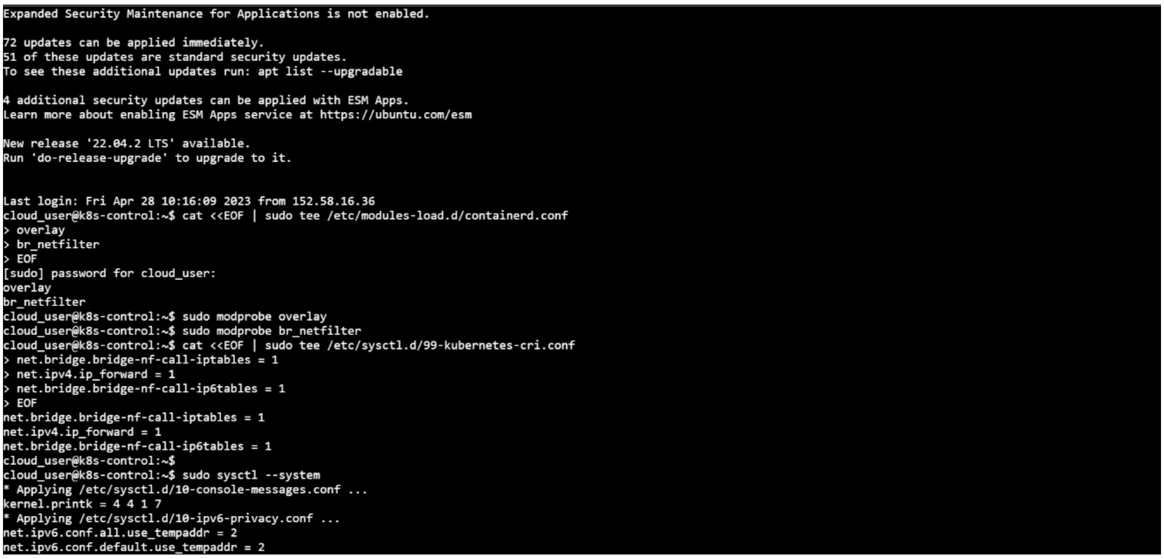
Building a Kubernetes 1.24 Cluster with Kubeadm

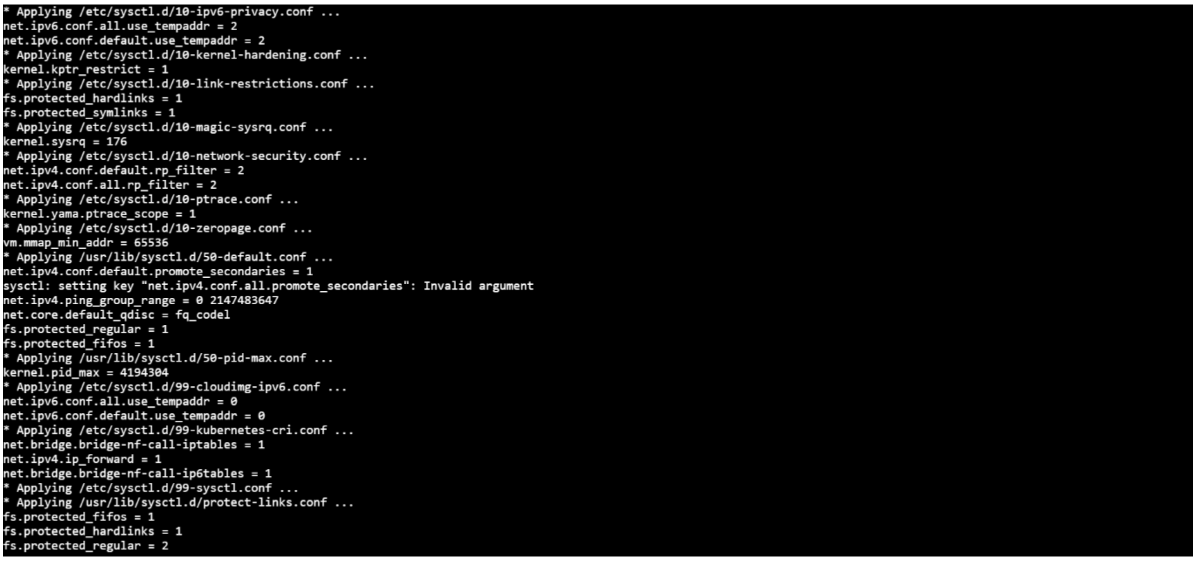
- **Packages Installation.**



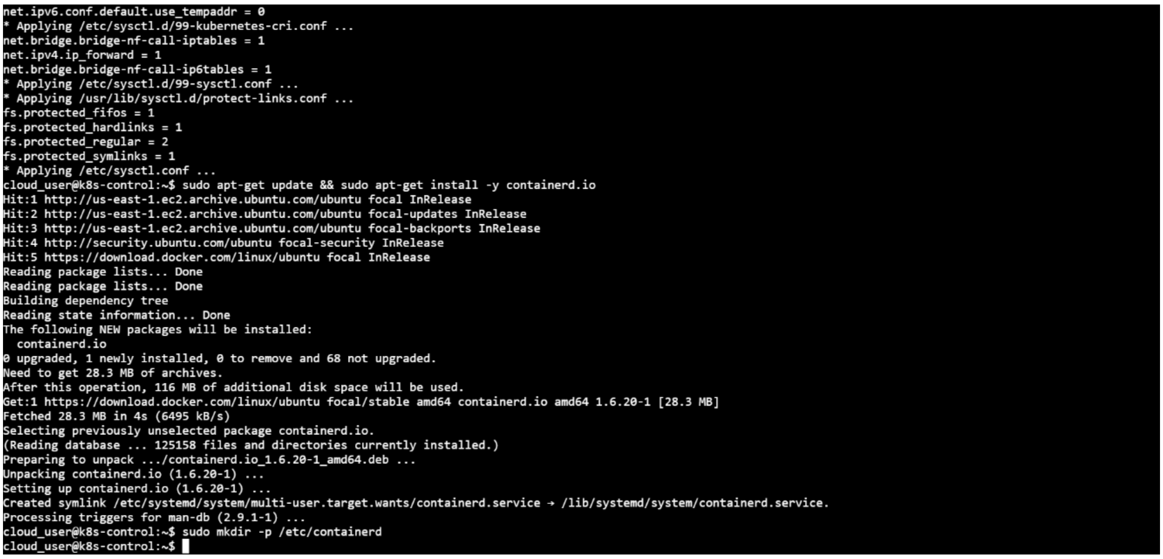
```
/ $ ssh cloud_user@<18.206.169.16>
/bin/bash: syntax error: unexpected newline
/ $ /home
/bin/bash: /home: Permission denied
/ $ ssh cloud_user@<18.206.169.16>
/bin/bash: syntax error: unexpected newline
/ $ ssh cloud_user@18.206.169.16

Host '18.206.169.16' is not in the trusted hosts file.
(ssh-ed25519 fingerprint sha1!! 72:4c:6a:6b:6d:7e:76:c8:41:a3:18:cc:aa:3b:d7:a7:43:fa:00:a2)
Do you want to continue connecting? (y/n) y
cloud_user@18.206.169.16's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1030-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Apr 28 10:50:15 UTC 2023

  System load:  0.0               Processes:             103
  Usage of /:   48.5% of 7.68GB   Users logged in:       1
  Memory usage: 6%                IPv4 address for ens5: 10.0.1.101
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.
```



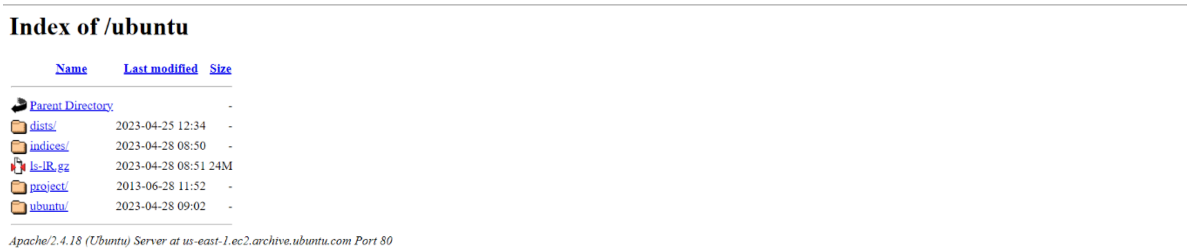
```
Expanded Security Maintenance for Applications is not enabled.

72 updates can be applied immediately.
51 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

4 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Apr 28 10:16:09 2023 from 152.58.16.36
cloud_user@k8s-control:~$ cat <<EOF | sudo tee /etc/modules-load.d/containerd.conf
> overlay
> br_netfilter
> EOF
[sudo] password for cloud_user:
overlay
br_netfilter
cloud_user@k8s-control:~$ sudo modprobe overlay
cloud_user@k8s-control:~$ sudo modprobe br_netfilter
cloud_user@k8s-control:~$ cat <<EOF | sudo tee /etc/sysctl.d/99-kubernetes-cri.conf
> net.bridge.bridge-nf-call-iptables = 1
> net.ipv4.ip_forward = 1
> net.bridge.bridge-nf-call-ip6tables = 1
> EOF
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
net.bridge.bridge-nf-call-ip6tables = 1
cloud_user@k8s-control:~$
cloud_user@k8s-control:~$ sudo sysctl --system
* Applying /etc/sysctl.d/10-console-messages.conf ...
kernel.printk = 4 4 1 7
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
```



```
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
kernel.kptr_restrict = 1
* Applying /etc/sysctl.d/10-link-restrictions.conf ...
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
* Applying /etc/sysctl.d/10-magic-sysrq.conf ...
kernel.sysrq = 176
* Applying /etc/sysctl.d/10-network-security.conf ...
net.ipv4.conf.default.rp_filter = 2
net.ipv4.conf.all.rp_filter = 2
* Applying /etc/sysctl.d/10-ptrace.conf ...
kernel.yama.ptrace_scope = 1
* Applying /etc/sysctl.d/10-zeropage.conf ...
vm.mmap_min_addr = 65536
* Applying /usr/lib/sysctl.d/50-default.conf ...
net.ipv4.conf.default.promote_secondaries = 1
sysctl: setting key "net.ipv4.conf.all.promote_secondaries": Invalid argument
net.ipv4.ping_group_range = 0 2147483647
net.core.default_qdisc = fq_codel
fs.protected_regular = 1
fs.protected_fifos = 1
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
kernel.pid_max = 4194304
* Applying /etc/sysctl.d/99-cloudimg-ipv6.conf ...
net.ipv6.conf.all.use_tempaddr = 0
net.ipv6.conf.default.use_tempaddr = 0
* Applying /etc/sysctl.d/99-kubernetes-cri.conf ...
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
net.bridge.bridge-nf-call-ip6tables = 1
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /usr/lib/sysctl.d/protect-links.conf ...
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
```



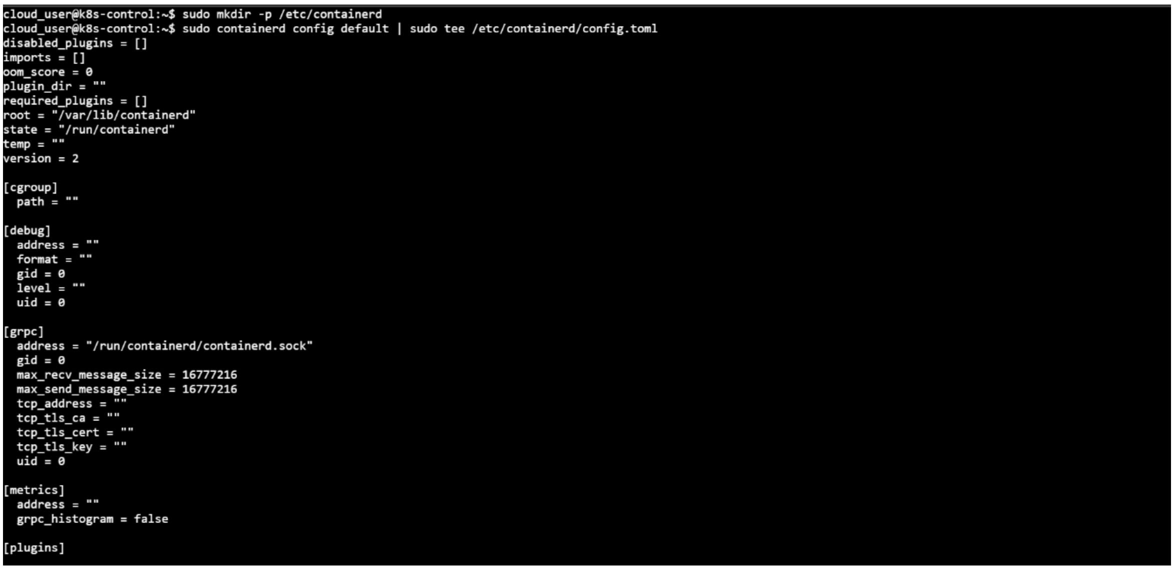
```
net.ipv6.conf.default.use_tempaddr = 0
* Applying /etc/sysctl.d/99-kubernetes-cri.conf ...
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
net.bridge.bridge-nf-call-ip6tables = 1
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /usr/lib/sysctl.d/protect-links.conf ...
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
fs.protected_symlinks = 1
* Applying /etc/sysctl.conf ...
cloud_user@k8s-control:~$ sudo apt-get update && sudo apt-get install -y containerd.io
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu focal InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  containerd.io
0 upgraded, 1 newly installed, 0 to remove and 68 not upgraded.
Need to get 28.3 MB of archives.
After this operation, 116 MB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.6.20-1 [28.3 MB]
Fetched 28.3 MB in 4s (6495 kB/s)
Selecting previously unselected package containerd.io.
(Reading database ... 125158 files and directories currently installed.)
Preparing to unpack .../containerd.io_1.6.20-1_amd64.deb ...
Unpacking containerd.io (1.6.20-1) ...
Setting up containerd.io (1.6.20-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Processing triggers for man-db (2.9.1-1) ...
cloud_user@k8s-control:~$ sudo mkdir -p /etc/containerd
cloud_user@k8s-control:~$ 
```



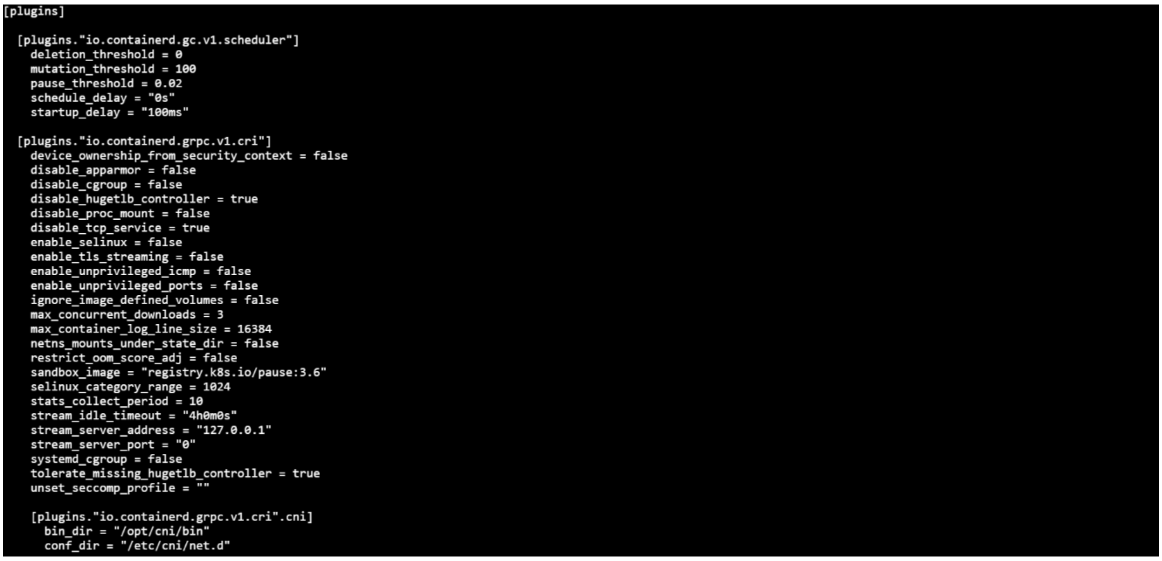
Index of /ubuntu

Name	Last modified	Size
Parent Directory		-
dists/	2023-04-25 12:34	-
indices/	2023-04-28 08:50	-
ls-lR.gz	2023-04-28 08:51	24M
project/	2013-06-28 11:52	-
ubuntu/	2023-04-28 09:02	-

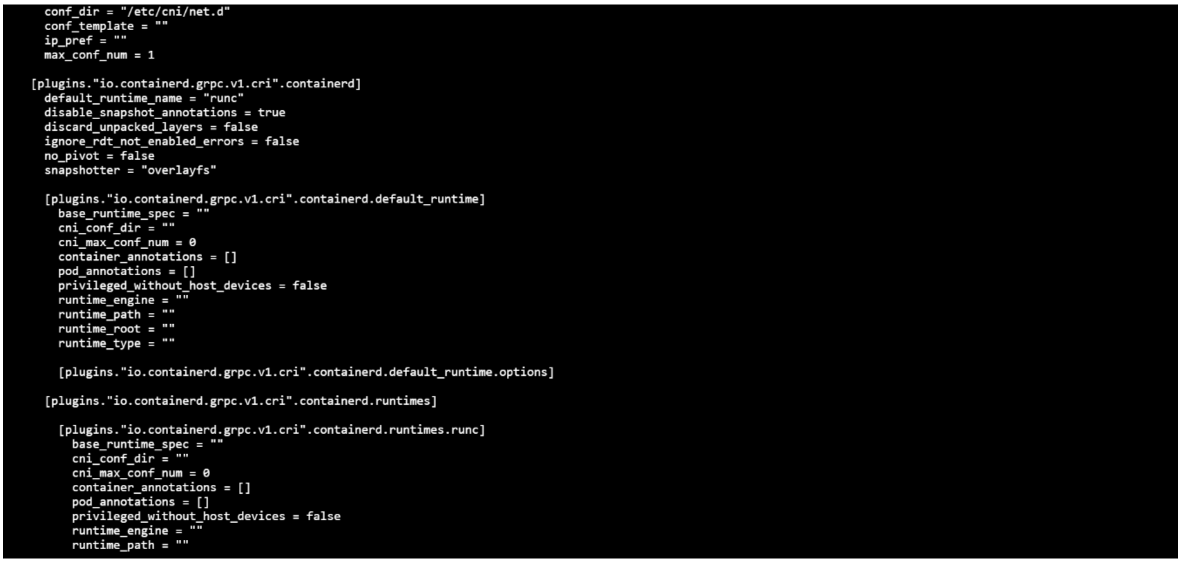
Apache/2.4.18 (Ubuntu) Server at us-east-1.ec2.archive.ubuntu.com Port 80



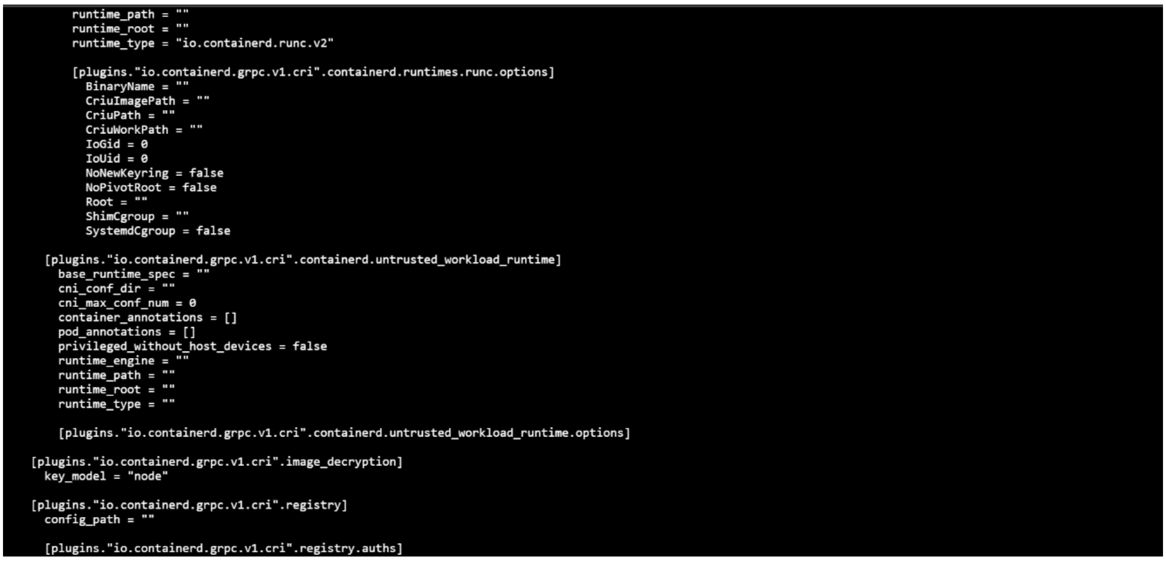
```
cloud_user@k8s-control:~$ sudo mkdir -p /etc/containerd
cloud_user@k8s-control:~$ sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

[debug]
  address = ""
  format = ""
  gid = 0
  level = ""
  uid = 0

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
  tcp_address = ""
  tcp_tls_ca = ""
  tcp_tls_cert = ""
  tcp_tls_key = ""
  uid = 0

[metrics]
  address = ""
  grpc_histogram = false

[plugins]
```



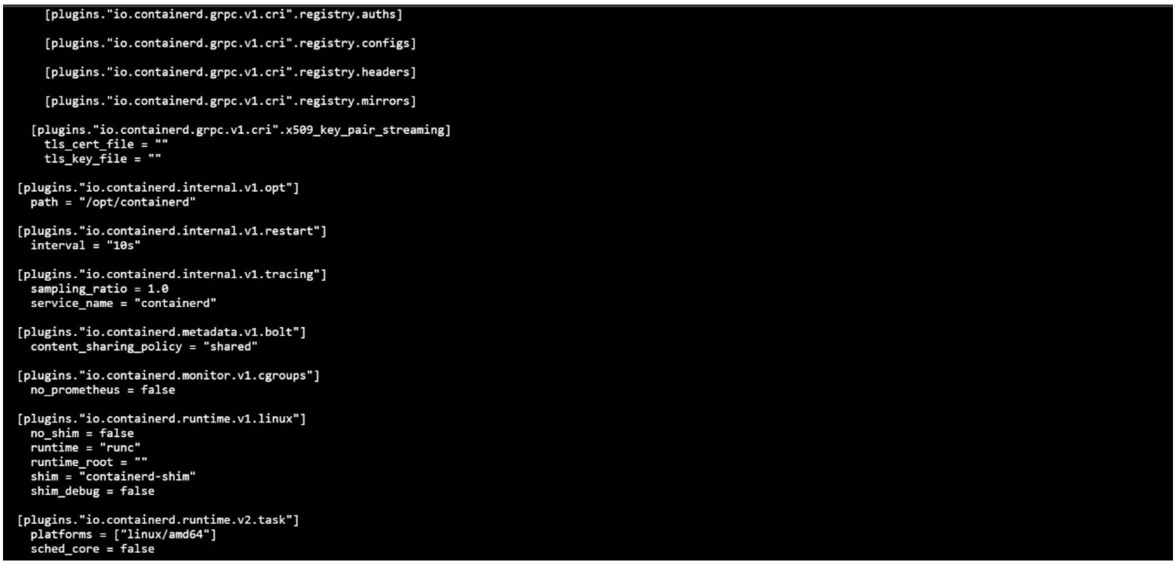
```
[plugins]

  [plugins."io.containerd.gc.v1.scheduler"]
    deletion_threshold = 0
    mutation_threshold = 100
    pause_threshold = 0.02
    schedule_delay = "0s"
    startup_delay = "100ms"

  [plugins."io.containerd.grpc.v1.cri"]
    device_ownership_from_security_context = false
    disable_apparmor = false
    disable_cgroup = false
    disable_hugetlb_controller = true
    disable_proc_mount = false
    disable_tcp_service = true
    enable_selinux = false
    enable_tls_streaming = false
    enable_unprivileged_icmp = false
    enable_unprivileged_ports = false
    ignore_image_defined_volumes = false
    max_concurrent_downloads = 3
    max_container_log_line_size = 16384
    netns_mounts_under_state_dir = false
    restrict_oom_score_adj = false
    sandbox_image = "registry.k8s.io/pause:3.6"
    selinux_category_range = 1024
    stats_collect_period = 10
    stream_idle_timeout = "4h0m0s"
    stream_server_address = "127.0.0.1"
    stream_server_port = "0"
    systemd_cgroup = false
    tolerate_missing_hugetlb_controller = true
    unset_seccomp_profile = ""

    [plugins."io.containerd.grpc.v1.cri".cni]
      bin_dir = "/opt/cni/bin"
      conf_dir = "/etc/cni/net.d"
```



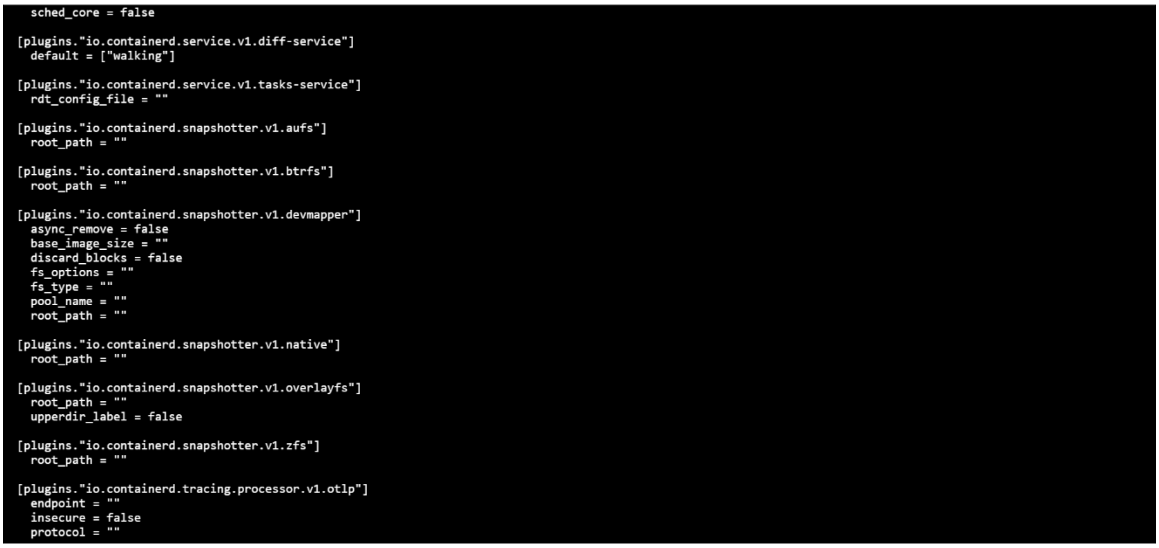
```
      conf_dir = "/etc/cni/net.d"
      conf_template = ""
      ip_pref = ""
      max_conf_num = 1

    [plugins."io.containerd.grpc.v1.cri".containerd]
      default_runtime_name = "runc"
      disable_snapshot_annotations = true
      discard_unpacked_layers = false
      ignore_rdt_not_enabled_errors = false
      no_pivot = false
      snapshotter = "overlayfs"

      [plugins."io.containerd.grpc.v1.cri".containerd.default_runtime]
        base_runtime_spec = ""
        cni_conf_dir = ""
        cni_max_conf_num = 0
        container_annotations = []
        pod_annotations = []
        privileged_without_host_devices = false
        runtime_engine = ""
        runtime_path = ""
        runtime_root = ""
        runtime_type = ""

        [plugins."io.containerd.grpc.v1.cri".containerd.default_runtime.options]

      [plugins."io.containerd.grpc.v1.cri".containerd.runtimes]

        [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc]
          base_runtime_spec = ""
          cni_conf_dir = ""
          cni_max_conf_num = 0
          container_annotations = []
          pod_annotations = []
          privileged_without_host_devices = false
          runtime_engine = ""
          runtime_path = ""
```



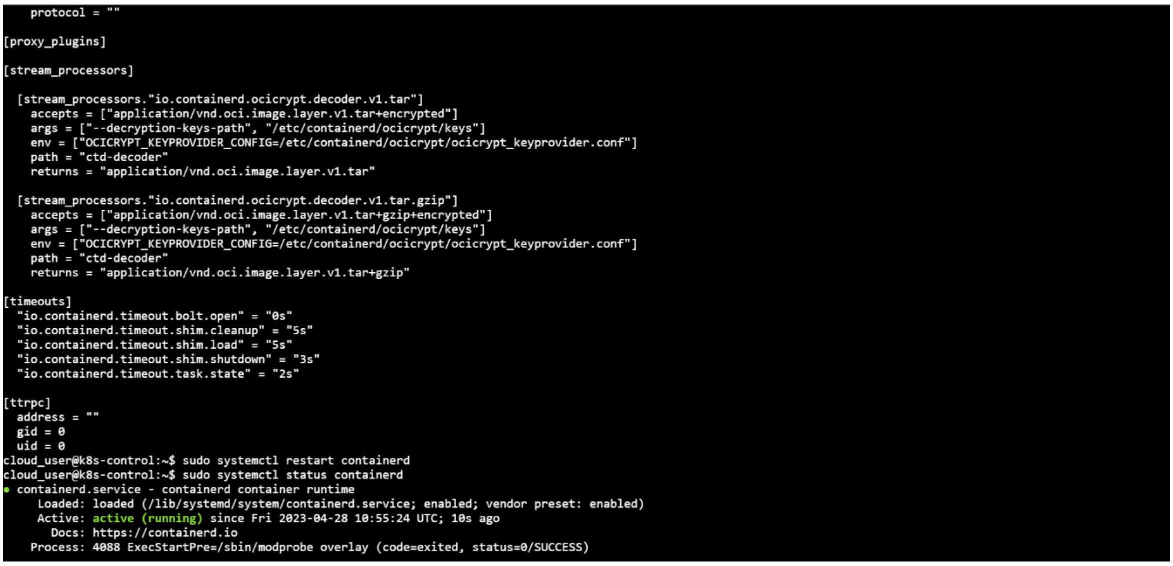
```
          runtime_path = ""
          runtime_root = ""
          runtime_type = "io.containerd.runc.v2"

          [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
            BinaryName = ""
            CriuImagePath = ""
            CriuPath = ""
            CriuWorkPath = ""
            IoGid = 0
            IoUid = 0
            NoNewKeyring = false
            NoPivotRoot = false
            Root = ""
            ShimCgroup = ""
            SystemdCgroup = false

      [plugins."io.containerd.grpc.v1.cri".containerd.untrusted_workload_runtime]
        base_runtime_spec = ""
        cni_conf_dir = ""
        cni_max_conf_num = 0
        container_annotations = []
        pod_annotations = []
        privileged_without_host_devices = false
        runtime_engine = ""
        runtime_path = ""
        runtime_root = ""
        runtime_type = ""

        [plugins."io.containerd.grpc.v1.cri".containerd.untrusted_workload_runtime.options]

    [plugins."io.containerd.grpc.v1.cri".image_decryption]
      key_model = "node"

    [plugins."io.containerd.grpc.v1.cri".registry]
      config_path = ""

      [plugins."io.containerd.grpc.v1.cri".registry.auths]
```



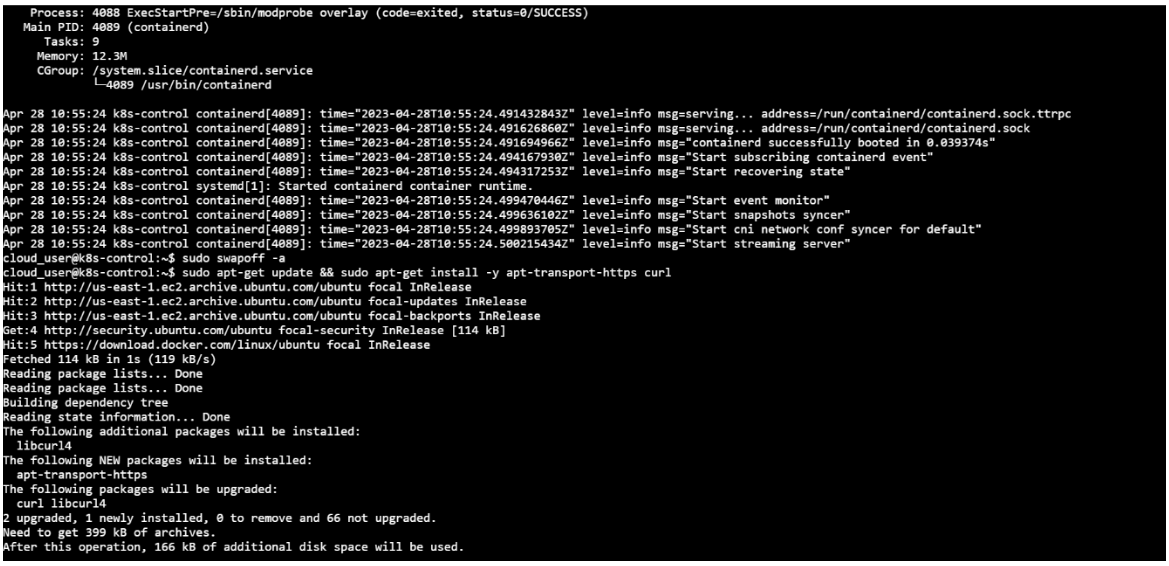
```
      [plugins."io.containerd.grpc.v1.cri".registry.auths]

      [plugins."io.containerd.grpc.v1.cri".registry.configs]

      [plugins."io.containerd.grpc.v1.cri".registry.headers]

      [plugins."io.containerd.grpc.v1.cri".registry.mirrors]

    [plugins."io.containerd.grpc.v1.cri".x509_key_pair_streaming]
      tls_cert_file = ""
      tls_key_file = ""

  [plugins."io.containerd.internal.v1.opt"]
    path = "/opt/containerd"

  [plugins."io.containerd.internal.v1.restart"]
    interval = "10s"

  [plugins."io.containerd.internal.v1.tracing"]
    sampling_ratio = 1.0
    service_name = "containerd"

  [plugins."io.containerd.metadata.v1.bolt"]
    content_sharing_policy = "shared"

  [plugins."io.containerd.monitor.v1.cgroups"]
    no_prometheus = false

  [plugins."io.containerd.runtime.v1.linux"]
    no_shim = false
    runtime = "runc"
    runtime_root = ""
    shim = "containerd-shim"
    shim_debug = false

  [plugins."io.containerd.runtime.v2.task"]
    platforms = ["linux/amd64"]
    sched_core = false
```



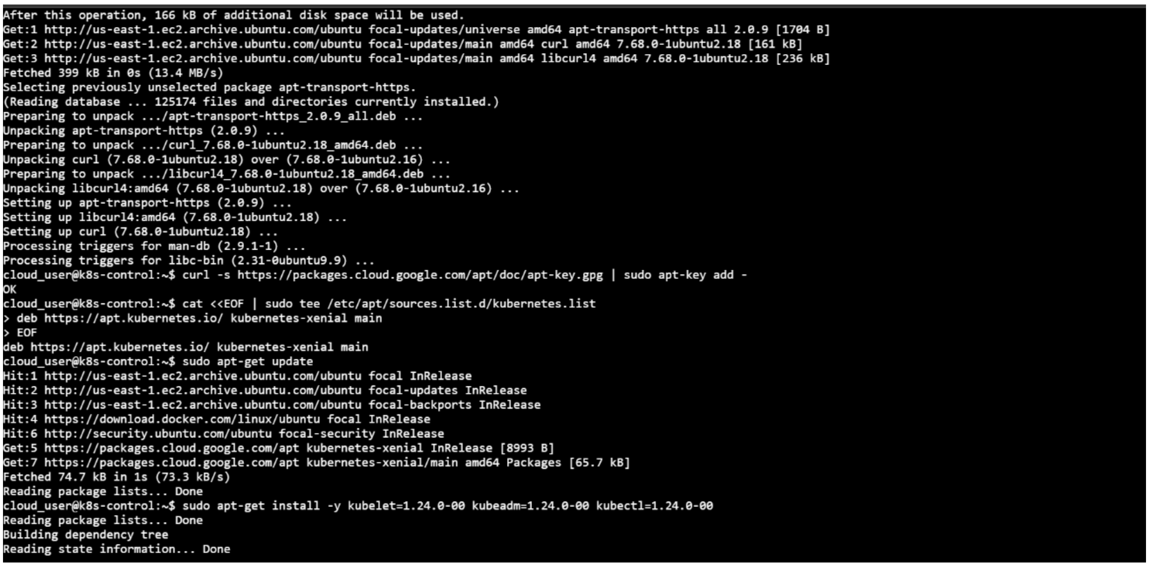
```
    sched_core = false

  [plugins."io.containerd.service.v1.diff-service"]
    default = ["walking"]

  [plugins."io.containerd.service.v1.tasks-service"]
    rdt_config_file = ""

  [plugins."io.containerd.snapshotter.v1.aufs"]
    root_path = ""

  [plugins."io.containerd.snapshotter.v1.btrfs"]
    root_path = ""

  [plugins."io.containerd.snapshotter.v1.devmapper"]
    async_remove = false
    base_image_size = ""
    discard_blocks = false
    fs_options = ""
    fs_type = ""
    pool_name = ""
    root_path = ""

  [plugins."io.containerd.snapshotter.v1.native"]
    root_path = ""

  [plugins."io.containerd.snapshotter.v1.overlayfs"]
    root_path = ""
    upperdir_label = false

  [plugins."io.containerd.snapshotter.v1.zfs"]
    root_path = ""

  [plugins."io.containerd.tracing.processor.v1.otlp"]
    endpoint = ""
    insecure = false
    protocol = ""
```



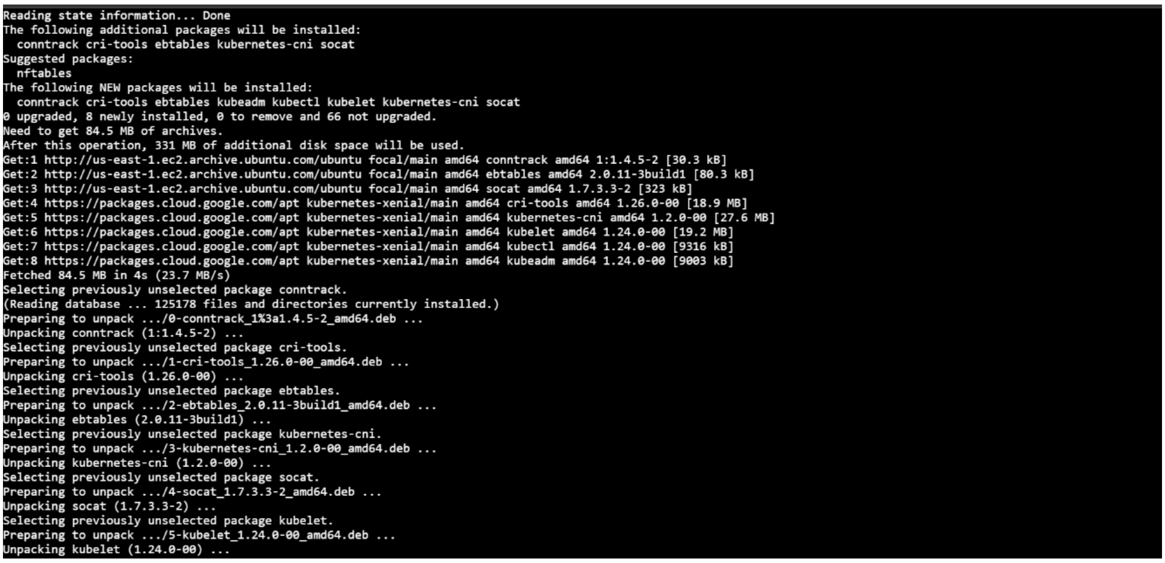
```
    protocol = ""

[proxy_plugins]

[stream_processors]

  [stream_processors."io.containerd.ocicrypt.decoder.v1.tar"]
    accepts = ["application/vnd.oci.image.layer.v1.tar+encrypted"]
    args = ["--decryption-keys-path", "/etc/containerd/ocicrypt/keys"]
    env = ["OCICRYPT_KEYPROVIDER_CONFIG=/etc/containerd/ocicrypt/ocicrypt_keyprovider.conf"]
    path = "ctd-decoder"
    returns = "application/vnd.oci.image.layer.v1.tar"

  [stream_processors."io.containerd.ocicrypt.decoder.v1.tar.gzip"]
    accepts = ["application/vnd.oci.image.layer.v1.tar+gzip+encrypted"]
    args = ["--decryption-keys-path", "/etc/containerd/ocicrypt/keys"]
    env = ["OCICRYPT_KEYPROVIDER_CONFIG=/etc/containerd/ocicrypt/ocicrypt_keyprovider.conf"]
    path = "ctd-decoder"
    returns = "application/vnd.oci.image.layer.v1.tar+gzip"

[timeouts]
  "io.containerd.timeout.bolt.open" = "0s"
  "io.containerd.timeout.shim.cleanup" = "5s"
  "io.containerd.timeout.shim.load" = "5s"
  "io.containerd.timeout.shim.shutdown" = "3s"
  "io.containerd.timeout.task.state" = "2s"

[ttrpc]
  address = ""
  gid = 0
  uid = 0
cloud_user@k8s-control:~$ sudo systemctl restart containerd
cloud_user@k8s-control:~$ sudo systemctl status containerd
● containerd.service - containerd container runtime
     Loaded: loaded (/lib/systemd/system/containerd.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-04-28 10:55:24 UTC; 10s ago
       Docs: https://containerd.io
    Process: 4088 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
```



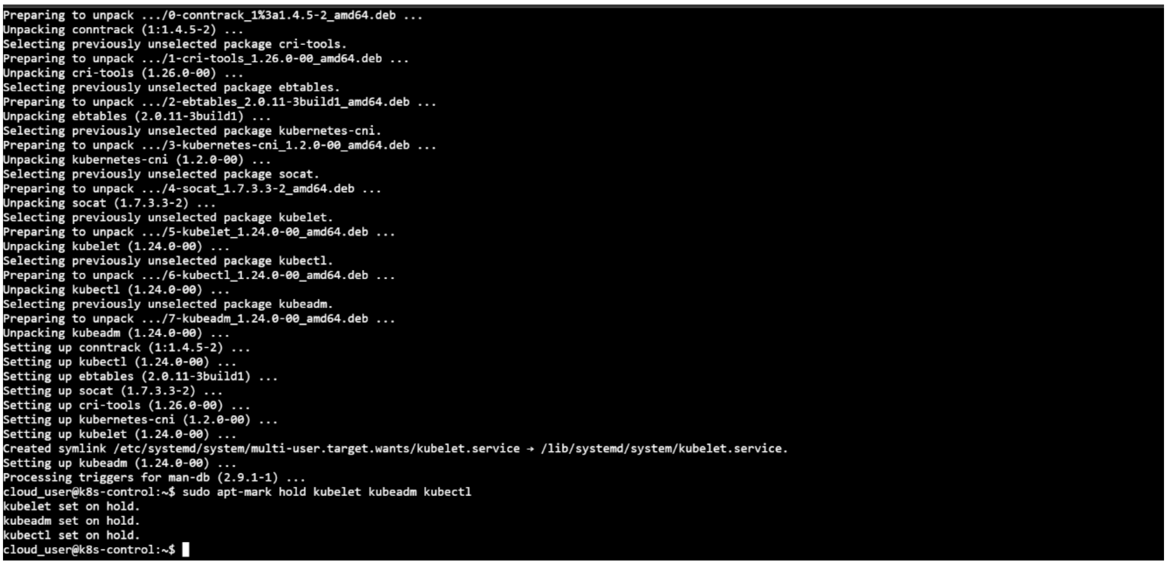
```
    Process: 4088 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
   Main PID: 4089 (containerd)
      Tasks: 9
     Memory: 12.3M
     CGroup: /system.slice/containerd.service
             └─4089 /usr/bin/containerd

Apr 28 10:55:24 k8s-control containerd[4089]: time="2023-04-28T10:55:24.491432843Z" level=info msg=serving... address=/run/containerd/containerd.sock.ttrpc
Apr 28 10:55:24 k8s-control containerd[4089]: time="2023-04-28T10:55:24.491626860Z" level=info msg=serving... address=/run/containerd/containerd.sock
Apr 28 10:55:24 k8s-control containerd[4089]: time="2023-04-28T10:55:24.491694966Z" level=info msg="containerd successfully booted in 0.039374s"
Apr 28 10:55:24 k8s-control containerd[4089]: time="2023-04-28T10:55:24.494167930Z" level=info msg="Start subscribing containerd event"
Apr 28 10:55:24 k8s-control containerd[4089]: time="2023-04-28T10:55:24.494317253Z" level=info msg="Start recovering state"
Apr 28 10:55:24 k8s-control systemd[1]: Started containerd container runtime.
Apr 28 10:55:24 k8s-control containerd[4089]: time="2023-04-28T10:55:24.499470446Z" level=info msg="Start event monitor"
Apr 28 10:55:24 k8s-control containerd[4089]: time="2023-04-28T10:55:24.499636102Z" level=info msg="Start snapshots syncer"
Apr 28 10:55:24 k8s-control containerd[4089]: time="2023-04-28T10:55:24.499893705Z" level=info msg="Start cni network conf syncer for default"
Apr 28 10:55:24 k8s-control containerd[4089]: time="2023-04-28T10:55:24.500215434Z" level=info msg="Start streaming server"
cloud_user@k8s-control:~$ sudo swapoff -a
cloud_user@k8s-control:~$ sudo apt-get update && sudo apt-get install -y apt-transport-https curl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:5 https://download.docker.com/linux/ubuntu focal InRelease
Fetched 114 kB in 1s (119 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl4
2 upgraded, 1 newly installed, 0 to remove and 66 not upgraded.
Need to get 399 kB of archives.
After this operation, 166 kB of additional disk space will be used.
```



```
After this operation, 166 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-transport-https all 2.0.9 [1704 B]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.18 [161 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libcurl4 amd64 7.68.0-1ubuntu2.18 [236 kB]
Fetched 399 kB in 0s (13.4 MB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 125174 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.9_all.deb ...
Unpacking apt-transport-https (2.0.9) ...
Preparing to unpack .../curl_7.68.0-1ubuntu2.18_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.18) over (7.68.0-1ubuntu2.16) ...
Preparing to unpack .../libcurl4_7.68.0-1ubuntu2.18_amd64.deb ...
Unpacking libcurl4:amd64 (7.68.0-1ubuntu2.18) over (7.68.0-1ubuntu2.16) ...
Setting up apt-transport-https (2.0.9) ...
Setting up libcurl4:amd64 (7.68.0-1ubuntu2.18) ...
Setting up curl (7.68.0-1ubuntu2.18) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
cloud_user@k8s-control:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
OK
cloud_user@k8s-control:~$ cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
> deb https://apt.kubernetes.io/ kubernetes-xenial main
> EOF
deb https://apt.kubernetes.io/ kubernetes-xenial main
cloud_user@k8s-control:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu focal InRelease
Hit:6 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [8993 B]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [65.7 kB]
Fetched 74.7 kB in 1s (73.3 kB/s)
Reading package lists... Done
cloud_user@k8s-control:~$ sudo apt-get install -y kubelet=1.24.0-00 kubeadm=1.24.0-00 kubectl=1.24.0-00
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

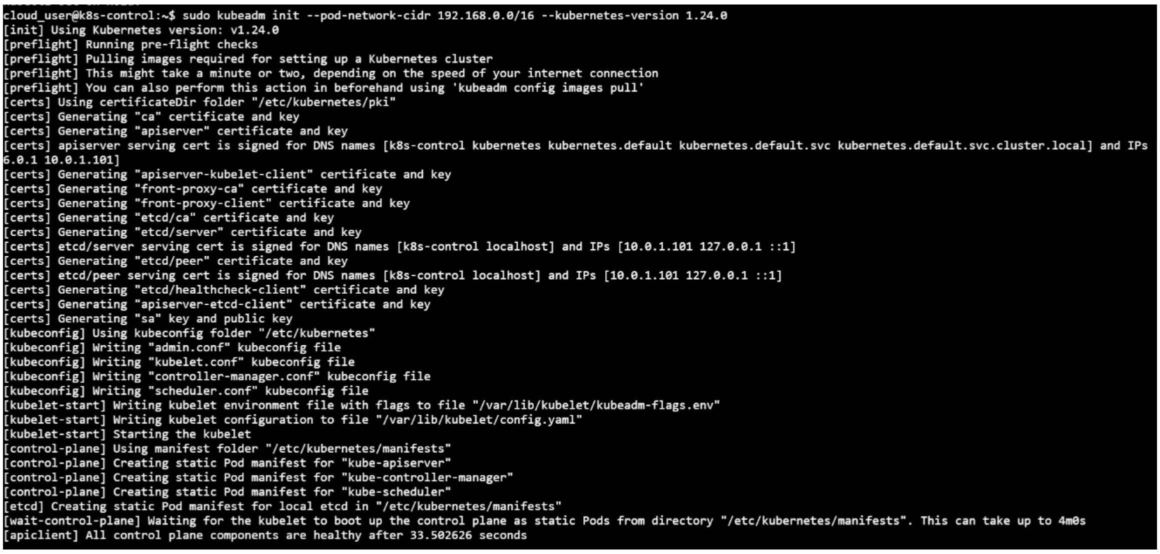


```
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebtables kubernetes-cni socat
Suggested packages:
  nftables
The following NEW packages will be installed:
  conntrack cri-tools ebtables kubeadm kubectl kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 66 not upgraded.
Need to get 84.5 MB of archives.
After this operation, 331 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 conntrack amd64 1:1.4.5-2 [30.3 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ebtables amd64 2.0.11-3build1 [80.3 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 socat amd64 1.7.3.3-2 [323 kB]
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.26.0-00 [18.9 MB]
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubernetes-cni amd64 1.2.0-00 [27.6 MB]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.24.0-00 [19.2 MB]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.24.0-00 [9316 kB]
Get:8 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.24.0-00 [9003 kB]
Fetched 84.5 MB in 4s (23.7 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 125178 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.5-2_amd64.deb ...
Unpacking conntrack (1:1.4.5-2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.26.0-00_amd64.deb ...
Unpacking cri-tools (1.26.0-00) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-3build1_amd64.deb ...
Unpacking ebtables (2.0.11-3build1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../3-kubernetes-cni_1.2.0-00_amd64.deb ...
Unpacking kubernetes-cni (1.2.0-00) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.3.3-2_amd64.deb ...
Unpacking socat (1.7.3.3-2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.24.0-00_amd64.deb ...
Unpacking kubelet (1.24.0-00) ...
```

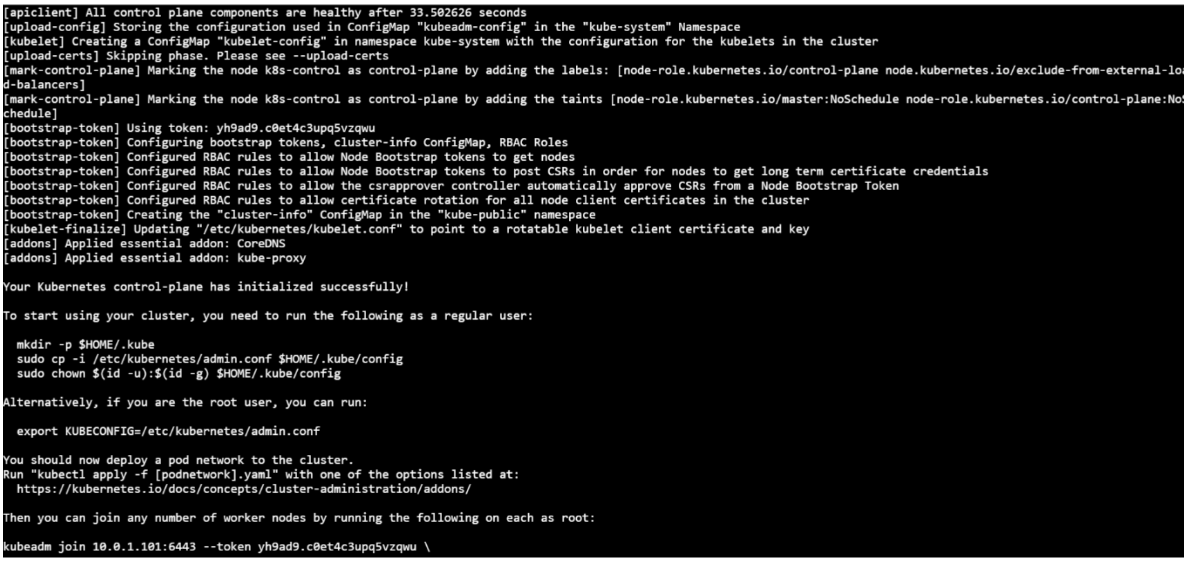


```
Preparing to unpack .../0-conntrack_1%3a1.4.5-2_amd64.deb ...
Unpacking conntrack (1:1.4.5-2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.26.0-00_amd64.deb ...
Unpacking cri-tools (1.26.0-00) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-3build1_amd64.deb ...
Unpacking ebtables (2.0.11-3build1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../3-kubernetes-cni_1.2.0-00_amd64.deb ...
Unpacking kubernetes-cni (1.2.0-00) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.3.3-2_amd64.deb ...
Unpacking socat (1.7.3.3-2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.24.0-00_amd64.deb ...
Unpacking kubelet (1.24.0-00) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.24.0-00_amd64.deb ...
Unpacking kubectl (1.24.0-00) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.24.0-00_amd64.deb ...
Unpacking kubeadm (1.24.0-00) ...
Setting up conntrack (1:1.4.5-2) ...
Setting up kubectl (1.24.0-00) ...
Setting up ebtables (2.0.11-3build1) ...
Setting up socat (1.7.3.3-2) ...
Setting up cri-tools (1.26.0-00) ...
Setting up kubernetes-cni (1.2.0-00) ...
Setting up kubelet (1.24.0-00) ...
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.
Setting up kubeadm (1.24.0-00) ...
Processing triggers for man-db (2.9.1-1) ...
cloud_user@k8s-control:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
cloud_user@k8s-control:~$ █
```

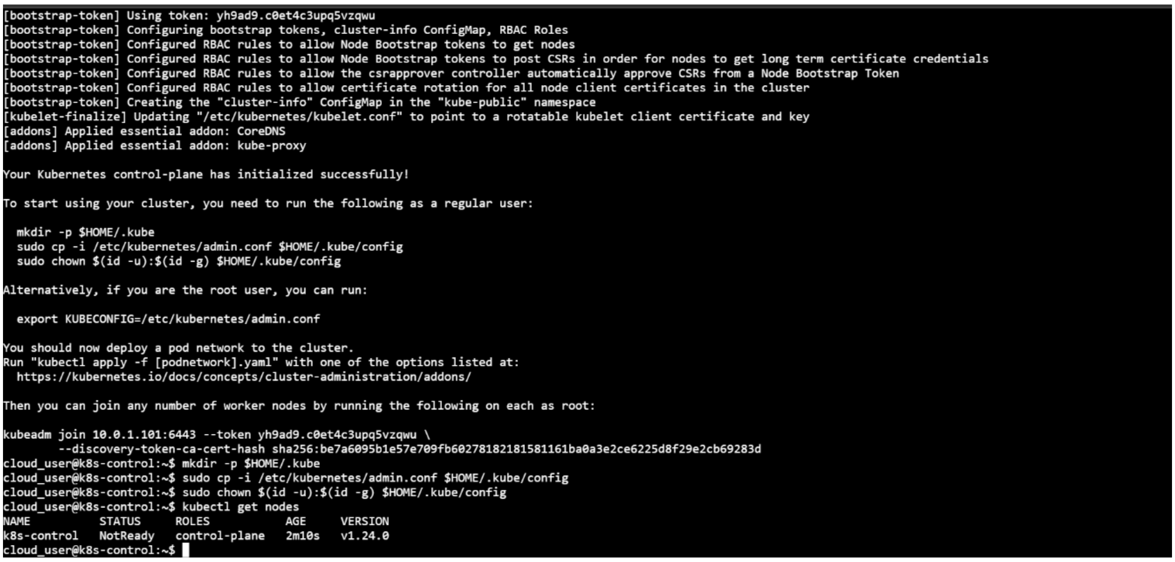
- ## Cluster Initialization



```
cloud_user@k8s-control:~$ sudo kubeadm init --pod-network-cidr 192.168.0.0/16 --kubernetes-version 1.24.0
[init] Using Kubernetes version: v1.24.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [k8s-control kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs
6.0.1 10.0.1.101]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [k8s-control localhost] and IPs [10.0.1.101 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [k8s-control localhost] and IPs [10.0.1.101 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
[apiclient] All control plane components are healthy after 33.502626 seconds
```

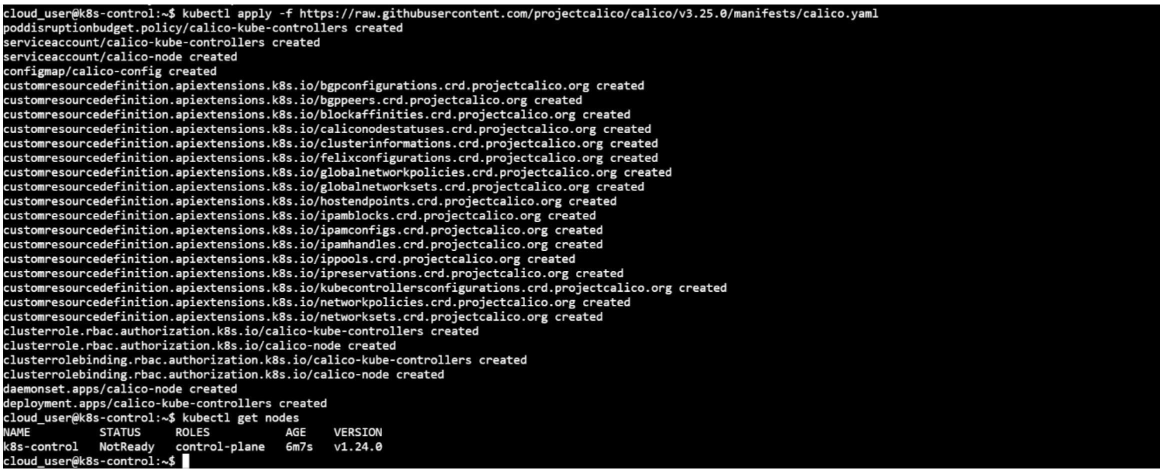


```
[apiclient] All control plane components are healthy after 33.502626 seconds
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node k8s-control as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-loa
d-balancers]
[mark-control-plane] Marking the node k8s-control as control-plane by adding the taints [node-role.kubernetes.io/master:NoSchedule node-role.kubernetes.io/control-plane:NoS
chedule]
[bootstrap-token] Using token: yh9ad9.c0et4c3upq5vzqwu
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 10.0.1.101:6443 --token yh9ad9.c0et4c3upq5vzqwu \
```



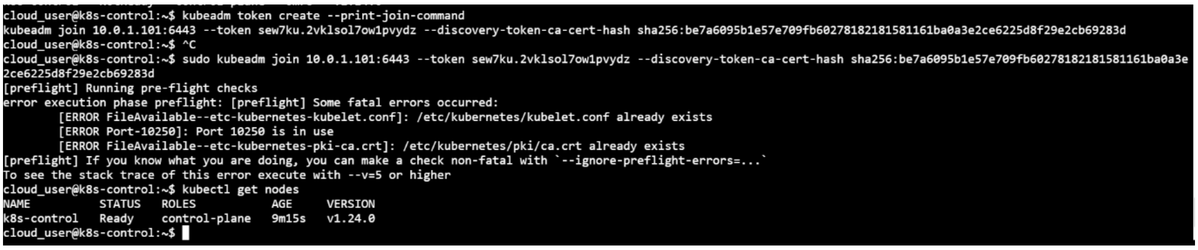
```
[bootstrap-token] Using token: yh9ad9.c0et4c3upq5vzqwu
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 10.0.1.101:6443 --token yh9ad9.c0et4c3upq5vzqwu \
        --discovery-token-ca-cert-hash sha256:be7a6095b1e57e709fb60278182181581161ba0a3e2ce6225d8f29e2cb69283d
cloud_user@k8s-control:~$ mkdir -p $HOME/.kube
cloud_user@k8s-control:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
cloud_user@k8s-control:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
cloud_user@k8s-control:~$ kubectl get nodes
NAME          STATUS     ROLES           AGE     VERSION
k8s-control   NotReady   control-plane   2m10s   v1.24.0
cloud_user@k8s-control:~$
```

- **Install the Calico network add-on.**



```
cloud_user@k8s-control:~$ kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
cloud_user@k8s-control:~$ kubectl get nodes
NAME          STATUS     ROLES           AGE    VERSION
k8s-control   NotReady   control-plane   6m7s   v1.24.0
cloud_user@k8s-control:~$
```

- **Join the Worker nodes to the cluster.**



```
cloud_user@k8s-control:~$ kubeadm token create --print-join-command
kubeadm join 10.0.1.101:6443 --token sew7ku.2vklsol7ow1pvydz --discovery-token-ca-cert-hash sha256:be7a6095b1e57e709fb60278182181581161ba0a3e2ce6225d8f29e2cb69283d
cloud_user@k8s-control:~$ ^C
cloud_user@k8s-control:~$ sudo kubeadm join 10.0.1.101:6443 --token sew7ku.2vklsol7ow1pvydz --discovery-token-ca-cert-hash sha256:be7a6095b1e57e709fb60278182181581161ba0a3e
2ce6225d8f29e2cb69283d
[preflight] Running pre-flight checks
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR FileAvailable--etc-kubernetes-kubelet.conf]: /etc/kubernetes/kubelet.conf already exists
        [ERROR Port-10250]: Port 10250 is in use
        [ERROR FileAvailable--etc-kubernetes-pki-ca.crt]: /etc/kubernetes/pki/ca.crt already exists
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
cloud_user@k8s-control:~$ kubectl get nodes
NAME          STATUS   ROLES           AGE     VERSION
k8s-control   Ready    control-plane   9m15s   v1.24.0
cloud_user@k8s-control:~$ █
```