



Red Hat OpenShift Installation Lab



OCP 4.6 DO322

Red Hat OpenShift Installation Lab

Ausgabe 2 20210712

Veröffentlicht 20210712

Autoren: Michael Jarrett, Alejandro Coma, Chris Caillouet, Benjamin Chardi
Editor: Nicole Muller

Copyright © 2021 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are
Copyright © 2021 Red Hat, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Red Hat, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat, Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed, please send email to training@redhat.com or phone toll-free (USA) +1 (866) 626-2994 or +1 (919) 754-3700.

Red Hat, Red Hat Enterprise Linux, the Red Hat logo, JBoss, OpenShift, Fedora, Hibernate, Ansible, CloudForms, RHCA, RHCE, RHCSA, Ceph, and Gluster are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a registered trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation or the OpenStack community.

All other trademarks are the property of their respective owners.

Mitwirkende: Eliezer Campos, Sajith Eyamkuzhy, David Sacco

Dokumentkonventionen	vii
Einführung	ix
Red Hat OpenShift Installation Lab	ix
Informationen zur Kursumgebung	x
1. Beschreiben des OpenShift-Installationsprozesses	1
Einführung in OpenShift-Installationsmethoden	2
Quiz: Einführung in OpenShift-Installationsmethoden	27
Ausführen des OpenShift-Installationsprogramms	29
Quiz: Ausführen des OpenShift-Installationsprogramms	59
Angeleitete Übung: Erfüllen der OpenShift-Installationsvoraussetzungen	61
Einführung in gehostete OpenShift-Angebote	71
Quiz: Einführung in gehostete OpenShift-Angebote	75
Quiz: Kapitelwiederholung: Beschreiben des OpenShift-Installationsprozesses	77
Zusammenfassung	81
2. Installieren von OpenShift auf einer Cloud-Anbieter-Plattform	83
Einführung in die OpenShift-Installation mit Full-Stack-Automatisierung auf einer Cloud-Anbieter-Plattform	84
Quiz: Einführung in die OpenShift-Installation mit Full-Stack-Automatisierung auf einer Cloud-Anbieter-Plattform	101
Beschreiben der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode	105
Quiz: Beschreiben der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode	120
Demonstrieren der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode	122
Quiz: Demonstrieren der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode	138
Überprüfen der Installation von OpenShift auf AWS	142
Quiz: Überprüfen der Installation von OpenShift auf AWS	156
Quiz: Kapitelwiederholung: Installieren von OpenShift auf einer Cloud-Anbieter-Plattform	158
Zusammenfassung	162
3. Installieren von OpenShift in einer virtualisierten Umgebung	163
Einführung in OpenShift-Installationsmethoden auf Hypervisors	164
Quiz: Einführung in OpenShift-Installationsmethoden auf Hypervisors	170
Beschreiben der Installation von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierungsmethode	172
Quiz: Beschreiben der Installation von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierungsmethode	179
Beschreiben der Installation von OpenShift auf vSphere bei bereits vorhandener Infrastruktur	181
Quiz: Beschreiben der Installation von OpenShift auf vSphere bei bereits vorhandener Infrastruktur	188
Quiz: Kapitelwiederholung: Installieren von OpenShift in einer virtualisierten Umgebung ...	190
Zusammenfassung	194
4. Planen der Installation von OpenShift ohne einen Infrastrukturanbieter	195
Einführung in die OpenShift-Installation ohne Infrastrukturanbieter	196
Quiz: Einführung in die OpenShift-Installation ohne Infrastrukturanbieter	199
Konfigurieren von Netzwerkservices und Hosts für die Installation von OpenShift ohne einen Infrastrukturanbieter	201
Angeleitete Übung: Konfigurieren von Netzwerkservices und Hosts für die Installation von OpenShift ohne einen Infrastrukturanbieter	211

Quiz: Kapitelwiederholung: Planen der Installation von OpenShift ohne einen Infrastrukturanbieter	228
Zusammenfassung	230
5. Installieren von OpenShift ohne Infrastrukturanbieter	231
Durchführen der Installation von OpenShift ohne Infrastrukturanbieter	232
Angeleitete Übung: Durchführen der Installation von OpenShift ohne Infrastrukturanbieter	236
Quiz: Kapitelwiederholung: Installieren von OpenShift ohne einen Infrastrukturanbieter ...	246
Zusammenfassung	250
6. Abschließen der Installation von OpenShift ohne Infrastrukturanbieter	251
Durchführen von Day 1- und Day 2-Vorgängen	252
Angeleitete Übung: Durchführen von Day 1- und Day 2-Vorgängen	254
Ersetzen eines Control Plane-Knotens	268
Angeleitete Übung: Ersetzen eines Control Plane-Knotens	270
Quiz: Kapitelwiederholung: Abschließen der Installation von OpenShift ohne Infrastrukturanbieter	279
Zusammenfassung	281
7. Red Hat OpenShift Installation Lab - Ausführliche Wiederholung	283
Ausführliche Wiederholung	284
Praktische Übung: Installieren eines kompakten OpenShift-Clusters	286

Dokumentkonventionen



Literaturhinweise

„Verweise“ geben an, wo Sie weitere Informationen zu einem Thema in externen Dokumentationen finden können.



Anmerkung

„Hinweise“ sind Tipps, Tastenkombinationen oder alternative Ansätze für die vorliegende Aufgabe. Wenn Sie einen Hinweis ignorieren, hat dies normalerweise keine negativen Konsequenzen. Allerdings können Hinweise helfen, einen Vorgang zu optimieren.



Wichtig

In den Feldern „Wichtig“ werden Details hervorgehoben, die andernfalls leicht übersehen werden könnten: Konfigurationsänderungen, die nur die aktuelle Sitzung betreffen, oder Dienste, die neu gestartet werden müssen, bevor ein Update angewendet werden kann. Wenn Sie ein Feld mit der Bezeichnung „Wichtig“ ignorieren, führt dies nicht zu Datenverlust, kann jedoch Irritationen und Frustration hervorrufen.



Warnung

„Warnungen“ dürfen nicht ignoriert werden. Ignorierte Warnungen führen mit großer Wahrscheinlichkeit zu einem Datenverlust.

Einführung

Red Hat OpenShift Installation Lab

In Red Hat OpenShift Installation Lab (DO322) erlernen Sie grundlegende Fertigkeiten für die Installation eines OpenShift-Clusters in verschiedenen Umgebungen – vom Proof of Concept bis zur Produktion. Die Teilnehmer erfahren zudem, wie sie Anpassungen identifizieren können, die aufgrund der zugrunde liegenden cloudbasierten, virtuellen oder physischen Infrastruktur erforderlich sein könnten.

Lerninhalte

- Installieren von OpenShift in einer cloudbasierten, virtuellen oder physischen Infrastruktur

Zielgruppe

- Cluster-Administratoren, Cluster-Entwickler und Techniker für Standortzuverlässigkeit, die OpenShift-Cluster entwerfen und bereitstellen möchten, um die Leistungs- und Zuverlässigkeitssanforderungen verschiedener Workloads zu erfüllen.

Voraussetzungen

- Zertifizierung als Red Hat Certified Specialist in OpenShift Administration (für OpenShift 4) oder gleichwertige Kenntnisse sind erforderlich.
- Zertifizierung als Red Hat Certified System Administrator (RHCSA) oder gleichwertige Kenntnisse der Red Hat Enterprise Linux-Systemadministration werden ebenfalls dringend empfohlen.

Informationen zur Kursumgebung

Der Rechner „utility“

In diesem Kurs wird **utility** als primäres Computersystem für praktische Übungen verwendet.

Der Rechner **utility** verfügt über ein Standard-Benutzerkonto (**lab**) mit passwortlosem SSH-Zugriff. Um auf den Rechner **utility** zuzugreifen, melden Sie sich zunächst als Benutzer **student** mit dem Passwort **student** beim virtuellen Rechner **workstation** an. Vom Rechner **workstation** aus können Sie als Benutzer **lab** mittels SSH auf den Rechner **utility** zugreifen. Einige Übungen in diesem Kurs können nur vom Benutzer **root** ausgeführt werden. Auf dem Rechner **utility** können Sie den Befehl **sudo -i** verwenden, um zum Benutzer **root** zu werden.

Auf dem Rechner **utility** geben Sie **oc**-Befehle ein, um den OpenShift-Cluster zu verwalten, den Sie im Rahmen der Kursumgebung installieren.

Auf dem Rechner **workstation** führen Sie Shell-Skripte aus, die zum Bearbeiten der Übungen in diesem Kurs notwendig sind.

Wenn für Übungen ein Webbrowser zum Zugreifen auf eine Anwendung oder Website erforderlich ist, müssen Sie die grafische Konsole des Rechners **workstation** verwenden und über diese den Firefox-Webbrowser öffnen.

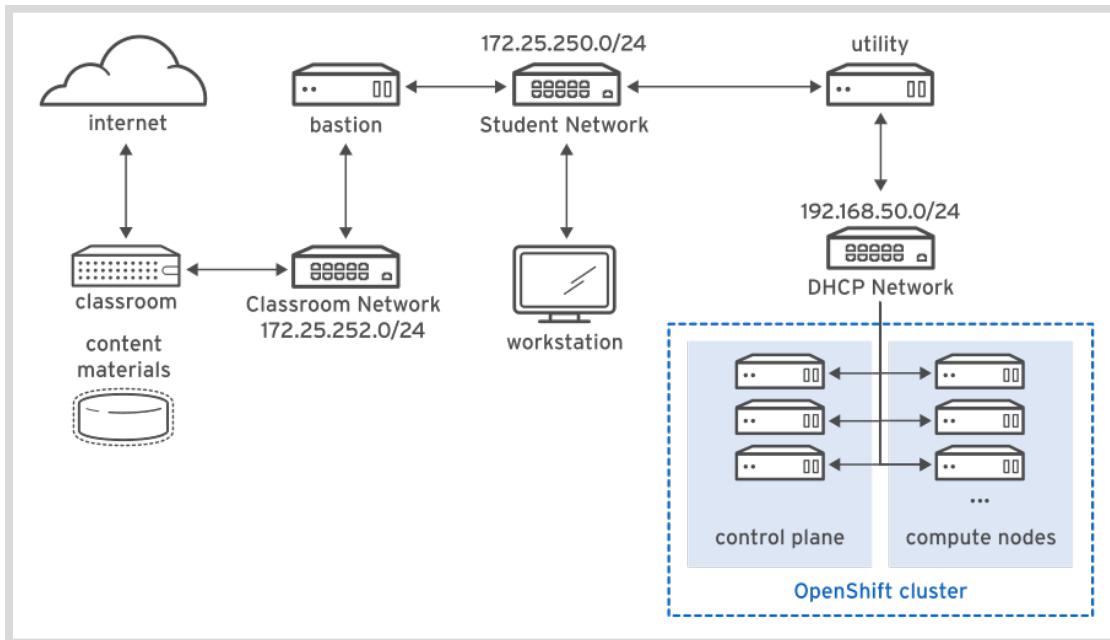
Die Kursumgebung

Jeder Teilnehmer erhält eine komplette Kursumgebung. Im Rahmen dieses Kurses werden alle Teilnehmer gebeten, einen OpenShift-Cluster zu installieren und Administrationsaufgaben auszuführen.

Die Kursumgebung wird vollständig als virtuelle Rechner in einem großen Red Hat OpenStack Platform-Cluster ausgeführt, der von vielen Teilnehmern gemeinsam genutzt wird.

Red Hat Training pflegt viele OpenStack Cluster in unterschiedlichen Rechenzentren weltweit, um Kursteilnehmern aus vielen Ländern eine geringere Latenz zu bieten.

Einführung



Auf allen Rechnern der Netzwerke *Student*, *Classroom* und *Cluster* wird Red Hat Enterprise Linux 8 (RHEL 8) ausgeführt, außer auf den Rechnern, die Knoten des OpenShift-Clusters sind. Diese führen RHEL-CoreOS aus.

Diese Systeme mit den Bezeichnungen **bastion**, **utility** und **classroom** müssen immer ausgeführt werden. Sie stellen Infrastruktur-Services bereit, die von der Kursumgebung und ihrem OpenShift-Cluster benötigt werden.

In der Regel greifen die Lab-Befehle aus den Übungen auf diese Rechner zu, wenn die Umgebung für die Übung eingerichtet werden muss. Sie müssen nichts weiter tun.

Alle Systeme im *Student Network* befinden sich in der DNS-Domain `lab.example.com`, und alle Systeme im *Classroom Network* befinden sich in der DNS-Domain `example.com`.

Die Systeme `masterXX` und `workerXX` sind Knoten des OpenShift 4-Clusters, der Teil der Kursumgebung ist.

Alle Systeme im *Cluster Network* befinden sich in der DNS-Domain `ocp4.example.com`.

Kursraum-Rechner

Rechnername	IP-Adressen	Rolle
<code>workstation.lab.example.com</code>	172.25.250.9	Grafische Workstation für die Systemadministration
<code>classroom.example.com</code>	172.25.254.254	Router, der das Classroom Network mit dem Internet verbindet
<code>bastion.lab.example.com</code>	172.25.250.254	Router, der das Student Network mit dem Classroom Network verbindet

Rechnername	IP-Adressen	Rolle
utility.lab.example.com	172.25.250.253	Router, der das Student Network mit dem Cluster Network und mit dem Storage Server verbindet.
master01.ocp4.example.com	192.168.50.10	Control Plane- und Server-Knoten
master02.ocp4.example.com	192.168.50.11	Control Plane- und Server-Knoten
master03.ocp4.example.com	192.168.50.12	Control Plane- und Server-Knoten
worker01.ocp4.example.com	192.168.50.13	Server-Knoten
worker02.ocp4.example.com	192.168.50.14	Server-Knoten

Abhängigkeiten von Internet-Services

Red Hat OpenShift Container Platform 4 benötigt Zugriff auf zwei Container-Registries, um Container-Images für Operatoren, S2I-Builder und andere Cluster-Services herunterzuladen. Dies sind die folgenden Registries:

- `registry.redhat.io`
- `quay.io`

Wenn eine der Registries bei der Installation von OpenShift nicht verfügbar ist, wird der OpenShift-Cluster möglicherweise nicht gestartet oder kann in einen beeinträchtigten Status wechseln. Um solche Situationen zu verhindern, ist für jede Kursumgebung eine lokale Registry mit einer Spiegelung der OpenShift-Release-Images verfügbar.

Wenn bei diesen Container-Registries ein Ausfall auftritt, während die Kursumgebung ausgeführt wird, können möglicherweise keine Übungen durchgeführt werden, bis der Ausfall behoben ist.

Der dedizierte OpenShift-Cluster

Der Red Hat OpenShift Container Platform 4-Cluster in der Kursumgebung verwendet die Installationsmethode „Bereits vorhandene Infrastruktur“. Alle Knoten werden als Bare-Metal-Server behandelt, obwohl es sich eigentlich um virtuelle Rechner in einem OpenStack-Cluster handelt.

Funktionen zur Integration von OpenShift-Cloud-Anbietern sind nicht aktiviert. Einige Funktionen, die von einer solchen Integration abhängen, wie z. B. MachineSets und die automatische Skalierung von Cluster-Knoten, sind nicht verfügbar.

Wiederherstellen des Zugriffs auf den OpenShift-Cluster

Wenn Sie vermuten, dass Sie sich nicht mehr mit `cluster-admin`-Berechtigungen beim OpenShift-Cluster anmelden können, weil Sie die Cluster-Authentifizierungseinstellungen falsch geändert haben, löschen Sie die Übungsumgebung im Red Hat Learning-Portal, und erstellen Sie eine neue.

Beheben von Fehlern beim Zugriff auf den OpenShift-Cluster

Der Rechner `utility` wird verwendet, um das OpenShift-Installationsprogramm in der Kursumgebung auszuführen. Er ist außerdem eine nützliche Ressource zur Behebung von Cluster-Problemen.

In den angeleiteten Übungen in diesem Kurs werden Sie aufgefordert, einen SSH-Schlüssel für die Fehlerbehebung zu erstellen.

Für reguläre Administrationsaufgaben sollte kein SSH-Zugriff auf die OpenShift-Cluster-Knoten notwendig sein, da OpenShift 4 den Befehl `oc debug` bereitstellt. Falls erforderlich, können Sie mit dem Benutzer `core` und dem für die Fehlerbehebung generierten SSH-Schlüssel auf die OpenShift-Knoten zugreifen, um auf alle Cluster-Knoten zu zugreifen. Beispiel:

```
[lab@utility ~]$ ssh -i ~/.ssh/ocp4upi core@node_ip
```

Ersetzen Sie im vorherigen Beispiel `node_ip` durch die IP-Adresse des gewünschten Cluster-Knotens.

Verwenden von Chromium als Alternative zu Firefox

Auf dem Rechner `workstation` wird RHEL 8.2 ausgeführt. Die neueste Version von Firefox, die in den Repositorys von RHEL 8.2 verfügbar ist, ist Firefox 68. Wenn Sie Probleme beim Rendern von Webseiten haben, ist das EPEL-Repository auf dem Rechner `workstation` aktiviert, sodass Sie den Chromium-Webbrowser installieren können. Führen Sie gegebenenfalls den Befehl `sudo yum install chromium` aus, um Chromium zu installieren.

Kapitel 1

Beschreiben des OpenShift-Installationsprozesses

Ziel

Beschreiben und Vergleichen der Installationsmethoden „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“

Ziele

- Beschreiben und Vergleichen der Installationsmethoden „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“
- Beschreiben des OpenShift-Installationsprogramms und seiner Konfigurationsdateien
- Beschreiben der Unterschiede zwischen einem selbstverwalteten OpenShift-Cluster und gehosteten OpenShift-Angeboten

Abschnitte

- Einführung in OpenShift-Installationsmethoden (und Test)
- Ausführen des OpenShift-Installationsprogramms (und Test, angeleitete Übung)
- Einführung in gehostete OpenShift-Angebote (und Test)
- Kapitelwiederholung: Beschreiben des OpenShift-Installationsprozesses (Test)

Einführung in OpenShift-Installationsmethoden

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die Installationsmethoden „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“ beschreiben und vergleichen können.

Einführung in die OpenShift-Installation

Da der Installationsprozess von Red Hat OpenShift Container Platform 3 ziemlich komplex war, hatten Kunden Schwierigkeiten, ihn reibungslos durchzuführen. Als Reaktion auf dieses Feedback hat Red Hat das Installationsprogramm in OpenShift 4 umgestaltet und dabei nur die Optionen beibehalten, die für eine „eigenwillige“ Erstellung eines funktionsfähigen Clusters erforderlich sind.

Nach Abschluss der Installation können Administratoren die Installation mithilfe von Day-2-Vorgängen anpassen und skalieren.

Der „eigenwillige“ Charakter des OpenShift 4-Installationsprozesses bietet zahlreiche Vorteile.

- Automatisiert die meisten Installationsschritte, um den OpenShift-Installationsprozess zu vereinfachen
- Minimiert menschliche Fehler während der Installation
- Wendet bewährte OpenShift 4-Vorgehensweisen an
- Vereinfacht die Integration mit zukünftigen Automatisierungsmaßnahmen, z. B. OpenShift Assisted Installer, Red Hat Advanced Cluster Management für Kubernetes (ACM) und Cluster-Bereitstellungsintegration in CI/CD-Pipelines.

Verweise auf OpenShift im Verlauf dieses Kurses beziehen sich auf Red Hat OpenShift 4.



Anmerkung

Opinionated Software ("eigenwillige" Software) ist ein Software-Produkt, das davon ausgeht, dass ein bestimmter Geschäftsprozess von Natur aus am besten ist. Die Software wird um diesen Ansatz herum entwickelt. Das Modell der Opinionated Software hilft dem Produktnutzer dabei, die im Produktdesign verwendeten Best Practices zu befolgen.

Beschreiben der Ignition-Konfigurationsdateien

Zum Bereitstellen eines OpenShift Container Platform-Clusters müssen Administratoren die Binärdatei `openshift-install` verwenden. Dieser Befehl wird auch als OpenShift-Installationsprogramm bezeichnet. Die wichtigsten vom OpenShift-Installationsprogramm generierten Ressourcen sind die Ignition-Konfigurationsdateien für den Bootstrap-Knoten, die Control Plane-Knoten und die Server-Knoten. Mit diesen drei Ignition-Konfigurationsdateien und der richtig konfigurierten zugrunde liegenden Infrastruktur können Administratoren die Installation eines OpenShift Container Platform-Clusters starten.

Ignition ist ein Bereitstellungstool zur Anwendung beim ersten Booten, das zur frühzeitigen Konfiguration von RHCOS-Systemen im Bootvorgang entwickelt wurde. Es verwendet die

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

Ignition-Konfigurationsdatei (.ign-Datei im JSON-Format), um den gewünschten Status des RHCOS-Systems zu deklarieren. Der Ignition-Prozess wendet die erforderliche Konfiguration beim ersten Booten des RHCOS-Systems an, um den gewünschten Status auf dem System zu erzwingen. Merkmale des Ignition-Prozesses:

- Er basiert auf dem Linux-Standardstartprozess.
- Er wird auf physischen Knoten, virtuellen Knoten und Cloud-Instanzen ausgeführt.
- Er vereint die Funktionen `kickstart` und `cloud-init` im RHCOS-Systemstart.
- Er wird im Schritt `initramfs` des RHCOS-Bootvorgangs ausgeführt.
- Er konfiguriert Storage, systemd-Units, Zertifikate, Benutzer und benutzerdefinierte Konfigurationen auf RHCOS-Systemen, wenn diese zum ersten Mal gebootet werden.
- Er verwendet die mit dem Befehl `openshift-install` und dem Machine Config-Operator (MCO) generierten Ignition-Konfigurationsdateien.
 - Der Befehl `openshift-install` verwendet die Ignition-Konfigurationsdateien, um den genauen Zustand jedes Knotens bei der Installation zu festlegen.
 - Der Machine Config-Operator wendet nach der Installation andere Änderungen auf die Knoten an, z. B. neue Zertifikate oder SSH-Schlüssel.

Der Ignition-Prozess lädt die Ignition-Konfigurationsdateien von einem der folgenden Speicherorte:

- Lokale Festplatte
- Cloud-Metadaten
- Über das Netzwerk unter Verwendung von http oder https

Sie können das folgende Beispiel einer Ignition-Konfigurationsdatei verwenden, um ein RHCOS-System mit Ignition zu konfigurieren:

```
[user@demo ~]$ sudo yum install jq
...output omitted...
[user@demo ~]$ cat bootstrap.ign | jq .
{
  "ignition": {
    "version": "3.1.0" ①
  },
  "passwd": { ②
    "users": [
      {
        "name": "core",
        "sshAuthorizedKeys": [
          "ssh-rsa AAA...hlw== lab@utility.lab.example.com\n",
          "ssh-rsa AAA...3DR\n"
        ]
      }
    ]
  },
  "storage": { ③
    "files": [
      ...
    ]
  }
}
```

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

```
{
...output omitted...
{
  "overwrite": false,
  "path": "/etc/motd", ④
  "user": {
    "name": "root"
  },
  "append": [
    {
      "source": "data:text/plain;charset=utf-8;base64,VGh...lCg=="
    }
  ],
  "mode": 420
},
...output omitted...
"systemd": { ⑤
  "units": [
    {
      "contents": "[Unit]\nDescription=Bootstrap a Kubernetes cluster\n\nRequires=crio-configure.service\nWants=kubelet.service\nAfter=kubelet.service\n\ncrio-configure.service\nConditionPathExists=!/opt/openshift/.bootkube.done\n\n[Service]\nWorkingDirectory=/opt/openshift\nExecStart=/usr/local/bin/bootkube.sh\n\n\nRestart=on-failure\nRestartSec=5s\n",
      "name": "bootkube.service"
    },
    ...output omitted...
  ]
}
}
```

- ① Ignition-Version
- ② Konfiguriert öffentliche SSH-Schlüssel für den Benutzer „core“.
- ③ Abschnitt „storage“
- ④ Hängt die Base64-codierten Quelldaten an die Datei „/etc/motd“ an.
- ⑤ Erstellt eine systemd-Service-Unit namens `bootkube.service`, die das Skript `/usr/local/bin/bootkube.sh` ausführt.

Zur Behebung von Fehlern im Ignition-Prozess können Sie die folgenden Befehle verwenden.

- Standardmäßig sind die in Ignition-Dateien gespeicherten Daten im Base64-Format codiert. Zur Überprüfung können Sie diese Daten mit dem Befehl `base64 -d` in Klartext decodieren.

```
[user@demo ~]$ echo "VGh...lCg==" | base64 -d
This is the bootstrap node; it will be destroyed when the master is fully up.
The primary services are release-image.service followed by bootkube.service. To
watch their status, run e.g.

journalctl -b -f -u release-image.service -u bootkube.service
```



Anmerkung

Weitere Informationen zu Beispielen für Ignition-Konfigurationsdateien finden Sie hier: <https://coreos.github.io/ignition/examples>

- Nach dem Booten des RHCOS-Systems können Sie die zum Konfigurieren des Systems verwendete Ignition-Konfiguration überprüfen.

```
[root@bootstrap ~]$ cat /boot/ignition/config.ign
>{"ignition": {"version": "3.1.0"}, "passwd": {"users": [{"name": "core", "sshAuthorizedKeys": ["ssh-rsa..."]}]}}
```

- Außerdem können Sie die Ignition-Protokolle überprüfen.

```
[root@bootstrap ~]# journalctl -t ignition
-- Logs begin at Wed 2021-01-27 17:48:36 UTC, end at Mon 2021-02-01 17:52:33 UTC.
--
Jan 27 17:48:38 localhost ignition[684]: Ignition 2.6.0
Jan 27 17:48:38 localhost ignition[684]: Stage: fetch-offline
Jan 27 17:48:38 localhost ignition[684]: reading system config file "/usr/lib/ignition/base.ign"
...output omitted...
Jan 27 17:48:38 localhost ignition[666]: reading system config file "/usr/lib/ignition/user.ign"
...output omitted...
Jan 27 17:48:44 bootstrap ignition[1176]: INFO      : Stage: umount
Jan 27 17:48:44 bootstrap ignition[1176]: INFO      : reading system config file "/usr/lib/ignition/base.ign"
...output omitted...
Jan 27 17:48:44 bootstrap ignition[1176]: INFO      : umount: umount passed
Jan 27 17:48:44 bootstrap ignition[1176]: INFO      : Ignition finished
    successfully
...output omitted...
```

Wie in diesem Kapitel erläutert, müssen Sie für die Installation von OpenShift keine Ignition-Konfigurationsdateien erstellen oder ändern.

- Sie können die Konfigurationsdatei `install-config.yaml` bearbeiten, um die OpenShift-Installation anzupassen.
- Führen Sie dann den Befehl `openshift-installer` aus, um die Kubernetes-Manifeste aus der Konfigurationsdatei `install-config.yaml` zu erstellen.
- Führen Sie abschließend den Befehl `openshift-installer` erneut aus, um die Ignition-Dateien aus den Kubernetes-Manifesten zu erstellen.



Anmerkung

Das Ändern von Ignition-Konfigurationsdateien wird nur unterstützt, wenn Sie dokumentierte Red Hat-Verfahren oder Anweisungen des Red Hat-Supports befolgen. Andernfalls wird dies nicht unterstützt.



Anmerkung

Weitere Informationen finden Sie im Abschnitt *Red Hat Enterprise Linux CoreOS* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/architecture.

Erläutern des OpenShift-Installationsprozesses

Die folgende detaillierte Schrittfolge erläutert den OpenShift-Installationsprozess:

- **Schritt 1:** Der Benutzer führt das OpenShift-Installationsprogramm aus. Das Installationsprogramm fragt die erforderlichen Cluster-Informationen ab und erstellt dann die Installationskonfigurationsdatei `install-config.yaml` entsprechend.
- **Schritt 2:** Das OpenShift-Installationsprogramm erstellt die Kubernetes-Manifeste aus dem Inhalt der Installationskonfigurationsdatei `install-config.yaml`. Die Kubernetes-Manifeste enthalten die erforderlichen Anweisungen zum Erstellen der Ressourcen für die OpenShift-Installation.
- **Schritt 3:** Anhand des Manifestinhalts erstellt der OpenShift-Installationsprozess die Ignition-Konfigurationsdateien für den Bootstrap-Knoten `bootstrap.ign`, die Control Plane-Knoten `master.ign` und die Server-Knoten `worker.ign`.

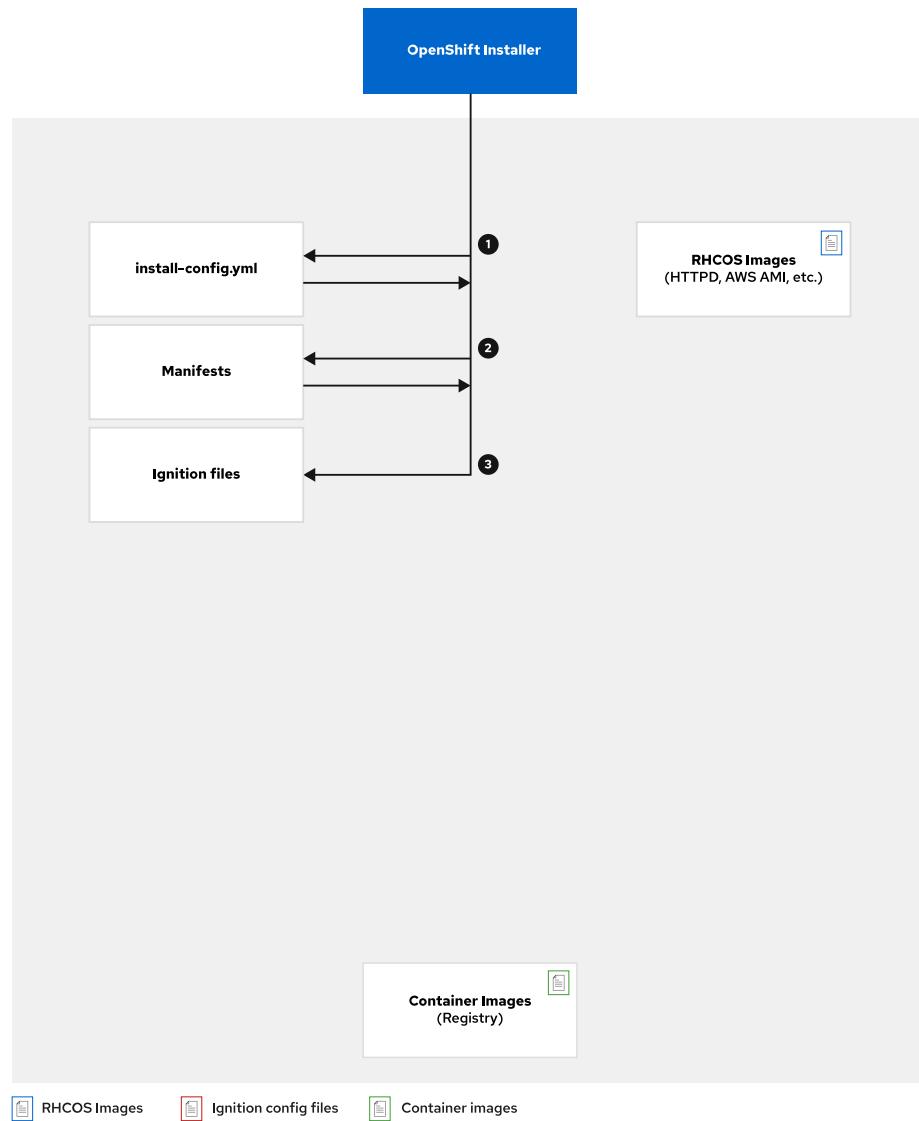
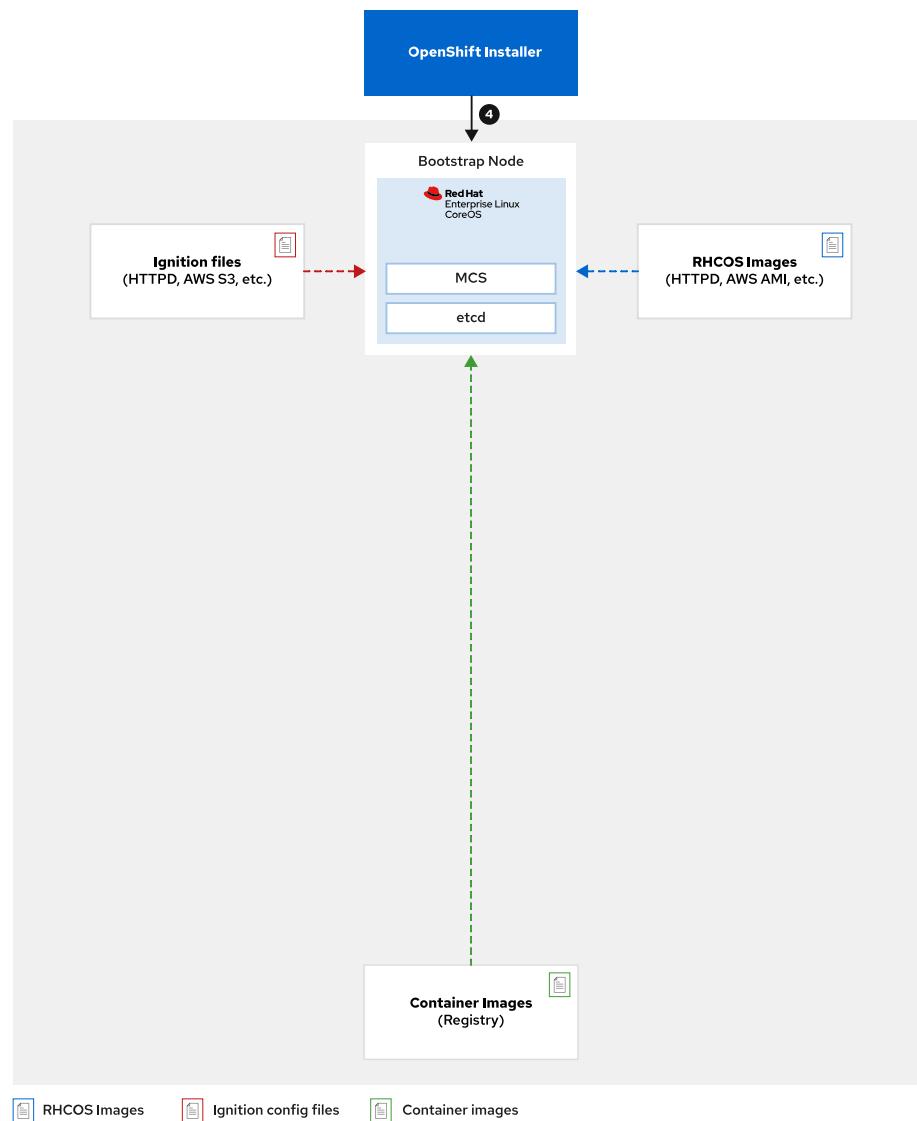


Abbildung 1.1: OpenShift-Installationsprozess – Phase der Ignition-Konfigurationsdateien

- Schritt 4: Der Bootstrap-Knoten bootet, ruft seine Remote-Ressourcen (bootstrap.ign) aus der initialen Ignition-Datenquelle ab und beendet dann den Bootvorgang. In dieser Phase wird die Kubernetes-API auf dem Bootstrap-Knoten ausgeführt.

Der Bootstrap-Knoten hostet die Remote-Ressourcen, die zum Booten von Control Plane-Knoten erforderlich sind (Ignition-Konfigurationsdateien), auf dem Machine Configuration-Server (MCS). Er führt auch eine einzelne Instanz des Clusters etcd aus.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses**Abbildung 1.2: OpenShift-Installationsprozess – Bootstrap-Phase (Bootkube)****Anmerkung**

In diesem Kurs beziehen sich Verweise auf die Kubernetes-API oder die OpenShift-API auf den Kubernetes-API-Server.

Während der OpenShift-Installation wird der Kubernetes API-Server zuerst auf dem Bootstrap-Knoten ausgeführt und dann zu den Control Plane-Knoten verschoben.

- **Schritt 5:** Die Control Plane-Knoten booten, rufen ihre Remote-Ressourcen (die Ignition-Konfigurationsdatei `master.ign`) vom Bootstrap-Knoten ab und beenden den Bootvorgang.
- **Schritt 6:** Der Bootstrap-Knoten startet eine temporäre Control Plane und installiert den Operator `etcd`.

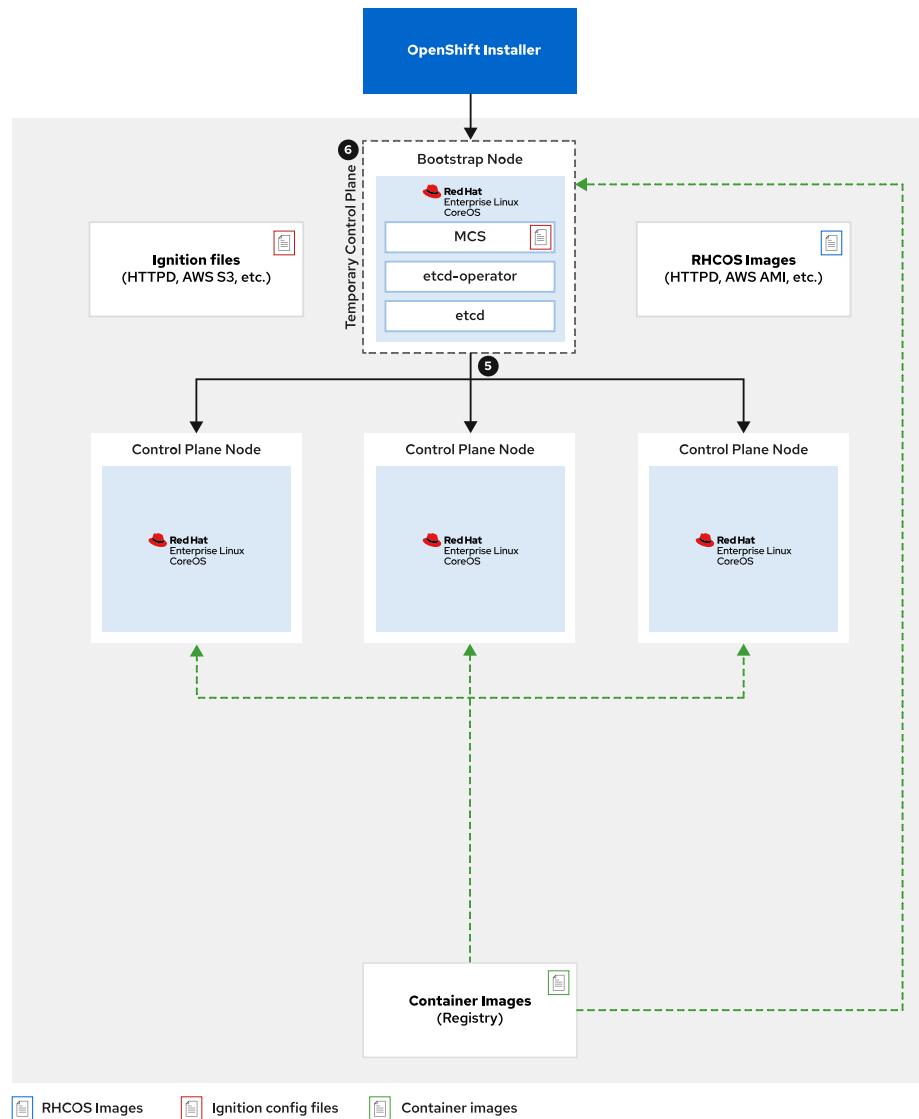


Abbildung 1.3: OpenShift-Installationsprozess – Bootstrap-Phase (temporäre Control Plane)



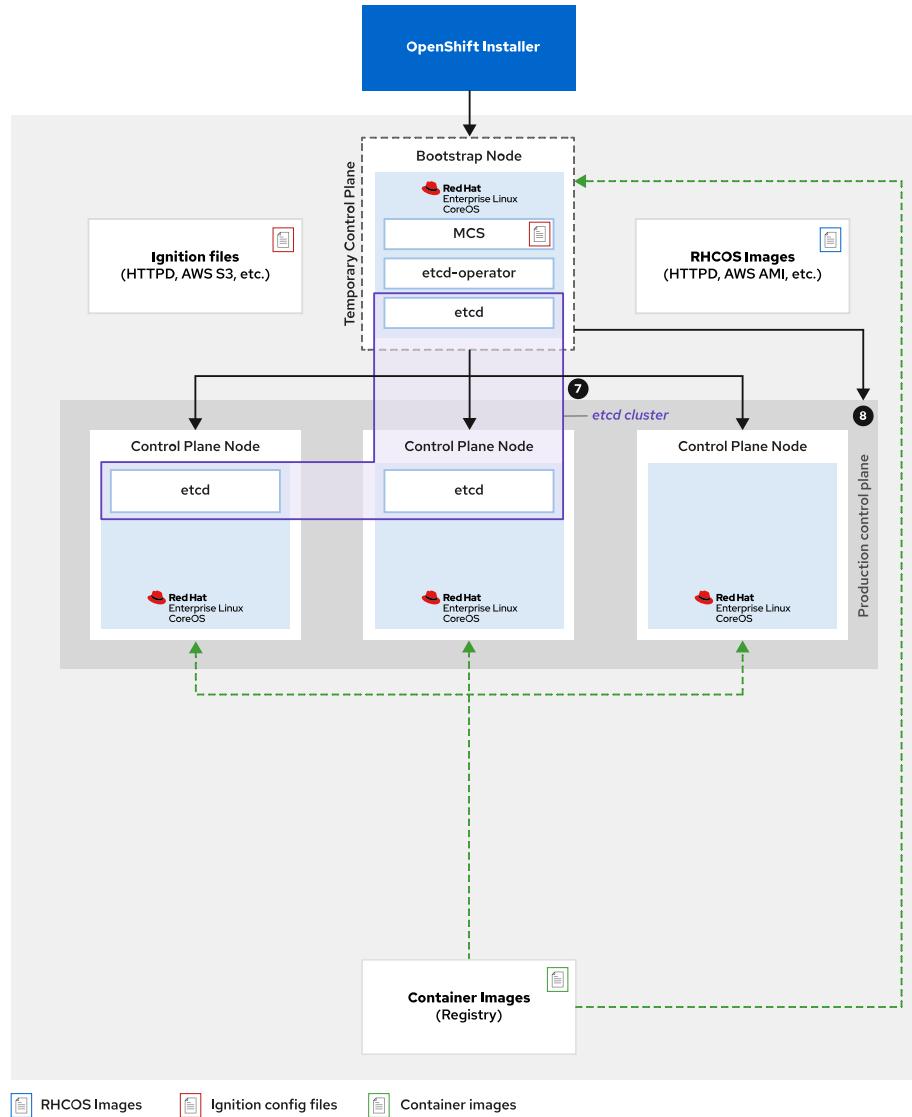
Anmerkung

Während der Installation der Control Plane-Knoten werden die Ignition-Konfigurationsdateien in zwei Phasen abgerufen: **stage-1** und **stage-2**. Zu Beginn der Installation der Control Plane-Knoten (**stage-1**) rufen die Control Plane-Knoten ihre Ignition-Konfigurationsdateien (`master.ign`) aus der initialen Ignition-Datenquelle ab.

Diese Ignition-Konfigurationsdateien enthalten nur eine Umleitungsanweisung, um die entsprechenden Ignition-Dateien vom Kubernetes API-MCS abzurufen. Schließlich rufen die Control Plane-Knoten ihre Ignition-Konfiguration vom Kubernetes API-MCS ab (**stage-2**) und beenden die Installation.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

- **Schritt 7:** Der auf dem Bootstrap-Knoten ausgeführte Operator etcd skaliert den Cluster etcd unter Verwendung von zwei Control Plane-Knoten auf drei Instanzen hoch.
- **Schritt 8:** Die auf dem Bootstrap-Knoten ausgeführte temporäre Control Plane plant die Produktions-Control Plane für die Control Plane-Knoten. Der OpenShift-Installationsprozess überträgt den Cluster etcd auf die Control Plane-Knoten.

**Abbildung 1.4: OpenShift-Installationsprozess – Phase der Produktions-Control Plane-Planung****Anmerkung**

Die temporäre Control Plane wird nur während der OpenShift-Installation verwendet. Der OpenShift-Installationsprozess überträgt die temporäre Control Plane auf die Produktions-Control Plane, die auf den Control Plane-Knoten ausgeführt wird.

Die Produktions-Control Plane ist die endgültige Control Plane, die den OpenShift-Cluster verwaltet.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

- **Schritt 9:** Die temporäre Control Plane fährt herunter und bringt die Produktions-Control Plane hervor. In dieser Phase wird die Kubernetes-API auf der Produktions-Control Plane ausgeführt.
- **Schritt 10:** Bei Full-Stack-Automatisierungsinstallationen fährt das Installationsprogramm den Bootstrap-Knoten herunter. Nach dieser Phase wird der Bootstrap-Knoten nicht mehr benötigt.

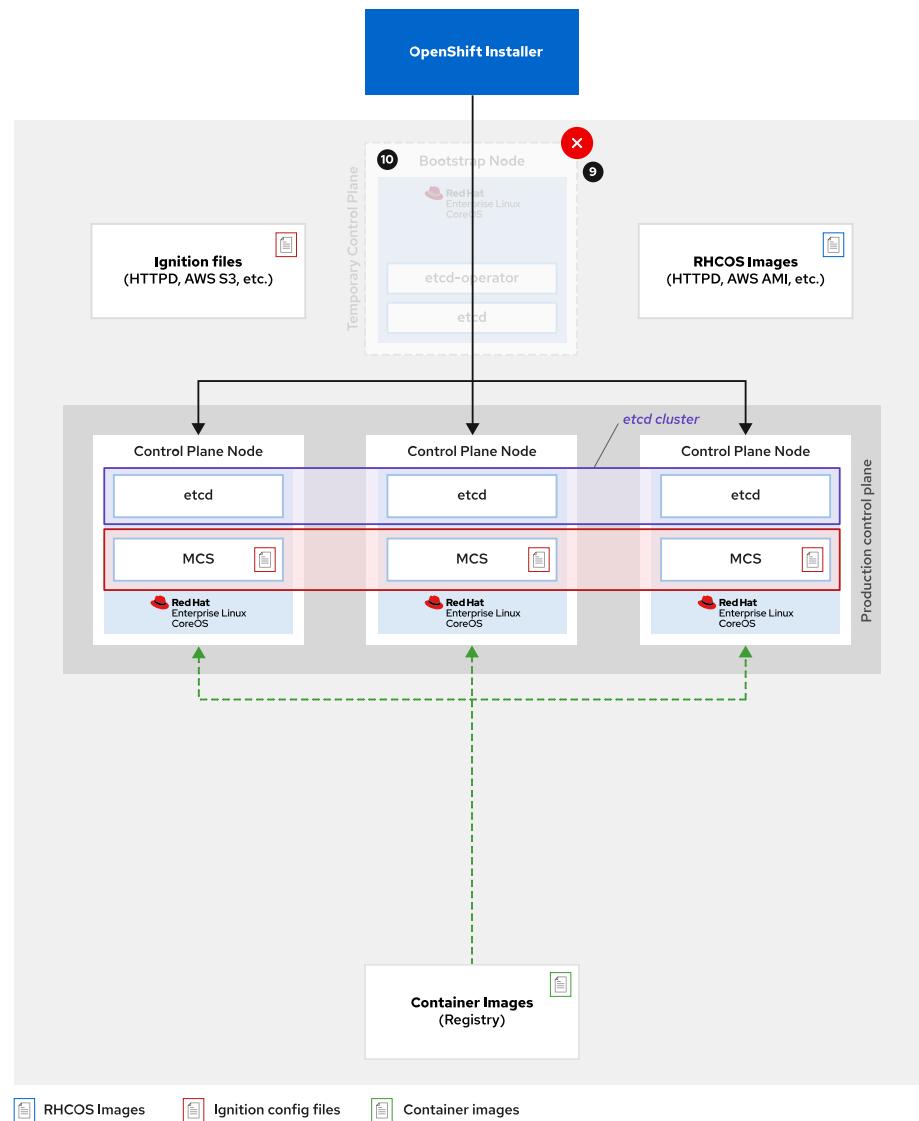


Abbildung 1.5: OpenShift-Installationsprozess – Produktions-Control Plane-Phase



Anmerkung

Der etcd-Cluster führt einen etcd-Pod auf jedem Control Plane-Knoten aus.

Der Kubernetes API-MCS-Service führt einen Machine Config-Server-Pod auf jedem Control Plane-Knoten aus. Der Kubernetes API-MCS-Service hostet die Ignition-Dateien `master.ign` und `worker.ign`.

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

- Schritt 11: In dieser Phase hostet die Produktions-Control Plane die Cluster-Remote-Ressourcen (Ignition-Konfigurationsdateien) für Control Plane-Knoten und Server-Knoten im MCS. Die Server-Knoten booten, rufen ihre Remote-Ressourcen (die Ignition-Konfigurationsdatei `worker.ign`) von den Control Plane-Knoten ab, beenden den Bootvorgang und verbinden sich mit dem Cluster.

Bei Verwendung der Installationsmethode „Bereits vorhandene Infrastruktur“ kann der OpenShift-Installationsprozess die Server-Knoten auch mit dem RHEL 7-Betriebssystem installieren, anstatt das RHCOS-Standardbetriebssystem zu verwenden.

Die Unterstützung für die Verwendung von RHEL 7-Server-Knoten ist veraltet und wird in einer künftigen Version von OpenShift 4 entfernt.

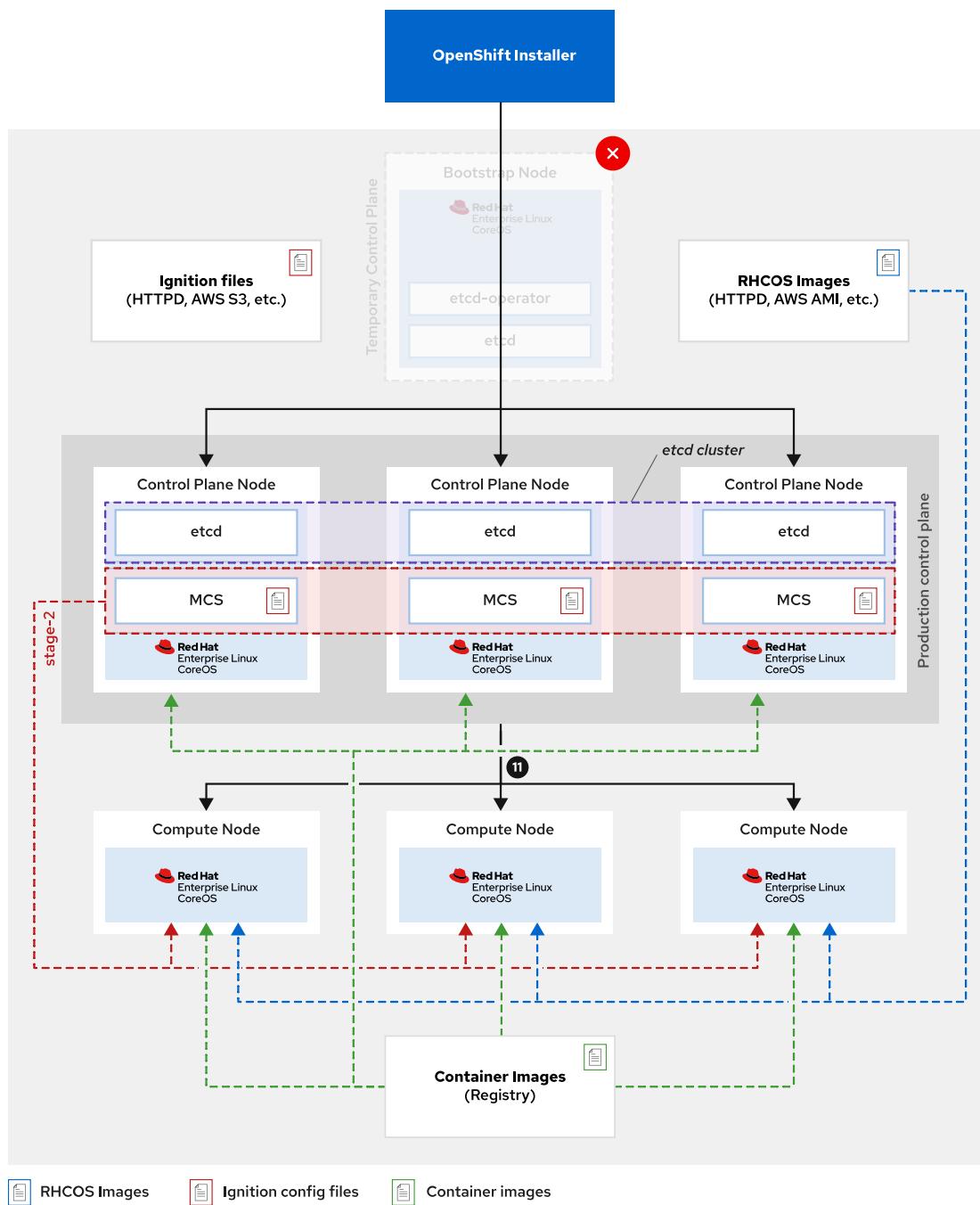


Abbildung 1.6: OpenShift-Installationsprozess – Phase der Server-Knoten-Installation

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

In Bezug auf OpenShift-Installationen gilt Folgendes:

- Red Hat unterstützt nur die Verwendung von drei Control Plane-Knoten.
- Red Hat hat maximal 500 Server-Knoten auf einem Red Hat OpenShift Container Platform 4.6-Cluster getestet.
- Red Hat empfiehlt die Verwendung von mindestens zwei Server-Knoten, um die Hochverfügbarkeit der auf dem Cluster ausgeführten Anwendungen sicherzustellen.
- Weitere Informationen finden Sie im Abschnitt *Planning your environment according to object maximums* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/scalability_and_performance.



Anmerkung

Während der Installation der Server-Knoten werden die Ignition-Konfigurationsdateien in zwei Phasen abgerufen: **stage-1** und **stage-2**. Zu Beginn der Server-Knoteninstallation (**stage-1**) rufen die Server-Knoten ihre Ignition-Konfigurationsdateien (**worker.ign**) aus der initialen Ignition-Datenquelle ab.

Diese Ignition-Konfigurationsdateien enthalten nur eine Umleitungsanweisung, um die entsprechenden Ignition-Dateien vom Kubernetes API-MCS abzurufen. Schließlich rufen die Cluster-Knoten ihre Ignition-Konfiguration vom Kubernetes API-MCS ab (**stage-2**) und beenden die Installation.

Beschreiben der OpenShift-Installationsmethoden

Das OpenShift-Installationsprogramm bietet die folgenden Installationsmethoden:

• Full-Stack-Automatisierung

Dies ist eine „eigenwillige“ Installationsmethode, bei der Administratoren OpenShift mit minimalem manuellem Eingriff installieren.

Das OpenShift-Installationsprogramm stellt den Cluster in einer Infrastruktur bereit, die vom Installationsprogramm bereitgestellt und vom Cluster verwaltet wird.

• Bereits vorhandene Infrastruktur

Bei dieser Installationsmethode sind Administratoren bei der Installation von OpenShift flexibler als bei der Verwendung der Full-Stack-Automatisierungsmethode.

Administratoren verwenden das Installationsprogramm, um einen Cluster in einer Infrastruktur bereitzustellen, die sie selbst vorbereiten und verwalten.

Installationsmethode „Full-Stack-Automatisierung“

Bei Verwendung dieser Installationsmethode gilt:

- Administratoren können das OpenShift-Installationsprogramm verwenden, um einen OpenShift-Cluster in einer Infrastruktur bereitzustellen, die vom OpenShift-Installationsprogramm bereitgestellt und vom OpenShift-Cluster verwaltet wird.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

- Das OpenShift-Installationsprogramm steuert alle Installationsbereiche, einschließlich der Infrastrukturbereitstellung, mit einer „eigenwilligen“ Bereitstellung von OpenShift nach bewährten Vorgehensweisen.
- Diese Installationsmethode wird häufig als „IPI“ (Installer-Provisioned Infrastructure, vom Installationsprogramm bereitgestellte Infrastruktur) bezeichnet.

Das folgende Diagramm beschreibt den Workflow der OpenShift-Installation für die Installationsmethode „Full-Stack-Automatisierung“.

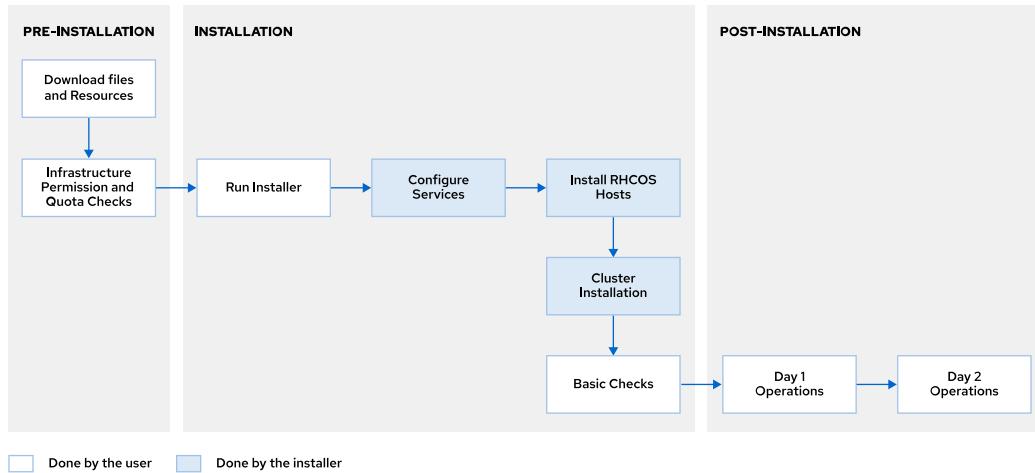


Abbildung 1.7: Workflow der OpenShift-Installation bei Verwendung der Full-Stack-Automatisierungsmethode

Installationsmethode „Bereits vorhandene Infrastruktur“

Bei Verwendung dieser Installationsmethode gilt:

- Administratoren können das OpenShift-Installationsprogramm verwenden, um einen Cluster in einer Infrastruktur bereitzustellen, die sie selbst vorbereiten und verwalten.
- Administratoren sind für das Erstellen und Verwalten ihrer Infrastruktur verantwortlich, was eine umfassendere Anpassung der Infrastruktur und größere betriebliche Flexibilität ermöglicht.
- Diese Installationsmethode wird häufig als „UPI“ (User-Provisioned Infrastructure, vom Benutzer bereitgestellte Infrastruktur) bezeichnet.

Das folgende Diagramm beschreibt den Workflow der OpenShift-Installation für die Installationsmethode „Bereits vorhandene Infrastruktur“.

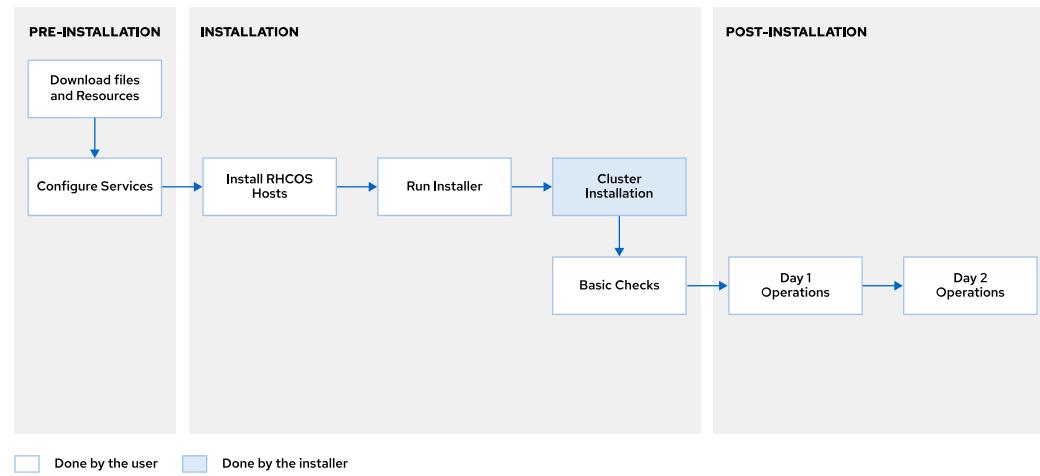


Abbildung 1.8: Workflow der OpenShift-Installation bei Verwendung der Methode „Bereits vorhandene Infrastruktur“



Anmerkung

Der Begriff vom **Installationsprogramm bereitgestellte Cluster** wird häufig im Zusammenhang mit Clustern verwendet, die mit der **Full-Stack-Automatisierungsmethode** installiert werden, und der Begriff vom **Benutzer bereitgestellte Cluster** für Cluster, für die die **Installationsmethode „Bereits vorhandenen Infrastruktur“** verwendet wird.

Vergleichen der Installationsmethoden „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“

Die folgende Tabelle zeigt die grundlegenden Unterschiede zwischen den Installationsmethoden „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“.

Vergleich zwischen Full-Stack-Automatisierung und bereits vorhandener Infrastruktur

Aktion	Full-Stack-Automatisierung	Bereits vorhandene Infrastruktur
Netzwerk aufbauen	Installationsprogramm	Benutzer
Load Balancer einrichten	Installationsprogramm	Benutzer
DNS konfigurieren	Installationsprogramm	Benutzer
Hardware- oder VM-Bereitstellung	Installationsprogramm	Benutzer
Betriebssysteminstallation	Installationsprogramm	Benutzer
Ignition-Konfigurationen generieren	Installationsprogramm	Installationsprogramm

Aktion	Full-Stack-Automatisierung	Bereits vorhandene Infrastruktur
Betriebssystemunterstützung für Control Plane-Knoten	Installationsprogramm: RHCOS	Benutzer: RHCOS
Betriebssystemunterstützung für Server-Knoten	Installationsprogramm: RHCOS (1)	Benutzer: RHCOS + RHEL 7 (1)
Persistenten Storage für die interne Registry konfigurieren	Installationsprogramm (2)	Benutzer
Dynamischen Storage-Anbieter konfigurieren	Installationsprogramm (2)	Benutzer
Knotenbereitstellung und automatische Cluster-Skalierung konfigurieren	Installationsprogramm	Nur für Anbieter mit OpenShift Machine API-Unterstützung (3)

- (1) Die Installationsmethoden „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“ unterstützen das Hinzufügen von RHEL 7-Server-Knoten, die vom Benutzer als Day-2-Vorgang bereitgestellt werden.

Die Unterstützung für die Verwendung von RHEL 7-Server-Knoten ist veraltet und wird in einer künftigen Version von OpenShift 4 entfernt. Nur RHCOS-Cluster-Knoten werden in zukünftigen OpenShift-Versionen unterstützt. Weitere Informationen finden Sie im Abschnitt *Deprecated features* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/release_notes

- (2) Mit Ausnahme der Full-Stack-Automatisierung erfolgen Installationen auf Bare Metal.
- (3) Beachten Sie den Unterschied zwischen einer Installation in bereits vorhandener Infrastruktur mit Cloud-Integration (UPI CI) und einer Installation in bereits vorhandener Infrastruktur auf Bare Metal (UPI BM).
 - Wenn bei UPI-CI-Installationen der Infrastrukturanbieter in die OpenShift Machine API integriert ist, konfiguriert das Installationsprogramm die Knotenbereitstellung und die automatische Cluster-Skalierung.
 - Bei UPI-BM-Installationen, bei denen der Infrastrukturanbieter die Integration in die OpenShift Machine API nicht unterstützt, konfiguriert das Installationsprogramm die Knotenbereitstellung und die automatische Cluster-Skalierung.

Beschreiben der OpenShift-Installationsvoraussetzungen

Vor der Installation von OpenShift müssen Administratoren sicherstellen, dass die für die Installation verwendete Infrastrukturumgebung die erforderlichen Voraussetzungen erfüllt. Die OpenShift-Installationsvoraussetzungen umfassen sowohl allgemeine Voraussetzungen, die für beide Installationsmethoden gelten, als auch für die ausgewählte Installationsmethode spezifische Voraussetzungen.

Allgemeine Voraussetzungen

- Bereitstellen eines Bastion-Hosts

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

- Generieren Sie einen SSH-Schlüssel auf dem Bastion-Host.
- Herunterladen und Installieren von `openshift-install` auf dem Bastion-Host
- Herunterladen und Installieren von `oc` auf dem Bastion-Host
- Rufen Sie das `pull-secret` der Registry ab.
- OpenShift-Cluster-Knoten müssen Zugriff auf einen NTP-Server (Network Time Protocol) haben.
- Stellen Sie sicher, dass die Netzwerk-Firewall der Infrastruktur die OpenShift-Netzwerzugriffsanforderungen erfüllt.

Installationsvoraussetzungen für die Methode „Full-Stack-Automatisierung“

- Überprüfen von Infrastrukturberechtigungen und -kontingenzen

Für Cloud-Umgebungen ist ein Cloud-Benutzerkonto mit den erforderlichen Berechtigungen und Kontingenzen erforderlich.

Installationsvoraussetzungen für die Methode „Bereits vorhandene Infrastruktur“

- Konfigurieren von Netzwerk-Services
- Bereitstellen von Hardware (physisch oder virtuell) für Cluster-Knoten
- Installieren von RHCOS auf Cluster-Knoten

Beschreiben der DNS-Voraussetzungen für OpenShift

Bei der Installation von OpenShift mithilfe der Methode „Bereits vorhandene Infrastruktur“ ist einer der Netzwerk-Services, die Administratoren konfigurieren müssen, der DNS-Service. Administratoren müssen den DNS-Service mithilfe von DNS-Einträgen konfigurieren. Ein vollständiger DNS-Eintrag hat die Form: <component>. <cluster_name>. <base_domain>:

- <cluster_name> ist der Name des Clusters (z. B. `ocp4`).
- <base_domain> ist die Cluster-Basis-Domain, die in der Konfigurationsdatei `install-config.yaml` konfiguriert ist (z. B. `example.com`).
- Die DNS-Reverse-Einträge (PTR) werden ebenfalls benötigt.

In der folgenden Tabelle werden die für die Installation von OpenShift erforderlichen DNS-Einträge beschrieben:

Für die Installation von OpenShift erforderliche DNS-Einträge

Komponente	Eintrag	Beschreibung
Kubernetes-API	api.<cluster_name>.<base_domain>.	DNS-A/AAAA- oder CNAME- und PTR-Einträge zum Identifizieren des API-Load Balancers für die Control Plane-Knoten (auflösbar von beiden Clients außerhalb des Clusters und von allen Knoten im Cluster).
Kubernetes-API	api-int.<cluster_name>.<base_domain>.	DNS-A/AAAA- oder CNAME- und PTR-Einträge zum Identifizieren des API-Load Balancers für die Control Plane-Knoten (auflösbar von allen Knoten im Cluster).
Routen	*.apps.<cluster_name>.<base_domain>.	DNS-A/AAAA- oder CNAME- Platzhaltereintrag zum Identifizieren des Application Ingress-Load Balancers, der für die Cluster-Knoten vorgesehen ist, die den Ingress-Router-Pod ausführen (auflösbar von beiden Clients außerhalb des Clusters und von allen Knoten im Cluster).
Cluster-Knoten	<name>.<cluster_name>.<base_domain>.	DNS-A/AAAA- oder CNAME- und PTR-Einträge zum Identifizieren jedes Cluster-Knotens, einschließlich Bootstrap-Knoten (auflösbar von den Knoten im Cluster).

Die folgende BIND-Konfiguration zeigt die DNS-Konfiguration, die für die Installation eines OpenShift-Clusters mit dem Namen `ocp4` in der Basis-Domain `example.com` verwendet wird:

```
[root@utility ~]# cat /etc/named.conf
...output omitted...
zone "example.com" {
    type master;
    file "example.com.db";
    allow-update { none; };
};

...output omitted...
```

```
[root@utility ~]# cat /var/named/example.com.db
$TTL 1D
@ IN SOA dns.ocp4.example.com. root.example.com. (
    2019022400 ; serial
```

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

```
        3h      ; refresh
        15      ; retry
        1w      ; expire
        3h      ; minimum
    )
IN NS dns.ocp4.example.com.
dns.ocp4      IN A 192.168.50.254
api.ocp4       IN A 192.168.50.254
api-int.ocp4   IN A 192.168.50.254
*.apps.ocp4   IN A 192.168.50.254
bootstrap.ocp4 IN A 192.168.50.9
master01.ocp4  IN A 192.168.50.10
master02.ocp4  IN A 192.168.50.11
master03.ocp4  IN A 192.168.50.12
worker01.ocp4  IN A 192.168.50.13
worker02.ocp4  IN A 192.168.50.14
```

```
[root@utility ~]# cat /etc/named.conf
...output omitted...
zone "50.168.192.in-addr.arpa" IN {
    type master;
    file "example.com.reverse.db";
    allow-update { none; };
};
...output omitted...
```

```
[root@utility ~]# cat /var/named/example.com.reverse.db
$TTL 1D
@    IN SOA dns.ocp4.example.com. root.example.com. (
            2019022400 ; serial
            3h      ; refresh
            15      ; retry
            1w      ; expire
            3h      ; minimum
)
IN NS dns.ocp4.example.com.
254  IN PTR api.ocp4.example.com.
254  IN PTR api-int.ocp4.example.com.
9    IN PTR bootstrap.ocp4.example.com.
10   IN PTR master01.ocp4.example.com.
11   IN PTR master02.ocp4.example.com.
12   IN PTR master03.ocp4.example.com.
13   IN PTR worker01.ocp4.example.com.
14   IN PTR worker02.ocp4.example.com.
```

Sie können den Befehl `dig` verwenden, um die DNS-Servicekonfiguration zu verifizieren, bevor Sie OpenShift installieren.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

```
[lab@utility ~]$ dig @dns.ocp4.example.com api.ocp4.example.com  
...output omitted...  
;; ANSWER SECTION:  
api.ocp4.example.com. 86400 IN A 192.168.50.254  
...output omitted...
```

```
[lab@utility ~]$ dig @dns.ocp4.example.com api-int.ocp4.example.com  
...output omitted...  
;; ANSWER SECTION:  
api-int.ocp4.example.com. 86400 IN A 192.168.50.254  
...output omitted...
```

```
[lab@utility ~]$ dig @dns.ocp4.example.com -x 192.168.50.254  
...output omitted...  
;; ANSWER SECTION:  
254.50.168.192.in-addr.arpa. 86400 IN PTR api.ocp4.example.com.  
254.50.168.192.in-addr.arpa. 86400 IN PTR api-int.ocp4.example.com.  
...output omitted...
```

```
[lab@utility ~]$ dig @dns.ocp4.example.com master01.ocp4.example.com  
...output omitted...  
;; ANSWER SECTION:  
master01.ocp4.example.com. 86400 IN A 192.168.50.10  
...output omitted...
```

```
[lab@utility ~]$ dig @dns.ocp4.example.com -x 192.168.50.10  
...output omitted...  
;; ANSWER SECTION:  
10.50.168.192.in-addr.arpa. 86400 IN PTR master01.ocp4.example.com.  
...output omitted...
```

Wiederholen Sie diesen Vorgang für alle Cluster-Knoten, einschließlich des Bootstrap-Knotens.

**Anmerkung**

Bei der Installation von OpenShift mit der Full-Stack-Automatisierungsmethode für unterstützte Cloud-Anbieter führt das OpenShift-Installationsprogramm standardmäßig automatisch die DNS-Servicekonfiguration durch.

Administratoren müssen den DNS-Service konfigurieren, wenn sie OpenShift mithilfe der Full-Stack-Automatisierungsmethode auf den folgenden Infrastrukturen installieren:

- Red Hat OpenStack Platform-Cloud-Anbieter mit einem externen DNS
- VMware vSphere
- Bare Metal

Beschreiben der Firewall-Voraussetzungen für OpenShift

Wie im Abschnitt *Allgemeine Voraussetzungen* erläutert, müssen Sie sicherstellen, dass die Firewall-Konfiguration Ihres Infrastrukturnetzwerks die OpenShift-Netzwerkzugriffsanforderungen erfüllt, bevor Sie OpenShift installieren. Bei verbundenen Installationen muss der OpenShift-Cluster auf die Remote-Standorte zugreifen können, die er zum Funktionieren benötigt. Weitere Informationen finden Sie im Abschnitt *Configuring your firewall* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing

Darüber hinaus müssen auf den OpenShift-Cluster-Knoten einige Netzwerk-Ports verfügbar sein, auf die im Cluster-Netzwerk zugegriffen werden kann. Weitere Informationen finden Sie im Kapitel *Installing on bare metal* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_bare_metal

Bei Verwendung einer Netzwerk-Firewall müssen Sie die Netzwerkkonnektivität zwischen den Cluster-Knoten sicherstellen. Außerdem muss jeder Cluster-Knoten in der Lage sein, die Hostnamen aller anderen Knoten im Cluster aufzulösen.

Anforderungen für die Konnektivität von allen Cluster-Knoten zu allen Cluster-Knoten

Protokoll	Port	Beschreibung
ICMP	n. a.	Tests zur Erreichbarkeit von Netzwerken
TCP	9000-9999	Services auf Hostebene, darunter „node-exporter“ an den Ports 9100-9101 und „Cluster Version Operator“ an Port 9099.
TCP	10250-10259	Die von Kubernetes reservierten Standard-Ports
TCP	10256	openshift-sdn
UDP	4789	VXLAN und Geneve
UDP	6081	VXLAN und Geneve
UDP	9000-9999	Services auf Hostebene, darunter „node-exporter“ an den Ports 9100-9101
TCP/UDP	30000-32767	Kubernetes-Knoten-Port

Anforderungen für die Konnektivität von allen Cluster-Knoten zu Control Plane-Knoten

Protokoll	Port	Beschreibung
TCP	2379-2380	etcd-Server-, Peer- und Metrik-Ports.
TCP	6443	Kubernetes-API

Stellen Sie sicher, dass bei den vom OpenShift-Cluster verwendeten Load Balancern sowohl die vorderen (Load Balancer) als auch die hinteren (Cluster-Knoten) Netzwerkports zugänglich sind.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

Der API-Load Balancer (API-LB) stellt einen gemeinsamen Endpunkt für die Interaktion mit und die Konfiguration der Plattform bereit. Der Application Ingress-Load Balancer (APP Ingress-LB) stellt einen Ingress-Punkt für den Anwendungsdatenverkehr bereit, der von außerhalb des Clusters einfließt.

API-Load Balancer-Ports

TCP-Port	Back-End-Knoten (Pool-Mitglieder)	Interner Zugriff	Externer Zugriff	Beschreibung
6443	Bootstrap-Knoten (temporär) und Control Plane-Knoten	Ja	Ja	Kubernetes-API-Server
22623	Bootstrap-Knoten (temporär) und Control Plane-Knoten	Ja	Nein	Machine Config-Server

Application Ingress-Load Balancer-Ports

TCP-Port	Back-End-Knoten (Pool-Mitglieder)	Interner Zugriff	Externer Zugriff	Beschreibung
443	Cluster-Knoten, auf denen die Ingress-Router-Pods ausgeführt werden	Ja	Ja	HTTPS-Datenverkehr
80	Cluster-Knoten, auf denen die Ingress-Router-Pods ausgeführt werden	Ja	Ja	HTTP-Datenverkehr

**Anmerkung**

Bei der in diesem Abschnitt beschriebenen Firewall-Konfiguration wird Folgendes vorausgesetzt:

- Sie installieren einen OpenShift-Cluster mit der Installationsmethode „Bereits vorhandene Infrastruktur“.
- Die zur Installation des Clusters verwendete Netzwerkinfrastruktur verfügt über Netzwerkbeschränkungen, die von einer Netzwerk-Firewall verwaltet werden.

Bei der Installation eines OpenShift-Clusters mit der Full-Stack-Automatisierungsmethode in einem unterstützten Cloud-Anbieter gilt Folgendes:

- Standardmäßig erstellt das Installationsprogramm ein neues virtuelles privates Netzwerk (VPC) für den Cluster, in dem alle Netzwerkzugriffsanforderungen erfüllt sind.
- Wenn Sie den Cluster auf einem vorhandenen virtuellen privaten Netzwerk (VPC) installieren möchten, müssen Sie sicherstellen, dass das vorhandene VPC alle Netzwerkzugriffsanforderungen erfüllt.

Beschreiben der OpenShift-Installationsmodi

Je nach externer Konnektivität des Clusters können Administratoren einen der folgenden Installationsmodi verwenden:

- **Verbunden**

Die Cluster-Knoten haben Internetzugriff, um Container-Images aus den Registries `quay.io` und `registry.redhat.io` abzurufen. Der verbundene Installationsmodus wird bei Verwendung der Installationsmethode „Full-Stack-Automatisierung“ oder „Bereits vorhandene Infrastruktur“ unterstützt.

- **Getrennt**

Administratoren verwenden diesen Installationsmodus, wenn eine verbundene Installation nicht möglich ist. Das OpenShift-Installationsprogramm verwendet eine lokale Container-Registry zum Abrufen von Container-Images. Der getrennte Installationsmodus wird nur bei Verwendung der Installationsmethode „Bereits vorhandene Infrastruktur“ unterstützt.

Einführung in getrennte Installationen

Administratoren können einen OpenShift-Cluster in einer Infrastruktur installieren, die sie in einem eingeschränkten Netzwerk bereitstellen. Dieser Installationsmodus wird auch als getrennter Installationsmodus bezeichnet. Für die getrennte Installation gilt Folgendes:

- Sie erfordert das Spiegeln von Images in eine mit der API-Schema2-Spezifikation konforme lokale Container-Registry.
- Sie verwendet die exakte Version von Images, die in der Payload per Digest bereitgestellt werden.
- Sie erfordert Zugriff auf das Internet, um die Container-Images aus den Quell-Registries in die lokale Container-Registry zu spiegeln.
- Die Images, die für die OpenShift-Installation und nachfolgende Produktaktualisierungen erforderlich sind, müssen in die lokale Container-Registry gespiegelt werden.

Bevor Sie mit einer getrennten Installation beginnen, müssen Sie die erforderlichen Images in die lokale Container-Registry spiegeln und die `imageContentSources`-Daten für Ihre OpenShift-Version abrufen. Spiegeln Sie den folgenden Inhalt in die lokale Container-Registry:

- Das OpenShift Container Platform Image-Repository

Dieses Repository stellt die Container-Images zur Verfügung, die während der Installation oder Aktualisierung des OpenShift-Clusters verwendet werden.

- Image-Streams des Cluster-Samples-Operators

Da die meisten Image-Streams im Namespace `openshift` Images aus der Red Hat-Registry `registry.redhat.io` verwenden, müssen Sie diese Images spiegeln und die Image-Streams entsprechend konfigurieren.

- Remote-OperatorHub-Quellen

In getrennten Clustern kann der Operator Lifecycle Manager (OLM) nicht auf die von Red Hat bereitgestellten OperatorHub-Quellen zugreifen, die auf `quay.io` gehostet werden. Administratoren müssen die OperatorHub-Quellen in die lokale Container-Registry spiegeln und OLM so konfigurieren, dass die Operatoren aus den lokalen Quellen anstatt aus den Standard-Remote-Quellen installiert und verwaltet werden.

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

Anweisungen zum Spiegeln des OpenShift Container Platform-Image-Repositorys und der Image-Streams des Cluster-Samples-Operators in die lokale Registry finden Sie im Abschnitt *Creating a mirror registry for installation in a restricted network* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing.

Anweisungen zum Spiegeln der Remote-OperatorHub-Quellen in die lokale Registry finden Sie im Abschnitt *Using Operator Lifecycle Manager on restricted networks* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/operators/index#olm-restricted-networks.

Für OpenShift-Cluster in eingeschränkten Netzwerken gelten die folgenden Einschränkungen:

- Für den `ClusterVersion`-Status wird die Meldung `Unable to retrieve available updates error` angezeigt.
- Standardmäßig sind die Developer Catalog-Inhalte nicht verfügbar, da der Cluster keinen Zugriff auf die erforderlichen Image-Stream-Tags hat.
- Der OpenShift-Telemetrie-Service ist deaktiviert.

Anpassen von getrennten Installationen

Nachdem Sie die lokale Container-Registry gespiegelt haben, starten Sie den OpenShift-Installationsprozess. Sie müssen die Datei `install-config.yaml` ändern, um die getrennte OpenShift-Installation anzupassen.

Im Folgenden finden Sie eine `install-config.yaml`-Beispieldatei, die in einer verbundenen Installation verwendet wird:

```
[user@demo ~]$ cat ${HOME}/ocp4-cluster/install-config.yaml
apiVersion: v1
baseDomain: example.com
#proxy: ①
# httpProxy: http://<username>:<pswd>@<ip>:<port>
# httpsProxy: http://<username>:<pswd>@<ip>:<port>
# noProxy: example.com
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 2
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: ocp4
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
```

```
fips: false
pullSecret: |
  {"auths":...output omitted...} ②
sshKey: |
  ssh-rsa AA...output omitted...
```

- ① Bei Bedarf können Sie in einer verbundenen Installation einen Netzwerk-Proxy verwenden.
- ② Das pull-secret enthält die Anmelde Daten für die Authentifizierung bei quay.io und registry.redhat.io.

Im Folgenden finden Sie eine `install-config.yaml`-Beispieldatei, die in einer getrennten Installation verwendet wird:

```
[user@demo ~]$ cat ${HOME}/ocp4-cluster/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 2
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: ocp4
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
fips: false
pullSecret: |
  {"auths":...output omitted...} ①
sshKey: |
  ssh-rsa AA...output omitted...
additionalTrustBundle: | ②
-----BEGIN CERTIFICATE-----
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
-----END CERTIFICATE-----
imageContentSources: ③
- mirrors:
  - nexus-registry-int.apps.tools.dev.nextcle.com/openshift/ocp4
    source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - nexus-registry-int.apps.tools.dev.nextcle.com/openshift/ocp4
    source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

- ① Das `pull-secret` enthält die Anmeldedaten für die Authentifizierung bei der lokalen Container-Registry `nexus-registry-int.apps.tools.dev.nextcle.com`.
- ② Wenn das von der lokalen Container-Registry verwendete Zertifikat nicht als vertrauenswürdig eingestuft wird, geben Sie das Zertifikatpaket an, das Sie für die lokale Container-Registry verwendet haben.
- ③ Fügen Sie den Abschnitt `imageContentSources` aus der Ausgabe des Befehls hinzu, der zum Spiegeln des Repository verwendet wurde. Durch diese Anpassung wird die Datei `/etc/containers/registries.conf` auf jedem Cluster-Knoten so konfiguriert, dass die Image-Spiegelung aktiviert wird.



Anmerkung

Die Kursumgebung, in der die angeleiteten Übungen für diesen Kurs durchgeführt werden, verfügt über vollständige Internetkonnektivität. In der Kursumgebung sind eine lokale Registry mit einer Spiegelung der OpenShift-Release-Images und andere Ressourcen verfügbar.



Literaturhinweise

- Weitere Informationen finden Sie im Kapitel *Installation and update* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/architecture

► Quiz

Einführung in OpenShift-Installationsmethoden

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche zwei der folgenden Voraussetzungen sind für die Installation von OpenShift im verbundenen Modus erforderlich? (Wählen Sie zwei Antworten aus.)
- a. Abrufen des pull-secret der Registry
 - b. Verwenden einer lokalen Registry
 - c. Herunterladen und Installieren des Befehls oc auf dem Bastion-Host
 - d. Installieren von Red Hat Enterprise Linux 8 auf dem Bootstrap-Knoten
- 2. Welche der folgenden OpenShift-Installationsmethoden wird für die Installation von OpenShift mit minimalem Administratoreingriff empfohlen?
- a. Installationsmethode „Bereits vorhandene Infrastruktur“
 - b. Installationsmethode „Full-Stack-Automatisierung“
- 3. Welche zwei der folgenden Ressourcen werden vom OpenShift-Installationsprogramm erstellt? (Wählen Sie zwei Antworten aus.)
- a. Ignition-Konfigurationsdateien
 - b. Operator-Images
 - c. Kubernetes-Manifeste
 - d. RHCOS-Images
- 4. Welche drei der folgenden OpenShift-Installationsschritte werden bei Verwendung der Full-Stack-Automatisierungsmethode automatisch vom OpenShift-Installationsprogramm durchgeführt? (Wählen Sie drei Antworten aus.)
- a. Einrichten eines Authentifizierungsanbieters
 - b. Installieren von RHCOS auf Cluster-Knoten
 - c. Einrichten eines Load Balancers für API-Datenverkehr
 - d. Installieren von Red Hat Enterprise Linux 8 auf Server-Knoten
 - e. Aufbauen des Netzwerks, in dem der Cluster installiert wird
- 5. Welche der folgenden Aussagen beschreibt am besten die Rolle des Bootstrap-Knotens im OpenShift-Installationsprozess?
- a. Ein temporärer Knoten, auf dem alle Installationsressourcen als RHCOS-Images gespeichert werden
 - b. Ein temporärer Knoten, auf dem eine minimale Kubernetes-Bereitstellung ausgeführt wird, die zur Installation der OpenShift-Produktions-Control Plane verwendet wird
 - c. Ein temporärer Knoten, auf dem die OpenShift-Produktions-Control Plane ausgeführt wird

► Lösung

Einführung in OpenShift-Installationsmethoden

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ **1. Welche zwei der folgenden Voraussetzungen sind für die Installation von OpenShift im verbundenen Modus erforderlich? (Wählen Sie zwei Antworten aus.)**
 - a. Abrufen des `pull-secret` der Registry
 - b. Verwenden einer lokalen Registry
 - c. Herunterladen und Installieren des Befehls `oc` auf dem Bastion-Host
 - d. Installieren von Red Hat Enterprise Linux 8 auf dem Bootstrap-Knoten
- ▶ **2. Welche der folgenden OpenShift-Installationsmethoden wird für die Installation von OpenShift mit minimalem Administratoreingriff empfohlen?**
 - a. Installationsmethode „Bereits vorhandene Infrastruktur“
 - b. Installationsmethode „Full-Stack-Automatisierung“
- ▶ **3. Welche zwei der folgenden Ressourcen werden vom OpenShift-Installationsprogramm erstellt? (Wählen Sie zwei Antworten aus.)**
 - a. Ignition-Konfigurationsdateien
 - b. Operator-Images
 - c. Kubernetes-Manifeste
 - d. RHCOS-Images
- ▶ **4. Welche drei der folgenden OpenShift-Installationsschritte werden bei Verwendung der Full-Stack-Automatisierungsmethode automatisch vom OpenShift-Installationsprogramm durchgeführt? (Wählen Sie drei Antworten aus.)**
 - a. Einrichten eines Authentifizierungsanbieters
 - b. Installieren von RHCOS auf Cluster-Knoten
 - c. Einrichten eines Load Balancers für API-Datenverkehr
 - d. Installieren von Red Hat Enterprise Linux 8 auf Server-Knoten
 - e. Aufbauen des Netzwerks, in dem der Cluster installiert wird
- ▶ **5. Welche der folgenden Aussagen beschreibt am besten die Rolle des Bootstrap-Knotens im OpenShift-Installationsprozess?**
 - a. Ein temporärer Knoten, auf dem alle Installationsressourcen als RHCOS-Images gespeichert werden
 - b. Ein temporärer Knoten, auf dem eine minimale Kubernetes-Bereitstellung ausgeführt wird, die zur Installation der OpenShift-Produktions-Control Plane verwendet wird
 - c. Ein temporärer Knoten, auf dem die OpenShift-Produktions-Control Plane ausgeführt wird

Ausführen des OpenShift-Installationsprogramms

Ziele

Nach Abschluss dieses Abschnitts sollten Sie das OpenShift-Installationsprogramm und seine Konfigurationsdateien beschreiben können.

Verwenden der OpenShift-Installations-Binärdatei

Unabhängig von der Installationsmethode müssen Sie die OpenShift-Installationsprogramm-Binärdatei `openshift-install` verwenden, um eine OpenShift-Installation zu starten. Dieses Installationsprogramm stellt ein einfaches, einheitliches und „eigenwilliges“ Verhalten während des OpenShift-Installationsprozesses sicher.



Anmerkung

Ab OpenShift Container Platform 4.6 verwendet eine Full-Stack-Installation auf Bare Metal die OpenShift-Installationsprogramm-Binärdatei `openshift-baremetal-install`.

Die folgende Tabelle enthält einige nützliche Optionen, die für den Befehl `openshift-install` zur Verfügung stehen:

`openshift-install <befehloptionen>`

Option	Beschreibung
<code>help</code>	Zeigt Hilfeinformationen an.
<code>explain -h</code>	Erläutert die Felder für jede unterstützte <code>InstallConfig</code> -API.
<code>explain installconfig</code>	Erläutert die Ressource <code>installconfig</code> und ihre Felder.
<code>--log-level debug</code>	Aktiviert den Debug-Modus.
<code>--dir</code>	Konfiguriert den Verzeichnispfad zum Speichern der generierten Ressourcen.
<code>create install-config</code>	Erstellt die Installationskonfigurationsdatei <code>install-config.yaml</code> .
<code>create manifests</code>	Erstellt Kubernetes-Manifeste.
<code>create ignition-configs</code>	Erstellt Ignition-Konfigurationsdateien.
<code>create cluster</code>	Stellt den Cluster bereit.

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

Option	Beschreibung
<code>wait-for bootstrap-complete</code>	Wartet, bis die Bootstrap-Installationsphase endet.
<code>wait-for install-complete</code>	Wartet, bis die Cluster-Installation endet.
<code>destroy cluster</code>	Entfernt den Cluster.

Nachfolgend wird die Abfolge der allgemeinen Schritte aufgeführt, die für die Installation von OpenShift ausgeführt werden müssen:

1. Erfüllen der Installationsvoraussetzungen
2. Erstellen des Installationsverzeichnisses
3. Erstellen der Konfigurationsdatei des Installationsprogramms, `install-config.yaml`
4. Generieren der Kubernetes-Manifeste
5. Generieren der Ignition-Konfigurationsdateien
6. Bereitstellen des OpenShift-Clusters
7. Überprüfen der Integrität des OpenShift-Clusters

Erstellen des Installationsverzeichnisses

Um mit der Installation von OpenShift zu beginnen, erstellen Sie zunächst das Installationsverzeichnis. In diesem Verzeichnis werden alle vom OpenShift-Installationsprogramm erstellten Dateien gespeichert.

```
[user@demo ~]$ mkdir ${HOME}/ocp4-cluster
```



Anmerkung

Das OpenShift-Installationsprogramm erstellt das Installationsverzeichnis, falls es noch nicht vorhanden ist.

Erstellen der Konfigurationsdatei des Installationsprogramms

Führen Sie nach dem Erstellen des Installationsverzeichnisses das OpenShift-Installationsprogramm aus, um die Konfigurationsdatei des Installationsprogramms, `install-config.yaml`, zu erstellen. Das Installationsprogramm fordert Sie zur Eingabe der erforderlichen Cluster-Informationen auf und erstellt dann die Datei `install-config.yaml` entsprechend.

```
[user@demo ~]$ openshift-install create install-config \
> --dir=${HOME}/ocp4-cluster
? SSH Public Key /home/user/.ssh/ocp4-cluster.pub
? Platform aws
INFO Credentials loaded from the "default" profile in file "/home/user/.aws/
credentials"
? Region us-east-2
? Base Domain mydomain.com
```

```
? Cluster Name ocp4
? Pull Secret [? for help] +++++
INFO Install-Config created in: /home/user/ocp4-cluster
```



Anmerkung

Dieses Beispiel veranschaulicht die Erstellung der Konfigurationsdatei `install-config.yaml` für eine OpenShift-Installation auf AWS.

Im Folgenden finden Sie eine `install-config.yaml`-Beispielkonfigurationsdatei:

```
[user@demo ~]$ cat ${HOME}/ocp4-cluster/install-config.yaml
apiVersion: v1
baseDomain: mydomain.com
compute:
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform: {}
  replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: ocp4
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  aws:
    region: us-east-2
publish: External
pullSecret: |
  {"auths":...}
sshKey: |
  ssh-rsa AA...
```



Anmerkung

Bevor Sie die Kubernetes-Manifeste erstellen, können Sie die Konfigurationsdatei `install-config.yaml` bearbeiten, um die OpenShift-Installation anzupassen. Die gängigsten Installationsanpassungen werden an anderer Stelle in diesem Kurs erläutert.

Generieren der Kubernetes-Manifeste

Führen Sie nach dem Erstellen der Datei `install-config.yaml` das OpenShift-Installationsprogramm aus, um die Kubernetes-Manifeste zu generieren. Ein Kubernetes-Manifest ist eine Datei, die ein oder mehrere Kubernetes-API-Objekte beschreibt, z. B. Pods, Services, Bereitstellungen oder MachineConfigs. Die Kubernetes-Manifeste enthalten die erforderlichen Anweisungen zum Erstellen der Ressourcen für die OpenShift-Installation.

```
[user@demo ~]$ openshift-install create manifests \
> --dir=${HOME}/ocp4-cluster
INFO Consuming Install Config from target directory
INFO Manifests created in: /home/user/ocp4-cluster/manifests and /home/user/ocp4-
cluster/openshift
```

Sie können die vom OpenShift-Installationsprogramm generierten Kubernetes-Manifeste im Installationsverzeichnis überprüfen.

```
[user@demo ~]$ find ${HOME}/ocp4-cluster/manifests
/home/user/ocp4-cluster/manifests
/home/user/ocp4-cluster/manifests/04-openshift-machine-config-operator.yaml
...output omitted...
```

Im Folgenden finden Sie ein Beispiel eines offiziellen Verfahrens von Red Hat, das erläutert, wie ein neues Kubernetes-Manifest erstellt wird, um während der Installation den Control Plane-Knoten das Kernel-Argument `loglevel=7` hinzuzufügen. Erstellen Sie im Verzeichnis `${HOME}/ocp4-cluster/manifests/openshift` eine Datei mit dem Namen `99-openshift-machineconfig-master-kargs.yaml`, die ein MachineConfig-Objekt zum Hinzufügen der Kernel-Einstellung definiert.

```
[user@demo ~]$ cd ${HOME}/ocp4-cluster/manifests/openshift
```

```
[user@demo ~]$ cat << EOF > 99-openshift-machineconfig-master-kargs.yaml
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: master
  name: 99-openshift-machineconfig-master-kargs
spec:
  kernelArguments:
    - 'loglevel=7'
EOF
```

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

Beim Erstellen der Ignition-Dateien aus den Manifesten wird dieses neue Manifest umschlossen und der Ignition-Datei `master.ign` der Control Plane-Knoten hinzugefügt.



Warnung

Das Ändern von Kubernetes-Manifesten wird nur unterstützt, wenn Sie dokumentierte Red Hat-Verfahren oder Anweisungen des Red Hat-Supports befolgen. Andernfalls wird dies nicht unterstützt.

Generieren der Ignition-Konfigurationsdateien

Führen Sie nach dem Generieren der Kubernetes-Manifeste das OpenShift-Installationsprogramm aus, um die Ignition-Konfigurationsdateien aus dem Manifestinhalt zu erstellen. Das OpenShift-Installationsprogramm erstellt die Ignition-Konfigurationsdateien für den Bootstrap-Knoten, die Control Plane-Knoten und die Server-Knoten. Bei der Bereitstellung des OpenShift-Clusters verwendet das OpenShift-Installationsprogramm diese Ignition-Konfigurationsdateien, um RHCOS auf dem Bootstrap-Knoten, den Control Plane-Knoten und den Server-Knoten zu installieren und zu konfigurieren.

```
[user@demo ~]$ openshift-install create ignition-configs \
> --dir=${HOME}/ocp4-cluster
INFO Consuming Master Machines from target directory
INFO Consuming OpenShift Install (Manifests) from target directory
INFO Consuming Openshift Manifests from target directory
INFO Consuming Worker Machines from target directory
INFO Consuming Common Manifests from target directory
INFO Ignition-Configs created in: /home/user/ocp4-cluster and /home/user/ocp4-
cluster/auth
```

Sie können die vom OpenShift-Installationsprogramm generierten Ignition-Konfigurationsdateien im Installationsverzeichnis überprüfen.

```
[user@demo ~]$ find ${HOME}/ocp4-cluster -name '*.ign' | xargs ls -lrt
-rw-r----- 1 user user 1732 Dec 27 19:35 /home/user/ocp4-cluster/master.ign
-rw-r----- 1 user user 1732 Dec 27 19:35 /home/user/ocp4-cluster/worker.ign
-rw-r----- 1 user user 307594 Dec 27 19:35 /home/user/ocp4-cluster/bootstrap.ign
```



Anmerkung

Ignition-Dateien sind 24 Stunden lang gültig, danach laufen die darin enthaltenen Zertifikate ab. Wenn die Cluster-Installation fehlschlägt, überprüfen Sie, ob die Ignition-Dateien älter als 24 Stunden sind. Wenn ja, erstellen Sie neue Ignition-Dateien.



Warnung

Das Ändern von Ignition-Konfigurationsdateien wird nur unterstützt, wenn Sie dokumentierte Red Hat-Verfahren oder Anweisungen des Red Hat-Supports befolgen. Andernfalls wird dies nicht unterstützt.

Bereitstellen des OpenShift-Clusters

Führen Sie nach dem Generieren der Ignition-Konfigurationsdateien das OpenShift-Installationsprogramm aus, um den OpenShift-Cluster bereitzustellen.

```
[user@demo ~]$ openshift-install create cluster \
> --dir=${HOME}/ocp4-cluster --log-level=debug
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
...output omitted...
INFO Consuming Bootstrap Ignition Config from target directory
INFO Consuming Worker Ignition Config from target directory
INFO Consuming Master Ignition Config from target directory
...output omitted...
DEBUG Apply complete! Resources: 122 added, 0 changed, 0 destroyed.
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
INFO Waiting up to 20m0s for the Kubernetes API at
  https://api.ocp4.example.com:6443...
INFO API v1.19.0+9f84db3 up
INFO Waiting up to 30m0s for bootstrapping to complete...
...output omitted...
DEBUG Bootstrap status: complete
INFO Destroying the bootstrap resources...
...output omitted...
INFO Waiting up to 40m0s for the cluster at https://api.ocp4.example.com:6443 to initialize...
DEBUG Still waiting for the cluster to initialize: Working towards 4.6.4: 82%
  complete
...output omitted...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export KUBECONFIG=/home/user/ocp4-aws-cluster/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-console.apps.ocp4.example.com
INFO Login to the console with user: "kubeadmin", and password: "xxxx"
...output omitted...
INFO Time elapsed: 34m10s
```

Bei einer Installation in einer bereits vorhandenen Infrastruktur können Sie den Befehl `openshift-install create cluster` zum Bereitstellen des Clusters nicht verwenden, da Sie die Cluster-Knoten bereits installiert haben. Die Installation des Bootstrap-Knotens löst die Cluster-Installation aus. Führen Sie daher die folgende Befehlsfolge aus, um die Cluster-Installation zu überwachen:

```
[user@demo ~]$ openshift-install wait-for bootstrap-complete \
> --dir=${HOME}/ocp4-cluster --log-level=debug
```

```
[user@demo ~]$ openshift-install wait-for install-complete \
> --dir=${HOME}/ocp4-cluster --log-level=debug
```



Anmerkung

Mit `openshift-install wait-for`-Befehlen wird die Cluster-Installation nicht ausgelöst. Es wird empfohlen, sie für die Überwachung der Cluster-Installation zu verwenden.

Überwachen von OpenShift-Installationen

Sie können nicht nur die Installationsprotokolle überprüfen, sondern auch mit dem Befehl `oc` den Installationsprozess aktiv überwachen. Sie können den Befehl `oc` verwenden, sobald im Bootstrap die Kubernetes-API auf der temporären Control Plane ausgeführt wird.

Um den Befehl `oc` nutzen zu können, authentifizieren Sie sich mit einer der folgenden Methoden bei der Kubernetes-API mit `cluster-admin`-Berechtigungen:

- Konfigurieren der Umgebungsvariable `KUBECONFIG` für die Verwendung der `kubeconfig`-Datei
- [user@demo ~]\$ `export KUBECONFIG=${HOME}/ocp4-cluster/auth/kubeconfig`
- Verwenden der in der Datei „`kubeadmin-password`“ gespeicherten `kubeadmin`-Anmelddaten



Warnung

Speichern Sie die Dateien „`kubeconfig`“ und „`kubeadmin-password`“ an einem sicheren Ort. Diese Dateien gewähren `cluster-admin`-Berechtigungen im Cluster. Behandeln Sie sie entsprechend.

Verwenden Sie nach der Anmeldung bei der Kubernetes-API als `cluster-admin` die folgende `oc`-Befehlsfolge, um den Installationsprozess zu überwachen:

```
[user@demo ~]$ watch 'oc get clusterversion; oc get clusteroperators; \
> oc get pods --all-namespaces | grep -v -E "Running|Completed"; oc get nodes' \
...output omitted...
```

Mit dem Befehl `watch` können Sie in Echtzeit anzeigen, wie der Cluster Version Operator (CVO) die Cluster-Operatoren installiert, welche Pods noch nicht auf dem Cluster ausgeführt werden und welchen Status die Cluster-Knoten haben. Der Befehl `watch` führt die `oc`-Befehlsfolge automatisch alle zwei Sekunden aus. Dies ist nützlich für die Überwachung des Fortschritts der Cluster-Installation.

```
[user@demo ~]$ watch 'oc get clusterversion; oc get clusteroperators; \
> oc get pods --all-namespaces | grep -v -E "Running|Completed"; oc get nodes' \
NAME      VERSION AVAILABLE PROGRESSING SINCE STATUS
version    False     True      13m   Working towards 4.6.4: 98% complete

NAME                           VERSION AVAILABLE PROGRESSING DEGRADED SINCE
authentication                 4.6.4   False    False    True    7m59s
cloud-credential               True     False    False    False   11m
cluster-autoscaler             4.6.4   True     False    False   6m34s
config-operator                4.6.4   True     False    False   8m4s
console                        4.6.4   False    True     False   60s
...output omitted...
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
openshift-apiserver	apiserver-5c4b794d87	0/2	Pending	0	2m13s
openshift-kube-apiserver	kube-apiserver-ip-10	0/5	Init:0/1	0	61s
<i>...output omitted...</i>					
NAME	STATUS	ROLES	AGE	VERSION	
master01	Ready	master	7m19s	v1.19.0+9f84db3	
master02	Ready	master	7m18s	v1.19.0+9f84db3	
master03	Ready	master	7m19s	v1.19.0+9f84db3	

Der Befehl `oc get events` gibt die OpenShift-Ereignismeldungen in Echtzeit aus, daher ist er hilfreich, um den Installationsfortschritt und Hintergrundereignisse detailliert zu überwachen.

```
[user@demo ~]$ oc get events -A -w
...output omitted...
openshift-infra      OS       Warning  NetworkNotReady
```

Beheben von Fehlern bei OpenShift-Installationen

Der Prozess der OpenShift-Cluster-Bereitstellung hat drei Hauptphasen:

- **Bootstrap-Phase (Bootkube)**

Der OpenShift-Installationsprozess stellt den Bootstrap-Knoten bereit. Als Nächstes lädt der auf dem Bootstrap-Knoten ausgeführte systemd-Service `release-image` die Container-Images herunter, die zum Starten der temporären Control Plane erforderlich sind. Schließlich startet der auf dem Bootstrap-Knoten ausgeführte systemd-Service `bootkube` die temporäre Control Plane.

- **Bootstrap-Phase (temporäre Control Plane)**

Die Kubernetes-API und die temporäre Control Plane werden auf dem Bootstrap-Knoten ausgeführt. Als Nächstes wartet der Bootstrap-Knoten, bis die Control Plane-Knoten booten und einen etcd-Cluster bilden. Schließlich plant der Bootstrap-Knoten die Produktions-Control Plane für die Control Plane-Knoten.

- **Produktions-Control Plane-Phase**

Nachdem die Produktions-Control Plane auf den Control Plane-Knoten ausgeführt wird, schließt der Cluster Version Operator (CVO) die OpenShift-Cluster-Bereitstellung ab.

Um Installationsprobleme zu beheben, müssen Sie je nach Installationsphase unterschiedliche Verfahren verwenden. Sie können die Phase anhand der OpenShift-Installationsprotokolle ermitteln.

Beheben von Fehlern in der Bootstrap-Phase (Bootkube)

In dieser Phase ist die Kubernetes-API noch nicht auf dem Bootstrap-Knoten verfügbar. Zur Fehlerbehebung müssen Sie die Protokolle über SSH vom Bootstrap-Knoten abrufen. Sie können sich beim Bootstrap-Knoten mit SSH über den Bastion-Host anmelden und die Journal- und Container-Protokolle überprüfen. Eine der Voraussetzungen für die Installation von OpenShift ist die Generierung eines SSH-Schlüssels (Cluster-SSH-Schlüssel), um SSH-Zugriff vom Bastion-Host auf die Cluster-Knoten zu ermöglichen.



Warnung

Führen Sie den Befehl `ssh-keygen` in dieser Phase nicht aus, da damit der Cluster-SSH-Schlüssel überschrieben wird. Sie haben mit dem neuen Cluster-SSH-Schlüssel keinen Zugriff auf die Cluster-Knoten. Sie müssen den Befehl `ssh-keygen` ausführen, bevor Sie die OpenShift-Installation ausführen.

```
[user@demo ~]$ ssh-keygen -t rsa -b 4096 -N '' -f ${HOME}/.ssh/ocp4-cluster
```

Bevor Sie die Cluster-Bereitstellung ausführen, müssen Sie den öffentlichen Cluster-SSH-Schlüssel `ocp4-cluster.pub` in die Datei `install-config.yaml` einschließen. Nachdem Sie die Cluster-Knoten installiert haben, gewährt Ihnen der Cluster-SSH-Schlüssel SSH-Zugriff vom Bastion-Host. Zum Beheben von Fehlern mit dem Bootstrap-Knoten führen Sie auf dem Bastion-Host den Befehl `ssh` aus, um eine Verbindung zum Bootstrap-Knoten herzustellen. Im folgenden Beispiel wird davon ausgegangen, dass `192.168.50.9` die IP-Adresse des Bootstrap-Knotens ist. Verwenden Sie den privaten Cluster-SSH-Schlüssel, um eine Verbindung vom Bastion-Host zum Bootstrap-Knoten herzustellen:

```
[user@demo ~]$ ssh -i ${HOME}/.ssh/ocp4-cluster core@192.168.50.9
```

Auf dem Bootstrap-Knoten werden zwei systemd-Services ausgeführt: `release-image` und `bootkube`.

- Der Service `release-image` lädt die während der Installation verwendeten OpenShift-Release-Container-Images herunter.
- Der Service `bootkube` orchestriert die Schritte der Bootstrap-Phase.

Führen Sie nach der Anmeldung beim Bootstrap-Knoten den folgenden Befehl aus, um beide Services zu debuggen:

```
sh-4.2# journalctl -b -f -u release-image.service -u bootkube.service
```

Führen Sie außerdem den Befehl `crictl` auf dem Bootstrap-Knoten aus, um nach fehlgeschlagenen Containern zu suchen und ihre Protokolle auszugeben:

```
sh-4.2# sudo bash  
sh-4.2# crictl ps -a  
sh-4.2# crictl logs <container_id>
```

Zu typischen Fehlern in dieser Phase zählen:

- Der Bootstrap-Knoten kann aufgrund von Authentifizierungs- oder Netzwerkproblemen keine Container-Images von `quay.io` oder aus der lokalen Registry herunterladen.
- Die Cluster-Knoten können aufgrund von DNS-Problemen die IP-Adresse der Kubernetes-API nicht abrufen.

Beheben von Fehlern in der Bootstrap-Phase (temporäre Control Plane)

In dieser Phase werden die Kubernetes-API und die temporäre Control Plane auf dem Bootstrap-Knoten ausgeführt.

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

Zur Fehlerbehebung müssen Sie die Protokolle vom Bootstrap-Knoten abrufen. Sie können SSH verwenden, um Protokolle vom Bootstrap-Knoten abzurufen, wie an anderer Stelle erläutert. Da die Kubernetes-API auf dem Bootstrap-Knoten verfügbar ist, können Sie auch die Befehle `openshift-install` und `oc` verwenden, um die Protokolle von Bootstrap- und Control Plane-Knoten abzurufen.

```
[user@demo ~]$ export KUBECONFIG=${HOME}/ocp4-cluster/auth/kubeconfig
```

```
[user@demo ~]$ openshift-install gather bootstrap \
> --dir=${HOME}/ocp4-cluster
INFO Pulling debug logs from the bootstrap machine
INFO Bootstrap gather logs captured here "/home/user/ocp4-cluster/log-
bundle-20210107135825.tar.gz"
```

```
[user@demo ~]$ cd ocp4-cluster
[user@demo ~]$ tar -xvzf log-bundle-20210107135825.tar.gz
[user@demo ~]$ cd log-bundle-20210107135825
[user@demo ~]$ ls
bootstrap/ control-plane/ failed-units.txt rendered-assets/ resources/ unit-
status/
```

Verwenden Sie zur Fehlerbehebung bei den Bootstrap-systemd-Services die im Verzeichnis `bootstrap/journals`/ gespeicherten Protokolldateien:

```
[user@demo ~]$ ls bootstrap/journals/
approve-csr.log bootkube.log crio-configure.log crio.log ironic.log
kubelet.log master-update-bmh.log release-image.log
```

Verwenden Sie zur Fehlerbehebung in den Bootstrap-Containern die im Verzeichnis `bootstrap/containers`/ gespeicherten Protokolldateien:

```
[user@demo ~]$ ls bootstrap/containers/
...output omitted...
cluster-version-operator-652...f5.inspect
cluster-version-operator-652...f5.log
...output omitted...
```

Verwenden Sie zur Fehlerbehebung bei den Control Plane-Knoten die im Verzeichnis `control-plane`/ gespeicherten Protokolldateien:

```
[user@demo ~]$ find control-plane/
control-plane/
control-plane/10.0.203.117
control-plane/10.0.203.117/unit-status
control-plane/10.0.203.117/journals
control-plane/10.0.203.117/journals/kubelet.log
control-plane/10.0.203.117/journals/crio.log
...output omitted...
```



Anmerkung

Bei Verwendung der Installationsmethode „Bereits vorhandene Infrastruktur“ müssen Sie die IP-Adressen der Bootstrap- und Control Plane-Knoten angeben, wenn Sie den Befehl `openshift-install gather bootstrap` ausführen.

```
[user@demo ~]$ openshift-install gather bootstrap \
> --dir=${HOME}/ocp4-cluster \
> --bootstrap <bootstrap_address> \
> --master <master_1_address> \
> --master <master_2_address> \
> --master <master_3_address>"
```

Sie können auch auf die Kubernetes-API zugreifen, die auf dem Bootstrap-Knoten ausgeführt wird, und den Fortschritt der Cluster-Installation überwachen. Verwenden Sie zur Fehlerbehebung die `oc`-Standardbefehle auf dem Bastion-Host.

```
[user@demo ~]$ export KUBECONFIG=${HOME}/ocp4-cluster/auth/kubeconfig
```

```
[user@demo ~]$ oc get nodes
NAME      STATUS    ROLES     AGE      VERSION
master01   Ready     master    2m35s   v1.19.0+9f84db3
master02   Ready     master    2m34s   v1.19.0+9f84db3
master03   Ready     master    2m35s   v1.19.0+9f84db3
```

```
[user@demo ~]$ oc get clusterversion
NAME      VERSION AVAILABLE PROGRESSING SINCE STATUS
version      False      True          8m32s Unable to apply 4.6.4: an unknown
error has occurred: MultipleErrors
```

```
[user@demo ~]$ oc get clusteroperators
NAME                      VERSION AVAILABLE PROGRESSING DEGRADED SINCE
authentication
cloud-credential           True      False      False      10m
console
csi-snapshot-controller
dns
etcd          4.6.4  Unknown  Unknown  False      1s
...output omitted...
```

Sie können den Installationsprozess detailliert verfolgen, indem Sie die Cluster-Ereignisse beobachten.

```
[user@demo ~]$ oc get events -A -w
...output omitted...
openshift-infra      0s      Warning  NetworkNotReady
```

In dieser Phase können Sie auch den Befehl `oc adm node-logs` verwenden, um die Cluster-Knotenprotokolle zu erfassen.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

```
[user@demo ~]$ oc adm node-logs -u crio master01
...output omitted...
Jan 27 18:25:57.423858 master01 crio[1627]: time="2021-01-27 18:25:57.423599752Z"
  level=info msg="Checking image status: quay.io/openshift-release-dev/ocp-
v4.0-art-dev@sha256:009...fd5" id=39...fe name=/runtime.v1alpha2.ImageService/
ImageStatus
```

```
[user@demo ~]$ oc adm node-logs -u kubelet master01
...output omitted...
Jan 27 18:31:13.904091 master01 hyperkube[1661]: I0127 18:31:13.903991      1661
  prober.go:126] Readiness probe for "etcd-master01_openshift-etcd(63...40):etcd"
  succeeded
```

```
[user@demo ~]$ oc adm node-logs master01
...output omitted...
Jan 27 18:32:47.487425 master01 hyperkube[1661]: I0127 18:32:47.487445      1661
  kubelet.go:1914] SyncLoop (housekeeping)
Jan 27 18:32:47.536556 master01 hyperkube[1661]: I0127 18:32:47.536513      1661
  exec.go:60] Exec probe response: ""
Jan 27 18:32:47.536556 master01 hyperkube[1661]: I0127 18:32:47.536546      1661
  prober.go:126] Readiness probe for "ovs-z57pn_openshift-sdn(9e...2c):openvswitch"
  succeeded
```



Anmerkung

Auf den OpenShift-Cluster-Knoten werden nur sehr wenige lokale Services ausgeführt, da die meisten Systemservices als Container ausgeführt werden. Die wichtigsten Ausnahmen sind die Container-Engine cri-o und das Kubelet, die systemd-Service-Units sind.

Verwenden Sie außerdem den Befehl `oc debug`, um zur Fehlerbehebung eine Debug-Sitzung auf den Control Plane-Knoten zu erstellen.

```
[user@demo ~]$ export KUBECONFIG=${HOME}/ocp4-cluster/auth/kubeconfig
```

```
[user@demo ~]$ oc get nodes
NAME     STATUS    ROLES   AGE     VERSION
master01  Ready     master  2m35s   v1.19.0+9f84db3
master02  Ready     master  2m34s   v1.19.0+9f84db3
master03  Ready     master  2m35s   v1.19.0+9f84db3
```

```
[user@demo ~]$ oc debug node/master01
...output omitted...
sh-4.2#
```

Führen Sie nach der Anmeldung den folgenden Befehl aus, um die Knotenprotokolle zu debuggen:

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

```
sh-4.2# chroot /host  
sh-4.2# journalctl -f
```

Verwenden Sie außerdem den Befehl `crlctl` auf dem Control Plane-Knoten, um nach fehlgeschlagenen Containern zu suchen und ihre Protokolle abzurufen:

```
sh-4.2# sudo bash  
sh-4.2# crictl ps -a  
sh-4.2# crictl logs <container_id>
```



Anmerkung

Nach Abschluss der Installation sollten Sie gemäß Empfehlung von Red Hat nicht mehr mit SSH auf die Cluster-Knoten zugreifen. Wenn Sie während der Installation den Befehl `oc debug` nicht verwenden können, können Sie über SSH vom Bastion-Host aus auf die Control Plane-Knoten zugreifen. Im folgenden Beispiel wird davon ausgegangen, dass `192.168.50.10` die IP-Adresse des Control Plane-Knotens ist. Verwenden Sie den privaten Cluster-SSH-Schlüssel, um eine Verbindung vom Bastion-Host zu diesem Control Plane-Knoten herzustellen.

```
[user@demo ~]$ ssh -i ${HOME}/.ssh/ocp4-cluster core@192.168.50.9
```

Zu typischen Fehlern in dieser Phase zählen:

- Probleme bei der Installation von Control Plane-Knoten
- Probleme mit der DNS-Auflösung

Beheben von Fehlern in der Produktions-Control Plane-Phase

In dieser Phase werden die Kubernetes-API und die Produktions-Control Plane auf den Control Plane-Knoten ausgeführt. Der Cluster Version Operator (CVO), der auf der Produktions-Control Plane ausgeführt wird, installiert alle Operatoren, die den OpenShift-Cluster erstellen, und beendet die Installation. Sie können dieselben Techniken zur Fehlerbehebung anwenden, die im vorherigen Abschnitt erläutert wurden. Zu typischen Fehlern in dieser Phase zählen:

- Probleme bei der Installation der OpenShift-Operatoren

Überprüfen von OpenShift-Installationen

Nachdem die OpenShift-Installation erfolgreich abgeschlossen wurde, müssen Administratoren sicherstellen, dass der installierte Cluster fehlerfrei ist und für Day 2-Aufgaben zum Onboarding von Benutzern und Anwendungen bereit ist.

Integrität des OpenShift-Clusters

Auf dem Bastion-Host können Sie mit dem Befehl `oc` eine einfache Integritätsprüfung durchführen.

- Konfigurieren Sie die Umgebungsvariable `KUBECONFIG` so, dass sie sich mit `cluster-admin`-Berechtigungen bei der Kubernetes-API authentifiziert.

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

```
[user@demo ~]$ export KUBECONFIG=${HOME}/ocp4-cluster/auth/kubeconfig
```

- Überprüfen Sie, ob die Systemuhr aller Cluster-Knoten mit einem NTP-Server (Network Time Protocol) synchronisiert wird.

```
[user@demo ~]$ oc debug node/master01
...output omitted...
sh-4.4# chroot /host
sh-4.4# cat /etc/chrony.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
pool 2.rhel.pool.ntp.org iburst
...output omitted...

sh-4.4# sudo chronyc tracking
Reference ID      : 8AEC8070 (time.gac.edu)
Stratum          : 3
Ref time (UTC)   : Thu Feb 11 13:06:57 2021
System time      : 0.000034756 seconds fast of NTP time
Last offset      : -0.000001187 seconds
RMS offset       : 0.004707427 seconds
Frequency        : 28.194 ppm fast
Residual freq    : -0.000 ppm
Skew              : 0.136 ppm
Root delay       : 0.052070152 seconds
Root dispersion  : 0.018801220 seconds
Update interval  : 64.9 seconds
Leap status       : Normal
```

Der auf dem Cluster-Knoten ausgeführte systemd-Service `chrony` verwendet den NTP-Pool `2.rhel.pool.ntp.org`. Die Systemuhr wird mit dem NTP-Server `time.gac.edu` synchronisiert.

Wiederholen Sie diesen Vorgang auf allen Cluster-Knoten.

- Überprüfen Sie, ob sich alle Cluster-Knoten im Status Ready befinden.

Wenn ein Cluster-Knoten sich nicht im Status Ready befindet, kann er nicht mit der OpenShift-Control Plane kommunizieren und ist für den Cluster nicht verfügbar.

```
[user@demo ~]$ oc get nodes
NAME     STATUS   ROLES     AGE     VERSION
master01  Ready    master    15h    v1.19.0+9f84db3
master02  Ready    master    15h    v1.19.0+9f84db3
master03  Ready    master    15h    v1.19.0+9f84db3
worker01  Ready    worker    15h    v1.19.0+9f84db3
worker02  Ready    worker    15h    v1.19.0+9f84db3
```

- Überprüfen Sie, ob alle Cluster-Knoten Nutzungsmetriken melden.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

```
[user@demo ~]$ oc adm top node
NAME      CPU(cores)   CPU%    MEMORY(bytes)   MEMORY%
master01  677m        19%    4747Mi          31%
master02  391m        11%    3300Mi          22%
master03  519m        14%    4037Mi          27%
worker01  273m        7%     2435Mi          35%
worker02  313m        8%     2906Mi          42%
```

- Stellen Sie sicher, dass keine Anforderungen für die Signierung des Zertifikats (Certificate Signing Requests, CSRs) auf Genehmigung warten.

```
[user@demo ~]$ oc get csr | grep Pending
```

- Überprüfen Sie, ob der OpenShift-Cluster im CVO-Bericht (Cluster Version Operator) als verfügbar und bereit aufgeführt wird.

```
[user@demo ~]$ oc get clusterversion
NAME      VERSION  AVAILABLE  PROGRESSING  SINCE    STATUS
version   4.6.4    True       False        22h      Cluster version is 4.6.4
```

- Überprüfen Sie, ob alle Cluster-Operatoren verfügbar und bereit sind.

Wenn der Cluster fehlerfrei ist, sollten alle Cluster-Operatoren verfügbar sein und nicht weiter ausgeführt werden, es sei denn, die Konfiguration wird noch von mindestens einem Operator weiterhin angewendet.

```
[user@demo ~]$ oc get clusteroperators
NAME                  VERSION AVAILABLE PROGRESSING DEGRADED SINCE
authentication        4.6.4   True     False      False   22h
cloud-credential     4.6.4   True     False      False   22h
cluster-autoscaler   4.6.4   True     False      False   22h
config-operator       4.6.4   True     False      False   22h
console              4.6.4   True     False      False   22h
...output omitted...
```

- Stellen Sie sicher, dass der Cluster keine Pods mit Planungs- oder Ausführungsproblemen enthält.

```
[user@demo ~]$ oc get pods --all-namespaces | grep -v -E 'Running|Completed'
NAMESPACE  NAME    READY  STATUS    RESTARTS  AGE
```

Integrität des OpenShift-etcd-Clusters

- Stellen Sie sicher, dass alle Mitglieder des etcd-Clusters fehlerfrei sind.

```
[user@demo ~]$ oc get pods -n openshift-etcd | grep etcd-master
etcd-master01  3/3 Running  0  22h
etcd-master02  3/3 Running  0  22h
etcd-master03  3/3 Running  0  22h
```

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

```
[user@demo ~]$ oc rsh -n openshift-etcd etcd-master01
sh-4.4# etcdctl endpoint health --cluster
https://192.168.50.10:2379 is healthy: successfully committed proposal:
took=10.8ms
https://192.168.50.12:2379 is healthy: successfully committed proposal:
took=11.8ms
https://192.168.50.11:2379 is healthy: successfully committed proposal:
took=12.1ms
```

Integrität von OpenShift-API und -Konsole

- Überprüfen Sie, ob der DNS-Eintrag der OpenShift-API, `api.ocp4.example.com`, für die Verwendung der externen Load Balancer-IP-Adresse `192.168.50.254` konfiguriert ist.

```
[user@demo ~]$ dig api.ocp4.example.com
...output omitted...
;; QUESTION SECTION:
;api.ocp4.example.com. IN A

;; ANSWER SECTION:
api.ocp4.example.com. 85333 IN A 192.168.50.254

;; Query time: 0 msec
;; SERVER: 172.25.250.254#53(172.25.250.254)
;; WHEN: Thu Jan 28 05:11:46 EST 2021
;; MSG SIZE rcvd: 71
```

- Überprüfen Sie, ob die OpenShift-API verfügbar ist, indem Sie die Kubernetes-Version anfordern.

```
[user@demo ~]$ curl -k https://api.ocp4.example.com:6443/version
...output omitted...
"gitVersion": "v1.19.0+9f84db3",
```

- Prüfen Sie, ob Sie eine Verbindung zur OpenShift-Konsole herstellen können.

```
[user@demo ~]$ curl -kIs \
> https://console-openshift-console.apps.ocp4.example.com
...output omitted...
HTTP/1.1 200 OK
...output omitted...
```

```
[user@demo ~]$ firefox https://console-openshift-console.apps.ocp4.example.com
```

Integrität der OpenShift-Registry

- Stellen Sie sicher, dass die Anzahl der internen Registry-Pods, die auf dem OpenShift-Cluster ausgeführt werden, mit der Bereitstellungskonfiguration übereinstimmt.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

```
[user@demo ~]$ oc -n openshift-image-registry get deployment.apps/image-registry
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
image-registry  2/2     2           2           24h
```

- Wenn mehrere Server-Knoten vorhanden sind, überprüfen Sie, ob jeder Registry-Pod auf einem anderen Server-Knoten ausgeführt wird.

```
[user@demo ~]$ oc -n openshift-image-registry get pods -o wide
...output omitted...
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE
image-registry-69d88984fb-tjpk1 1/1   Running   0          16m  10.128.2.32 worker02
image-registry-59b67d44f6-n7wkq 1/1   Running   0          16m  10.131.2.12 worker01
...output omitted...
```

- Überprüfen Sie auf jedem Cluster-Knoten die Integrität der internen Registry.

```
[user@demo ~]$ oc debug node/master01
...output omitted...
sh-4.4# chroot /host
sh-4.4# curl -kIs \
> https://image-registry.openshift-image-registry.svc:5000/healthz
...output omitted...
HTTP/2 200
...output omitted...
```

- Überprüfen Sie, ob die Bereitstellung der internen Registry persistenten Storage verwendet. Stellen Sie außerdem sicher, dass sich der Image-Registry-Operator im Verwaltungsstatus Managed befindet.

```
[user@demo ~]$ oc get configs.imageregistry.operator.openshift.io cluster -o yaml
...output omitted...
spec:
  managementState: Managed
  ...output omitted...
  storage:
    pvc:
      claim: registry-claim
  ...output omitted...
```

Integrität von OpenShift Ingress

- Überprüfen Sie, ob der DNS-Platzhaltereintrag für Anwendungen, *.apps.ocp4.example.com, für die Verwendung der externen Load Balancer-IP-Adresse 192.168.50.254 konfiguriert ist.

```
[user@demo ~]$ dig test.apps.ocp4.example.com
...output omitted...
;; QUESTION SECTION:
;test.apps.ocp4.example.com. IN A

;; ANSWER SECTION:
```

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

```
test.apps.ocp4.example.com. 86358 IN A 192.168.50.254

;; Query time: 0 msec
;; SERVER: 172.25.250.254#53(172.25.250.254)
;; WHEN: Thu Jan 28 05:11:46 EST 2021
;; MSG SIZE rcvd: 71
```

- Stellen Sie sicher, dass Sie auf eine über eine OpenShift Ingress-Route verfügbar gemachte Anwendung zugreifen können.

```
[user@demo ~]$ oc get routes -A | grep downloads
openshift-console downloads downloads-openshift-console.apps.ocp4.example.com
```

```
[user@demo ~]$ curl -kIs \
> https://downloads-openshift-console.apps.ocp4.example.com
...output omitted...
HTTP/1.0 200 OK
...output omitted...
```

- Stellen Sie sicher, dass die Anzahl der Router-Pods, die auf dem OpenShift-Cluster ausgeführt werden, mit der Bereitstellungskonfiguration übereinstimmt.

```
[user@demo ~]$ oc -n openshift-ingress get deployment.apps/router-default
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
router-default  2/2     2            2           24h
```

- Wenn mehrere Server-Knoten vorhanden sind, überprüfen Sie, ob jeder Router-Pod auf einem anderen Server-Knoten ausgeführt wird.

```
[user@demo ~]$ oc -n openshift-ingress get pods -o wide | grep router
NAME          READY STATUS RESTARTS AGE IP           NODE
router-default-b7567-l2z4d 1/1   Running 1      16h 192.168.50.13 worker01
router-default-b7567-qf8x4 1/1   Running 1      16h 192.168.50.14 worker02
```

Integrität des dynamischen OpenShift-Storage-Anbieters

Bei Installation von OpenShift in einem unterstützten Cloud-Anbieter konfiguriert das Installationsprogramm einen dynamischen Storage-Anbieter. In diesem Fall müssen Sie den Status des dynamischen Storage-Anbieters verifizieren.

Während der OpenShift-Installation auf AWS mit der Full-Stack-Automatisierungsmethode konfiguriert das Installationsprogramm einen dynamischen Storage-Anbieter AWS EBS. Dieser dynamische Storage-Anbieter verwendet den Storage-Provisioner `aws-ebs`.

Der OpenShift-Installationsprozess erstellt eine Storage-Klasse namens `gp2`, die den dynamischen Storage-Anbieter AWS EBS als Back-End verwendet. Die Storage-Klasse `gp2` stellt persistenten Storage dynamisch für die containerisierten Anwendungen bereit, die auf dem OpenShift-Cluster ausgeführt werden. Im OpenShift-Installationsprozess wird die Storage-Klasse `gp2` als standardmäßige Storage-Klasse konfiguriert. Sofern Sie in der PVC-Definition keine andere Storage-Klasse angeben, verwendet jede PVC-Anforderung die Storage-Klasse `gp2`, um das PV dynamisch zu erstellen und zu binden.

- Überprüfen Sie den Status der AWS EBS-Storage-Klasse `gp2`.

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

```
[user@demo ~]$ oc get sc
NAME          PROVISIONER      RECLAIMPOLICY BINDINGMODE      EXPANSION AGE
gp2 (default)  kubernetes.io/aws-ebs  Delete  WaitForFirstConsumer true   32m
gp2-csi       ebs.csi.aws.com    Delete  WaitForFirstConsumer true   32m
```

Die Storage-Klasse „gp2“ verwendet den Volume-Bindungsmodus `WaitForFirstConsumer`. Dieser Volume-Bindungsmodus verzögert die Bindung und Bereitstellung eines `PersistentVolume`, bis ein Pod erstellt wird, der `PersistentVolumeClaim` verwendet. Diese Konfiguration ist für diese Storage-Klasse unveränderlich.

- Überprüfen Sie, ob die Storage-Klasse gp2 wie erwartet funktioniert.

Erstellen Sie eine einfache httpd-Anwendung, die persistenten Storage für das Verzeichnis `DocumentRoot` unter `/var/www/html` verwendet.

```
[user@demo ~]$ oc new-project httpd-persistent
...output omitted...
[user@demo ~]$ cat /tmp/httpd-persistent.yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: httpd-claim
  namespace: httpd-persistent
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
  ---
  apiVersion: v1
  kind: Pod
  metadata:
    name: httpd
    namespace: httpd-persistent
  spec:
    containers:
    - image: registry.redhat.io/rhel8/httpd-24:latest
      name: httpd
      ports:
      - containerPort: 8080
        name: http
        protocol: TCP
      volumeMounts:
      - mountPath: /var/www/html
        name: httpd-claim
    volumes:
    - name: httpd-claim
      persistentVolumeClaim:
        claimName: httpd-claim
```

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

```
[user@demo ~]$ oc create -f /tmp/httpd-persistent.yaml
persistentvolumeclaim/httpd-claim created
pod/httpd created
```

Überprüfen Sie, ob die PVC-Erstellung automatisch die PV-Bereitstellung und -Bindung durch die standardmäßige Storage-Klasse gp2 auslöst.

```
[user@demo ~]$ oc get pvc
NAME      STATUS   VOLUME      CAPACITY   ACCESS MODES  STORAGECLASS   AGE
httpd-claim  Bound    pvc-d965cb1f  3Gi        RWO          gp2           29s
```

```
[user@demo ~]$ oc rsh httpd
sh-4.4$ df -h
Filesystem      Size  Used Avail Use% Mounted on
...output omitted...
/dev/nvme2n1  2.9G  9.0M  2.9G   1% /var/www/html
...output omitted...
```

Testen der OpenShift-Anwendungserstellung und -bereitstellung

- Erstellen Sie eine Testanwendung, und stellen Sie sie bereit, um den Build-Zyklus der OpenShift-Anwendung zu überprüfen.

```
[user@demo ~]$ oc new-project validate  
...output omitted...  
[user@demo ~]$ oc new-app django-psql-example  
...output omitted...
```

```
[user@demo ~]$ oc get pods -n validate  
NAME                      READY   STATUS    RESTARTS   AGE  
django-psql-example-1-build 0/1     Completed  0          11m  
django-psql-example-1-deploy 0/1     Completed  0          10m  
django-psql-example-1-vfb5l  1/1     Running   0          10m  
postgresql-1-bdgkk          1/1     Running   0          11m  
postgresql-1-deploy         0/1     Completed  0          11m
```

```
[user@demo ~]$ oc logs -f django-psql-example-1-build  
...output omitted...  
Successfully pushed image-registry.openshift-image-registry.svc:5000/validate/  
django-psql-example@sha256:b97b...ff82  
Push successful
```

```
[user@demo ~]$ oc logs -f django-psql-example-1-deploy  
...output omitted...  
--> Scaling django-psql-example-1 to 1  
--> Success
```

```
[user@demo ~]$ oc get routes -n validate  
NAME           HOST/PORT          PATH  
SERVICES      PORT  TERMINATION WILDCARD  
django-psql-example  django-psql-example-validate.apps.ocp4.example.com  
django-psql-example <all>        None
```

```
[user@demo ~]$ curl -Is \  
> django-psql-example-validate.apps.ocp4.example.com  
...output omitted...  
HTTP/1.1 200 OK  
...output omitted...
```

```
[user@demo ~]$ firefox  
http://django-psql-example-validate.apps.ocp4-aws.example.com
```

OpenShift-Cluster-Netzwerk

- Überprüfen Sie die Konfiguration des OpenShift-Cluster-Netzwerks.

```
[user@demo ~]$ oc get network.config/cluster -o yaml
apiVersion: config.openshift.io/v1
kind: Network
metadata:
...output omitted...
status:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
    clusterNetworkMTU: 8142
    networkType: OpenShiftSDN
    serviceNetwork:
      - 172.30.0.0/16
```

Storage-Leistung von OpenShift-etcd

- Überprüfen Sie die Storage-Leistung des etcd-Clusters.

```
[user@demo ~]$ oc get pods -n openshift-etcd | grep etcd-master
etcd-master01 3/3 Running 0 22h
etcd-master02 3/3 Running 0 22h
etcd-master03 3/3 Running 0 22h

[user@demo ~]$ oc rsh -n openshift-etcd etcd-master01
...output omitted...
sh-4.4# etcdctl check perf --load="s"
60 / 60 Boooooooooooooooooooooooo! 100.00% 1m0s
PASS: Throughput is 150 writes/s
PASS: Slowest request took 0.220329s
PASS: Stddev is 0.018010s
PASS

sh-4.4# etcdctl check perf --load="m"
60 / 60 Boooooooooooooooo! 100.00% 1m0s
PASS: Throughput is 964 writes/s
PASS: Slowest request took 0.379547s
PASS: Stddev is 0.022218s
PASS

sh-4.4# etcdctl check perf --load="l"
60 / 60 Boooooooooooooooo! 100.00% 1m0s
FAIL: Throughput too low: 4586 writes/s
PASS: Slowest request took 0.258474s
PASS: Stddev is 0.032695s
FAIL
```

Gemäß dem Testergebnis funktioniert der etcd-Cluster gut für einen mittelgroßen Cluster (–load="m") und schlägt bei einem großen Cluster (–load="l") fehl. Weitere Informationen

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

zur etcd-Leistung finden Sie auf der Dokumentationsseite zu etcd unter <https://etcd.io/docs/current/op-guide/hardware/>

Verwenden Sie das Tool **fio** aus dem Container **etcd-perf**, um einen Leistungstest auf den Control Plane-Knoten durchzuführen und so detailliertere Informationen zur etcd-Storage-Leistung zu erhalten. Die Ausgabe des Leistungstests gibt an, ob die Festplatte schnell genug ist, um etcd zu hosten. Dabei wird das 99. Perzentil der aus der Ausführung erfassten fsync-Metrik verglichen, um festzustellen, ob es kleiner als **10 ms** ist.

```
[user@demo ~]$ oc debug node/master01
...output omitted...
sh-4.4# chroot /host
sh-4.4# podman run --volume /var/lib/etcd:/var/lib/etcd:Z
quay.io/openshift-scale/etcd-perf
...output omitted...
{
    "fio version" : "fio-3.7",
...output omitted...
    "global options" : {
        "rw" : "write",
        "ioengine" : "sync",
        "fdatasync" : "1",
        "directory" : "/var/lib/etcd",
        "size" : "22m",
        "bs" : "2300"
    },
...output omitted...
    "write" : {
        "iops" : 328.808892,
...output omitted..
    },
...output omitted...
}
]
}

-----
99th percentile of fsync is 6193152 ns
99th percentile of the fsync is within the recommended threshold - 10 ms, the disk can be used to host etcd
```

Der **fio**-Leistungstest generiert das folgende Ergebnis:

1. Dieser Test schreibt 22 MiB Daten in Blöcken von 2.300 Byte in das Verzeichnis `/var/lib/etcd`.
2. Das 99. Perzentil von fsync ist 6.193.152 ns, was einer Schreiblatenz von **6 ms** entspricht.
3. Das Betriebssystem hat während des Tests durchschnittlich **328 IOPS** erreicht.

OpenShift Machine API

Bei der Installation von OpenShift in einem unterstützten Cloud-Anbieter konfiguriert das Installationsprogramm die automatische Skalierung des Server-Knotens mithilfe der OpenShift Machine API-Komponente. In diesem Fall müssen Sie den Status der automatischen Skalierung des Server-Knotens überprüfen. Einer der wesentlichen Vorteile einer Installation unter

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

Verwendung der Full-Stack-Automatisierungsmethode auf AWS besteht darin, dass der OpenShift-Installationsprozess die automatische Skalierung des Server-Knotens konfiguriert.

Die OpenShift Machine API ist die Komponente, die die OpenShift-Ressource Machines definiert und verwaltet. Die OpenShift-Ressource Machines stellt die OpenShift-Cluster-Knoten dar. Funktionen der OpenShift Machine API:

- Erstellen, Aktualisieren und Löschen von Machines
- Erstellen der Infrastruktur (Instanz oder VM) für den Knoten

Sie können die OpenShift-Ressource MachineSets verwenden, um Sätze von Machines zu steuern. Ein MachineSet stellt Folgendes dar:

- Einen Satz von Machines
- Eine Abstraktion der zugrunde liegenden Infrastruktur

Bei der Installation von OpenShift auf AWS mit der Full-Stack-Automatisierungsmethode erstellt und konfiguriert das OpenShift-Installationsprogramm eine MachineSet-Ressource für jede Verfügbarkeitszone in der ausgewählten Region.

[user@demo ~]\$ oc get machines -n openshift-machine-api					
NAME	PHASE	TYPE	REGION	ZONE	
ocp4-aws-9r678-master-0	Running	m5.2xlarge	us-east-2	us-east-2a	16h
ocp4-aws-9r678-master-1	Running	m5.2xlarge	us-east-2	us-east-2b	16h
ocp4-aws-9r678-master-2	Running	m5.2xlarge	us-east-2	us-east-2c	16h
ocp4-aws-9r678-worker-us-east-2a-gq2ps	Running	m5.4xlarge	us-east-2	us-east-2a	16h
ocp4-aws-9r678-worker-us-east-2b-slp7l	Running	m5.4xlarge	us-east-2	us-east-2b	16h
ocp4-aws-9r678-worker-us-east-2c-vj7pj	Running	m5.4xlarge	us-east-2	us-east-2c	16h

[user@demo ~]\$ oc get machinesets -n openshift-machine-api					
NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
ocp4-aws-9r678-worker-us-east-2a	1	1	1	1	16h
ocp4-aws-9r678-worker-us-east-2b	1	1	1	1	16h
ocp4-aws-9r678-worker-us-east-2c	1	1	1	1	16h

[user@demo ~]\$ oc get nodes --label-columns \> failure-domain.beta.kubernetes.io/region,failure-domain.beta.kubernetes.io/zone					
NAME	STATUS	ROLES	AGE	VERSION	
REGION	ZONE				
ip-10-0-151-177.us-east-2.compute.internal	Ready	master	16h		
v1.19.0+9f84db3	us-east-2	us-east-2a			
ip-10-0-157-4.us-east-2.compute.internal	Ready	worker	16h		
v1.19.0+9f84db3	us-east-2	us-east-2a			
ip-10-0-166-182.us-east-2.compute.internal	Ready	worker	16h		
v1.19.0+9f84db3	us-east-2	us-east-2b			

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

```
ip-10-0-180-27.us-east-2.compute.internal  Ready  master  17h
  v1.19.0+9f84db3  us-east-2  us-east-2b
ip-10-0-205-233.us-east-2.compute.internal  Ready  master  17h
  v1.19.0+9f84db3  us-east-2  us-east-2c
ip-10-0-217-153.us-east-2.compute.internal  Ready  worker  16h
  v1.19.0+9f84db3  us-east-2  us-east-2c
```

Mithilfe des Workers MachineSets können Sie die Anzahl der auf dem Cluster ausgeführten Server-Knoten hoch- oder herunterskalieren. Beim Hochskalieren eines Worker-MachineSet geschieht Folgendes:

- Die OpenShift Machine API stellt automatisch eine AWS EC2-Instanz für den neuen Server-Knoten bereit und startet sie.
- Der neue Server-Knoten ruft seine Ignition-Konfigurationsdatei ab und installiert RHCOS.
- Der neue Server-Knoten verbindet sich automatisch mit dem OpenShift-Cluster.

```
[user@demo ~]$ oc scale machineset ocp4-aws-9r678-worker-us-east-2c \
> --replicas=2 -n openshift-machine-api
machineset.machine.openshift.io/ocp4-aws-9r678-worker-us-east-2c scaled
```

NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
ocp4-aws-9r678-worker-us-east-2a	1	1	1	1	17h
ocp4-aws-9r678-worker-us-east-2b	1	1	1	1	17h
ocp4-aws-9r678-worker-us-east-2c	2	2	1	1	17h

Nach ein paar Minuten muss der neue Server-Knoten den Status Ready aufweisen.

```
[user@demo ~]$ oc get machinesets -n openshift-machine-api
NAME
      DESIRED  CURRENT  READY  AVAILABLE  AGE
ocp4-aws-9r678-worker-us-east-2a  1        1        1        1        17h
ocp4-aws-9r678-worker-us-east-2b  1        1        1        1        17h
ocp4-aws-9r678-worker-us-east-2c  2        2        2        2        17h
```

```
[user@demo ~]$ oc get machines -n openshift-machine-api
...output omitted..
ocp4-aws-9r678-worker-us-east-2c-65hln  Running  m5.4xlarge  us-east-2
  us-east-2c  3m41s
ocp4-aws-9r678-worker-us-east-2c-vj7pj  Running  m5.4xlarge  us-east-2  us-
  east-2c  17h
```

```
[user@demo ~]$ oc get nodes --label-columns \
> failure-domain.beta.kubernetes.io/region,failure-domain.beta.kubernetes.io/zone
...output omitted..
ip-10-0-196-32.us-east-2.compute.internal  Ready  worker  2m31s
  v1.19.0+9f84db3  us-east-2  us-east-2c
ip-10-0-205-233.us-east-2.compute.internal  Ready  master  17h
  v1.19.0+9f84db3  us-east-2  us-east-2c
```

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

Wie Sie im vorherigen Beispiel sehen können, wurde mit der OpenShift Machine API automatisch ein neuer Server-Knoten bereitgestellt und dem Cluster (`ip-10-0-196-32.us-east-2.compute.internal`) in der gewünschten AWS-VZ us-east-2c hinzugefügt.

Erfassen von OpenShift-Daten

Wenn Administratoren sich zur Behebung eines OpenShift-Problems an den Red Hat Support wenden, werden sie zur Angabe von Cluster-Debug-Informationen aufgefordert. In Abhängigkeit von der betroffenen OpenShift-Komponente können Administratoren unterschiedliche Debug-Mechanismen verwenden.

OpenShift-Cluster-Daten

Sie können Cluster-Debug-Informationen mit dem CLI-Befehl `oc adm must-gather` als Benutzer `cluster-admin` erfassen. Dieser CLI-Befehl sammelt die Informationen aus Ihrem Cluster, z. B.:

- Ressourcendefinitionen
- Audit-Protokolle
- Serviceprotokolle

Durch die Ausführung des Befehls `oc adm must-gather` wird ein neuer Pod im Cluster erstellt. Dieser Pod sammelt die Cluster-Daten und speichert sie in einem neuen Verzeichnis. Der Name des neuen Verzeichnisses beginnt mit `must-gather.local`. Der Befehl `oc adm must-gather` erstellt dieses Verzeichnis im aktuellen Arbeitsverzeichnis.

```
[user@demo ~]$ export KUBECONFIG=${HOME}/ocp4-cluster/auth/kubeconfig
```

```
[user@demo ~]$ cd ${HOME}
[user@demo ~]$ oc adm must-gather
[must-gather      ] OUT Using must-gather plugin-in image: quay.io/openshift-release-dev/ocp-v4.0-art-dev@sha256:47..86
[must-gather      ] OUT namespace/openshift-must-gather-ndwcz created
[must-gather      ] OUT clusterrolebinding.rbac.authorization.k8s.io/must-gather-5ptk2 created
[must-gather      ] OUT pod for plug-in image quay.io/openshift-release-dev/ocp-v4.0-art-dev@sha256:47..86 created
[must-gather-276db] POD Wrote inspect data to must-gather.
...output omitted...
```

```
[user@demo ~]$ ls
install-config.yaml  must-gather.local.1227184995617480385/  ocp4-cluster/
```

```
[user@demo ~]$ find must-gather.local.1227184995617480385/
must-gather.local.1227184995617480385/
must-gather.local.1227184995617480385/timestamp
...output omitted...
```

Wie Sie sehen können, enthält dieses Verzeichnis die OpenShift-Ressourcendefinitionen, -Serviceprotokolle und -Audit-Protokolle.



Anmerkung

Wenn Sie im Red Hat Customer Portal ein OpenShift-Support-Ticket öffnen, erstellen Sie eine TAR-Datei mit der Ausgabe, die bei Ausführung von `oc adm must-gather` generiert wurde, und hängen Sie die Datei an das Support-Ticket an.

```
[user@demo ~]$ tar cvaf must-gather.tar.gz \
> must-gather.local.1227184995617480385/
...output omitted...
```

Um Debug-Informationen über bestimmte Features zu erfassen, verwenden Sie den CLI-Befehl `oc adm must-gather` mit dem Argument `--image` oder `--image-stream`.

```
[user@demo ~]$ oc adm must-gather \
> --image-stream=openshift/must-gather \
> --image=registry.redhat.io/container-native-virtualization/\
> cnv-must-gather-rhel8:v2.5.2
...output omitted...
```

Wenn Sie beispielsweise das Image `cnv-must-gather-rhel8` verwenden, erfasst der Befehl `oc adm must-gather` für OpenShift Virtualization spezifische Daten.

Am häufigsten verwendete must-gather-Images

Image	Zweck
registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel8:v2.5.2	Datensammlung für OpenShift Virtualization
registry.redhat.io/openshift-serverless-1/svls-must-gather-rhel8	Datensammlung für OpenShift Serverless
registry.redhat.io/openshift-service-mesh/istio-must-gather-rhel7	Datensammlung für Red Hat OpenShift Service Mesh
registry.redhat.io/rhcam-1-2/openshift-migration-must-gather-rhel8	Datensammlung für migrationsbezogene Informationen
registry.redhat.io/ocs4/ocs-must-gather-rhel8	Datensammlung für Red Hat OpenShift Container Storage
registry.redhat.io/openshift4/ose-cluster-logging-operator	Datensammlung für Red Hat OpenShift-Cluster-Protokollierung

OpenShift-Knotendaten

In einigen Szenarien werden Sie vom Red Hat Support gebeten, eine `sosreport`-Datei aus einem bestimmten OpenShift-Cluster-Knoten zu erfassen. Der Befehl `sosreport` ist ein Tool, das Konfigurationsdetails, Systeminformationen und Diagnosedaten aus Red Hat Enterprise Linux (RHEL)- und Red Hat Enterprise Linux CoreOS (RHCOS)-Systemen sammelt.

Red Hat empfiehlt die Verwendung eines Debug-Pods zum Generieren einer `sosreport`-Datei aus einem OpenShift-Cluster-Knoten.

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

```
[user@demo ~]$ oc debug node/ip-10-0-151-177.us-east-2.compute.internal
...output omitted...
sh-4.4# chroot /host
sh-4.4# toolbox
Trying to pull registry.redhat.io/rhel8/support-tools...
...output omitted...

[root@ip-10-0-151-177 /]# sosreport -k crio.all=on -k crio.logs=on
...output omitted...
Your sosreport has been generated and saved in:
/host/var/tmp/sosreport-ip-10-0-151-177-1234-2021-01-11-crlsmlr.tar.xz

Size   31.07MiB
Owner   root
md5    3ddfb7a774002fc8fb18da9c9c1534bc

Please send this file to your support representative.

[root@ip-10-0-151-177 /]# exit
sh-4.4# exit
sh-4.4# ls -lrt \
> /host/var/tmp/sosreport-ip-10-0-151-177-1234-2021-01-11-crlsmlr.tar.xz
-rw-----. 1 root root 32578896 Jan 11 13:42 /host/var/tmp/sosreport-
ip-10-0-151-177-1234-2021-01-11-crlsmlr.tar.xz
```

Remote-Zustandsüberwachung in OpenShift

OpenShift sammelt anonymisierte aggregierte Informationen über die Integrität, Verwendung und Größe der Cluster. Mit diesen Informationen kann Red Hat proaktiv auf Probleme reagieren, die Auswirkungen auf Kunden haben können.

Diese Informationen werden Red Hat über zwei Komponenten vom OpenShift-Cluster zur Verfügung gestellt:

- Telemetrie
- Insights-Operator

Die Telemetriekomponente sendet eine ausgewählte Teilmenge der Cluster-Überwachungsmetriken an Red Hat. Diese Metriken werden kontinuierlich gesendet und beschreiben Folgendes:

- Größe des OpenShift-Clusters
- Integrität der OpenShift-Komponenten
- Integrität von OpenShift-Upgrades
- Eingeschränkte Informationen zur OpenShift-Nutzung
- Zusammenfassung von OpenShift-Alarmen

Der Insights-Operator erfasst regelmäßig den Status von Cluster-Konfiguration und Komponentenfehlern und übermittelt diese Daten an Red Hat. Mithilfe dieser Informationen identifiziert Red Hat OpenShift Cluster Manager proaktiv potenzielle Cluster-Probleme und stellt Lösungen und vorbeugende Maßnahmen bereit.



Anmerkung

Weitere Informationen finden Sie im Leitfaden *Remote health monitoring with connected clusters* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/support

Zum Zugreifen auf Ihre Cluster-Informationen können Sie die Red Hat OpenShift Cluster Manager-Konsole [<https://cloud.redhat.com/openshift>] verwenden. Wenn Sie mehrere OpenShift-Cluster verwalten, benötigen Sie Ihre **Cluster-ID**, um Ihren Cluster in der Red Hat OpenShift Cluster Manager-Konsole zu identifizieren.

```
[user@demo ~]$ oc get clusterversion \
> -o jsonpath='{.items[].spec.clusterID}{"\n"}'
9d1c5e73-9deb-452b-b327-376d04315246
```

Öffnen Sie nach Abrufen der **Cluster-ID** Ihren Webbrowser, und navigieren Sie mit Ihrem Red Hat-Benutzerkonto zur Red Hat OpenShift Cluster Manager-Konsole [<https://cloud.redhat.com/openshift>]. Klicken Sie dann auf den Link Ihrer **Cluster-ID**, und überprüfen Sie Ihre Cluster-Informationen auf den Navigationsregisterkarten „Overview“, „Monitoring“, „Insights“ und „Support“.

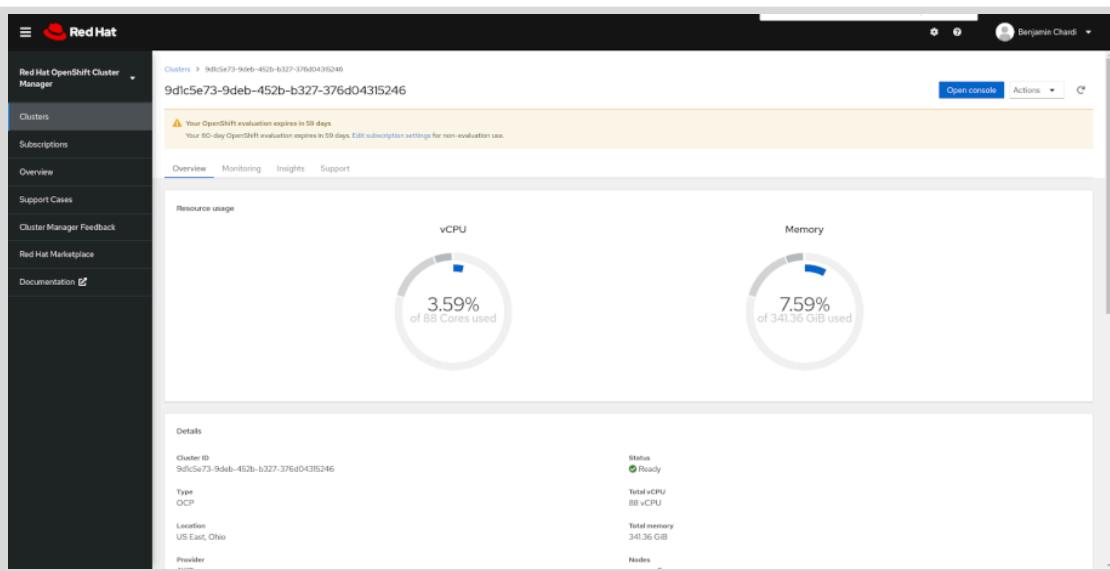


Abbildung 1.9: Red Hat OpenShift Cluster Manager-Konsole

Löschen eines OpenShift-Clusters

Wenn Sie Ihren OpenShift-Cluster entfernen müssen, können Sie das OpenShift-Installationsprogramm mit der Option `destroy cluster` ausführen.

```
[user@demo ~]$ openshift-install destroy cluster \
> --dir=${HOME}/ocp4-cluster
...output omitted...
```



Literaturhinweise

- Weitere Informationen finden Sie im Kapitel *Installation configuration* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing
- Weitere Informationen finden Sie im Abschnitt *Installing a cluster on AWS with customizations* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_aws

► Quiz

Ausführen des OpenShift-Installationsprogramms

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. **Wählen Sie die richtige Schrittfolge aus, die bei der Installation von OpenShift mit dem OpenShift-Installationsprogramm auszuführen ist.**
- a. Ignition-Konfigurationsdateien erstellen, Kubernetes-Manifeste erstellen, install-config.yaml erstellen, Cluster erstellen
 - b. Kubernetes-Manifeste erstellen, install-config.yaml erstellen, Ignition-Konfigurationsdateien erstellen, Cluster erstellen
 - c. install-config.yaml erstellen, Kubernetes-Manifeste erstellen, Ignition-Konfigurationsdateien erstellen, Cluster erstellen
 - d. install-config.yaml erstellen, Ignition-Konfigurationsdateien erstellen, Kubernetes-Manifeste erstellen, Cluster erstellen
- 2. **Kubernetes-Manifeste und Ignition-Konfigurationsdateien werden automatisch im Schritt der Cluster-Erstellung erstellt, sofern sie noch nicht vorhanden sind. (Richtig oder falsch)**
- a. Richtig
 - b. Falsch
- 3. **Welche zwei der folgenden Vorgehensweisen werden bei der Installation von OpenShift empfohlen? (Wählen Sie zwei Antworten aus.)**
- a. Das OpenShift-Installationsprogramm auf einem Bastion-Host ausführen
 - b. Mindestens 5 Control Plane-Knoten einrichten
 - c. Benutzerdefinierte Operator-Container-Images verwenden
 - d. Sicherstellen, dass die Cluster-Knoten Zugriff auf einen NTP-Server haben
- 4. **Welcher der folgenden Befehle kann auf dem Bastion-Host ausgeführt werden, um die Cluster-Verwaltung mit dem Befehl oc zu ermöglichen?**
- a. export KUBEADMIN=\${HOME}/ocp4-aws-cluster/auth/kubeconfig
 - b. export KUBECONFIG=\${HOME}/ocp4-aws-cluster/auth/kubeconfig
 - c. export KUBECONFIG=\${HOME}/ocp4-aws-cluster/auth/kubeadmin

► Lösung

Ausführen des OpenShift-Installationsprogramms

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. **Wählen Sie die richtige Schrittfolge aus, die bei der Installation von OpenShift mit dem OpenShift-Installationsprogramm auszuführen ist.**
 - a. Ignition-Konfigurationsdateien erstellen, Kubernetes-Manifeste erstellen, install-config.yaml erstellen, Cluster erstellen
 - b. Kubernetes-Manifeste erstellen, install-config.yaml erstellen, Ignition-Konfigurationsdateien erstellen, Cluster erstellen
 - c. install-config.yaml erstellen, Kubernetes-Manifeste erstellen, Ignition-Konfigurationsdateien erstellen, Cluster erstellen
 - d. install-config.yaml erstellen, Ignition-Konfigurationsdateien erstellen, Kubernetes-Manifeste erstellen, Cluster erstellen
- ▶ 2. **Kubernetes-Manifeste und Ignition-Konfigurationsdateien werden automatisch im Schritt der Cluster-Erstellung erstellt, sofern sie noch nicht vorhanden sind. (Richtig oder falsch)**
 - a. Richtig
 - b. Falsch
- ▶ 3. **Welche zwei der folgenden Vorgehensweisen werden bei der Installation von OpenShift empfohlen? (Wählen Sie zwei Antworten aus.)**
 - a. Das OpenShift-Installationsprogramm auf einem Bastion-Host ausführen
 - b. Mindestens 5 Control Plane-Knoten einrichten
 - c. Benutzerdefinierte Operator-Container-Images verwenden
 - d. Sicherstellen, dass die Cluster-Knoten Zugriff auf einen NTP-Server haben
- ▶ 4. **Welcher der folgenden Befehle kann auf dem Bastion-Host ausgeführt werden, um die Cluster-Verwaltung mit dem Befehl oc zu ermöglichen?**
 - a. export KUBEADMIN=\${HOME}/ocp4-aws-cluster/auth/kubeconfig
 - b. export KUBECONFIG=\${HOME}/ocp4-aws-cluster/auth/kubeconfig
 - c. export KUBECONFIG=\${HOME}/ocp4-aws-cluster/auth/kubeadmin

► Angeleitete Übung

Erfüllen der OpenShift-Installationsvoraussetzungen

In dieser Übung bereiten Sie die Umgebung für die Installation von OpenShift in einer bereits vorhandenen Infrastruktur vor.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Herunterladen und Installieren der Binärdateien `oc` und `openshift-install`
- Generieren des SSH-Schlüssels für die Fehlerbehebung
- Ermitteln der lokalen Registry für Ihre Kursumgebungsregion
- Herunterladen und Konfigurieren des `pull-secret` für die Verwendung einer lokalen Registry
- Überprüfen der `pull-secret`-Anmelde Daten
- Überprüfen der DNS-Einträge für die OpenShift-API- und APP Ingress-Load Balancer.

Bevor Sie Beginnen

Um diese Übung durchzuführen, stellen Sie sicher, dass Sie die Lektionen *Einführung in OpenShift-Installationsmethoden* und *Ausführen des OpenShift-Installationsprogramms* abgeschlossen haben. Wenn Sie Änderungen an der Kursumgebung vorgenommen haben, müssen Sie den Kursraum löschen und neu erstellen, um diese Aktivität erfolgreich abschließen zu können.



Anmerkung

Die Firefox-Version auf dem Rechner „workstation“ rendert die Website <https://cloud.redhat.com/openshift/> nicht ordnungsgemäß. Um diese angeleitete Übung durchzuführen, installieren und verwenden Sie den Chromium-Webbrowser auf dem Rechner „workstation“:

```
[student@workstation ~]$ sudo yum install chromium
```

Anweisungen

- 1. Melden Sie sich beim Server `utility` an, um die CLI-Tools herunterzuladen und zu installieren.
 - 1.1. Melden Sie sich beim Server `utility` als Benutzer `lab` an. Machen Sie sich zum Benutzer `root`.

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

```
[student@workstation ~]$ ssh lab@utility  
...output omitted...  
[lab@utility ~]$ sudo -i  
[root@utility ~]#
```

- 1.2. Laden Sie die Binärdateien oc und openshift-install für die OpenShift-Version 4.6.4 herunter.

```
[root@utility ~]# mirror="https://mirror.openshift.com/pub/openshift-v4/clients"  
[root@utility ~]# wget ${mirror}/ocp/4.6.4/openshift-client-linux-4.6.4.tar.gz  
...output omitted...  
  
[root@utility ~]# wget ${mirror}/ocp/4.6.4/openshift-install-linux-4.6.4.tar.gz  
...output omitted...
```

- 1.3. Extrahieren Sie die Binärdateien oc und openshift-install, und kopieren Sie sie in das Verzeichnis /usr/bin.

```
[root@utility ~]# tar -xvf openshift-client-linux-4.6.4.tar.gz -C /usr/bin/  
README.md  
oc  
kubectl  
  
[root@utility ~]# tar -xvf openshift-install-linux-4.6.4.tar.gz -C /usr/bin/  
README.md  
openshift-install
```

- 1.4. Legen Sie die Bash-Vervollständigung für oc und openshift-install fest. Führen Sie die Bash-Vervollständigungsdateien als Benutzer root mit dem Befehl „source“ aus. Führen Sie die Bash-Vervollständigungsdateien außerdem als Benutzer lab mit dem Befehl „source“ aus.

```
[root@utility ~]# oc completion bash > /etc/bash_completion.d/openshift
```

```
[root@utility ~]# openshift-install completion \  
> bash > /etc/bash_completion.d/openshift-install
```

```
[root@utility ~]# source /etc/bash_completion.d/openshift  
[root@utility ~]# source /etc/bash_completion.d/openshift-install  
[root@utility ~]# exit  
logout  
[lab@utility ~]$ source /etc/bash_completion.d/openshift  
[lab@utility ~]$ source /etc/bash_completion.d/openshift-install
```

- 1.5. Überprüfen Sie als Benutzer lab die Installation der Binärdateien oc und openshift-install.

```
[lab@utility ~]$ oc version  
Client Version: 4.6.4
```

```
[lab@utility ~]$ openshift-install version
openshift-install 4.6.4
built from commit 6e0...6c5
release image quay.io/openshift-release-dev/ocp-release@sha256:66...fc
```

► 2. Erstellen Sie den SSH-Schlüssel für die Installation.

- 2.1. Führen Sie zum Erstellen des SSH-Schlüssels den folgenden Befehl aus.

```
[lab@utility ~]$ ssh-keygen -t rsa -b 4096 -N '' -f .ssh/ocp4upi
Generating public/private rsa key pair.
Your identification has been saved in .ssh/ocp4upi.
Your public key has been saved in .ssh/ocp4upi.pub.
...output omitted...
```

► 3. Ermitteln Sie den FQDN der lokalen Registry für die Region Ihrer Kursumgebung. Sie können die Region Ihrer Kursumgebung auf der Registerkarte **Lab Environment** auf der Webseite des Kurses „Red Hat OpenShift Installation Lab“ ermitteln. Installieren Sie den Chromium-Webbrowser und verwenden Sie ihn während der restlichen angeleiteten Übung. In der folgenden Tabelle wird der FQDN der lokalen Registry für die einzelnen Regionen aufgeführt.

FQDN der lokalen Registry nach Kursumgebungsregion

Region	FQDN der lokalen Registry
Nordamerika	nexus-registry-int.apps.tools-na.prod.nextcle.com
EMEA	nexus-registry-int.apps.tools-eu.prod.nextcle.com
APAC	nexus-registry-int.apps.tools-ap.prod.nextcle.com

In diesem Kurs verwenden Sie eine lokale Registry, die die aus `quay.io` gespiegelten OpenShift-Release-Images enthält. Die lokale Registry wird in derselben Region wie die Kursumgebung ausgeführt, und ihr vollständig qualifizierter Domain-Name (FQDN) hängt von dieser Region ab.

- 3.1. Installieren Sie den Chromium-Webbrowser über einen Terminal auf dem Rechner **workstation**.

```
[student@workstation ~]$ sudo yum install chromium
...output omitted...
```

Wenn Sie dazu aufgefordert werden, geben Sie yes ein und drücken Sie die **Eingabetaste**, um die Installation zu akzeptieren.

- 3.2. Ermitteln Sie nun die Region Ihrer Kursumgebung.

Navigieren Sie auf dem Rechner **workstation** mit dem Chromium-Webbrowser zur Registerkarte **Lab Environment** auf der Webseite des Kurses „Red Hat OpenShift Installation Lab“.

Klicken Sie auf **Information**, um Informationen zur Kursumgebung anzuzeigen. Die Kursumgebungsregion finden Sie im Feld **Published region**.

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

The screenshot shows the 'Lab Controls' section of the OpenShift Lab Environment. It displays a table of three virtual machines: 'bastion', 'bootstrap', and 'classroom', all currently in a 'stopped' state. A tooltip is shown over the 'Published region' field for the 'bastion' row, which is highlighted in grey. The tooltip contains the text 'Published region: northamerica' and 'openstack region: us-east-1'. At the top of the interface, there are tabs for 'Table of Contents', 'Course', and 'Lab Environment', along with a star icon and a help icon. A red 'WATCH TUTORIAL' button is located in the top right corner of the main content area.

- 3.3. Suchen Sie anhand der vorherigen Tabelle nach dem FQDN der lokalen Registry für Ihre Region.
- 3.4. Überprüfen Sie als Benutzer `lab` auf dem Server `utility` mit dem Befehl `curl`, ob Sie mit dem FQDN der lokalen Registry für Ihre Kursumgebung kommunizieren können.

Der folgende Befehl geht davon aus, dass die Kursumgebungsregion `northamerica` ist, die als FQDN der lokalen Registry `nexus-registry-int.apps.tools-na.prod.nextcle.com` verwendet.

```
[lab@utility ~]$ curl -s \
> https://nexus-registry-int.apps.tools-na.prod.nextcle.com -o /dev/null; echo $?
0
```

Wenn Sie den falschen FQDN für die lokalen Registry verwenden, ist die Ausgabe des Befehls `curl` nicht 0.

**Anmerkung**

Standardmäßig wird die lokale Registry der Region `northamerica` (`nexus-registry-int.apps.tools-na.prod.nextcle.com`) zum Beispiel in allen Lektionen und angeleiteten Übungen dieses Kurses verwendet.

Stellen Sie sicher, dass Sie in allen angeleiteten Übungen den FQDN der lokalen Registry der Region Ihrer Kursumgebung verwenden. Wenn Sie eine lokale Registry aus anderen Regionen verwenden, schlagen die angeleiteten Übungen fehl.

- ▶ 4. Rufen Sie als Benutzer `student` auf dem Rechner `workstation` ein `pull-secret` von `cloud.redhat.com` ab. Ersetzen Sie die Anmelddaten für die Registry `quay.io` durch die Anmelddaten für Ihre lokale Registry.

Angenommen, die Region Ihrer Kursumgebung ist `northamerica`. In diesem Fall lautet der Inhalt des `pull-secret` für die lokale Registry wie folgt:

```
"nexus-registry-int.apps.tools-na.prod.nextcle.com": 
{"auth":"cmVndXNlcjpJbnN0YWxsTTM=", "email":"nobody@example.com"}
```

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

- 4.1. Navigieren Sie auf dem Rechner **workstation** mit dem Chromium-Webbrowser zu <https://cloud.redhat.com/openshift/install/metal/user-provisioned>. Melden Sie sich mit den Anmeldedaten Ihres Red Hat-Benutzerkontos an.
- 4.2. Klicken Sie auf **Download pull secret**. Wählen Sie dann **Save File** aus, und klicken Sie auf **OK**.
- 4.3. Öffnen Sie auf dem Rechner **workstation** ein Terminal, und formatieren Sie die **pull-secret**-Datei im Ordner **/home/student/Downloads** als JSON-Objekt:

```
[lab@utility ~]$ exit  
...output omitted...  
[student@workstation ~]$
```

```
[student@workstation ~]$ python3 -m json.tool \  
> Downloads/pull-secret > pull-secret.json
```

```
[student@workstation ~]$ cat pull-secret.json  
{  
    "auths": {  
        "cloud.openshift.com": {  
            "auth":  
"UxUUjYwSTMyb3BlUxUUjYwSTMybnNUxUUjYwSTMyoawZUxUU...YMy3NfNGUxUUjYw==",  
            "email": "student@redhat.com"  
        },  
        "quay.io": {  
  
            "auth": "UxUUjYwSTMyb3BlUxUUjYwSTMybnNUxUUjYwSTMyoawZUxUU...YMy3NfNGUxUUjYw==",  
            "email": "student@redhat.com"  
        },  
        "registry.connect.redhat.com": {  
            "auth":  
"CvmsWaJUROSkhCkEQ71NiM1BsracE9Z0VmBMIDrS3R20KOH8Eq...Tx09phozeXpqKLJN=",  
            "email": "student@redhat.com"  
        },  
        "registry.redhat.io": {  
            "auth":  
"CvmsWaJUROSkhCkEQ71NiM1BsracE9Z0VmBMIDrS3R20KOH8Eq...Tx09phozeXpqKLJN=",  
            "email": "student@redhat.com"  
        }  
    }  
}
```

- 4.4. Bearbeiten Sie die Datei **pull-secret.json**, indem Sie die Anmeldedaten für **quay.io** durch die Anmeldedaten für Ihre lokale Registry ersetzen.



Anmerkung

Der Inhalt des `pull-secret` für die lokale Registry in der Region `northamerica` lautet:

```
"nexus-registry-int.apps.tools-na.prod.nextcle.com":  
{"auth": "cmVndXNlcjpJbnN0YWxstTM=", "email": "nobody@example.com"}
```

Die Werte der Attribute `auth` und `email` sind für alle lokalen Registries identisch, unabhängig von der Kursumgebungsregion.

```
[student@workstation ~]$ vi pull-secret.json  
{  
    "auths": {  
        "cloud.openshift.com": {  
            "auth":  
                "UxUUjYwSTMyb3BlUxUUjYwSTMybnNUxUUjYwSTMyoawZUxUU...YMy3NfNGUxUUjYw==",  
            "email": "student@redhat.com"  
        },  
        "nexus-registry-int.apps.tools-na.prod.nextcle.com": {  
            "auth": "cmVndXNlcjpJbnN0YWxstTM=",  
            "email": "nobody@example.com"  
        },  
        "registry.connect.redhat.com": {  
            "auth":  
                "CvmsWaJUROSkhCkEQ71NiM1BsracE9Z0VmBMIDrs3R20K0H8Eq...Tx09phozeXpqKLJN=",  
            "email": "student@redhat.com"  
        },  
        "registry.redhat.io": {  
            "auth":  
                "CvmsWaJUROSkhCkEQ71NiM1BsracE9Z0VmBMIDrs3R20K0H8Eq...Tx09phozeXpqKLJN=",  
            "email": "student@redhat.com"  
        }  
    }  
}
```

Speichern Sie die Datei, und schließen Sie anschließend den Editor.

- 4.5. Generieren Sie mithilfe des Tools `jq` die kompakte Version der Datei `pull-secret.json` in der Datei `pull-secret-oneline.json`.

```
[student@workstation ~]$ cat pull-secret.json | jq . -c > pull-secret-oneline.json
```



Anmerkung

Mit der im vorherigen Befehl verwendeten `jq`-Analyse wird zudem überprüft, ob die Datei `pull-secret.json` gültig ist. Wenn beim Ausführen dieses Befehls kein Fehler auftritt, sind die Dateien `pull-secret.json` und `pull-secret-oneline.json` gültig.

```
[student@workstation ~]$ cat pull-secret-oneline.json
{"auths":{"cloud.openshift.com":
{"auth":"UxUUj...UujYw==","email":"student@redhat.com"},"nexus-
registry-int.apps.tools-na.prod.nextcle.com":
{"auth":"cmVnd...sTTM=","email":"nobody@example.com"},"registry.connect.redhat.com":
 {"auth":"CvmsW...qKLJN=","email":"student@redhat.com"},"registry.redhat.io":
 {"auth":"CvmsW...qKLJN=","email":"student@redhat.com"}}}
```

4.6. Kopieren Sie die Datei `pull-secret-oneline.json` auf den Server `utility`.

```
[student@workstation ~]$ scp pull-secret-oneline.json lab@utility:
...output omitted...
pull-secret-oneline.json
```

Sie müssen die Datei `pull-secret-oneline.json` später in diesem Kurs verwenden, um die Datei `install-config.yaml` zu vervollständigen.

- 5. Überprüfen Sie als Benutzer `lab` auf dem Server `utility`, ob die in der Datei `~/pull-secret-oneline.json` gespeicherten Anmelddaten gültig sind, um Images aus der lokalen Registry und der Registry `registry.redhat.io` abzurufen.
- 5.1. Installieren Sie das Tool `podman`, um das OpenShift-Installations-Release-Image aus der lokalen Registry abzurufen.

```
[student@workstation ~]$ ssh lab@utility
...output omitted...
```

```
[lab@utility ~]$ sudo yum install podman
...output omitted...
Complete!
```

```
[lab@utility ~]$ sudo podman pull --authfile ~/pull-secret-oneline.json \
> nexus-registry-int.apps.tools-na.prod.nextcle.com/openshift/ocp4:4.6.4-x86_64
Trying to pull nexus-registry-int.apps.tools-na.prod.nextcle.com/openshift/
ocp4:4.6.4-x86_64...
Getting image source signatures
Copying blob ec1681b6a383 done
Copying blob c4d668e229cd done
Copying blob 6b8afa2699e1 done
Copying blob 12dcdecc154a done
Copying blob 1cefa0e873d7 done
Copying blob 09bfd7265184 done
Copying config 26f7cd4cf1 done
Writing manifest to image destination
Storing signatures
26f...fc9
```

5.2. Überprüfen Sie als Benutzer `lab` auf dem Server `utility`, ob die in der Datei `~/pull-secret-oneline.json` gespeicherten Anmelddaten gültig sind, um Images aus der Registry `registry.redhat.io` abzurufen.

Kapitel1 | Beschreiben des OpenShift-Installationsprozesses

```
[lab@utility ~]$ sudo podman pull --authfile ~/pull-secret-oneline.json \
> registry.redhat.io/ubi8/ubi:latest
Trying to pull registry.redhat.io/ubi8/ubi...
Getting image source signatures
Copying blob d9e72d058dc5 done
Copying blob cca21acb641a done
Copying config 3269c37eae done
Writing manifest to image destination
Storing signatures
326...a5e
```

**Anmerkung**

Wenn Sie die `podman pull`-Befehle mit einer ungültigen `pull-secret`-Datei (oder mit dem falschen FQDN für die lokale Registry) ausführen, schlagen die Befehle fehl. Wenn die Ausführung des Befehls `podman pull` fehlschlägt, wiederholen Sie die vorherigen Schritte ab dem Schritt, in dem Sie den FQDN der lokalen Registry für die Region Ihrer Kursumgebung ermitteln.

5.3. Überprüfen Sie, ob Sie die Test-Images richtig abgerufen haben.

```
[lab@utility ~]$ sudo podman images
REPOSITORY                                     TAG
IMAGE ID          CREATED        SIZE
registry.redhat.io/ubi8/ubi                   latest
3269c37eae33    8 weeks ago   208 MB
nexus-registry-int.apps.tools-na.prod.nextcle.com/openshift/ocp4  4.6.4-x86_64
26f7cd4cf1fb    2 months ago  322 MB
```

- 6. Überprüfen Sie als Benutzer `lab` auf dem Server `utility` die DNS-Voraussetzungen für die OpenShift-API- und APP Ingress-Load Balancer. Der DNS-Eintrag des OpenShift-API-Load Balancers und der DNS-Platzhaltereintrag des OpenShift-APP Ingress-Load Balancers müssen die IP-Adresse des Load Balancers verwenden.

In den folgenden Kapiteln werden folgende Kenntnisse vermittelt:

- Der Load Balancer der Kursumgebung wird auf dem Server `utility` mit der IP-Adresse `192.168.50.254` ausgeführt.
 - Da Sie einen OpenShift-Cluster namens `ocp4` in der Domain `example.com` installieren, lautet die OpenShift-Cluster-Domain `ocp4.example.com`.
- 6.1. Überprüfen Sie, ob der DNS-Eintrag des OpenShift-API-Load Balancers für die Verwendung der IP-Adresse des Load Balancers der Kursumgebung (192.168.50.254) konfiguriert ist.

```
[lab@utility ~]$ dig api.ocp4.example.com

; <>> DiG 9.11.13-RedHat-9.11.13-3.el8 <>> api.ocp4.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47126
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2
```

```
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags;; udp: 4096  
; COOKIE: 386ab7d875e394653ca11a22601bfe40a8ddbd530978d9d7 (good)  
;; QUESTION SECTION:  
;api.ocp4.example.com. IN A  
  
;; ANSWER SECTION:  
api.ocp4.example.com. 86400 IN A 192.168.50.254  
  
;; AUTHORITY SECTION:  
example.com. 86400 IN NS dns.ocp4.example.com.  
  
;; ADDITIONAL SECTION:  
dns.ocp4.example.com. 86400 IN A 192.168.50.254  
  
;; Query time: 2 msec  
;; SERVER: 172.25.250.254#53(172.25.250.254)  
;; WHEN: Thu Feb 04 09:01:36 EST 2021  
;; MSG SIZE rcvd: 127
```

```
[lab@utility ~]$ dig api-int.ocp4.example.com  
  
; <>> DiG 9.11.13-RedHat-9.11.13-3.el8 <>> api-int.ocp4.example.com  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26234  
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags;; udp: 4096  
; COOKIE: 010170465441c41bb60331c9601bfe78fe03a7a63159f818 (good)  
;; QUESTION SECTION:  
;api-int.ocp4.example.com. IN A  
  
;; ANSWER SECTION:  
api-int.ocp4.example.com. 86400 IN A 192.168.50.254  
  
;; AUTHORITY SECTION:  
example.com. 86400 IN NS dns.ocp4.example.com.  
  
;; ADDITIONAL SECTION:  
dns.ocp4.example.com. 86400 IN A 192.168.50.254  
  
;; Query time: 2 msec  
;; SERVER: 172.25.250.254#53(172.25.250.254)  
;; WHEN: Thu Feb 04 09:02:32 EST 2021  
;; MSG SIZE rcvd: 131
```

- 6.2. Überprüfen Sie, ob der DNS-Platzhaltereintrag des OpenShift-APP Ingress-Load Balancers für die Verwendung der IP-Adresse des Load Balancers der Kursumgebung (192.168.50.254) konfiguriert ist.

```
[lab@utility ~]$ dig test.apps.ocp4.example.com

; <>> DiG 9.11.13-RedHat-9.11.13-3.el8 <>> test.apps.ocp4.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47551
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 1c160cfb1d57ff3f547b7e85601bff32fe98468a22aea3f9 (good)
;; QUESTION SECTION:
;test.apps.ocp4.example.com. IN A

;; ANSWER SECTION:
test.apps.ocp4.example.com. 86400 IN A 192.168.50.254

;; AUTHORITY SECTION:
example.com. 86400 IN NS dns.ocp4.example.com.

;; ADDITIONAL SECTION:
dns.ocp4.example.com. 86400 IN A 192.168.50.254

;; Query time: 3 msec
;; SERVER: 172.25.250.254#53(172.25.250.254)
;; WHEN: Thu Feb 04 09:05:38 EST 2021
;; MSG SIZE rcvd: 133
```

Beenden

Nehmen Sie erst in der nächsten angeleiteten Übung weitere Änderungen an der Übungsumgebung vor. Sie werden diese Umgebung in den nächsten Übungen weiter verwenden.

Hiermit ist die angeleitete Übung beendet.

Einführung in gehostete OpenShift-Angebote

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die Unterschiede zwischen einem selbstverwalteten OpenShift-Cluster und gehosteten OpenShift-Angeboten beschreiben können.

Vergleichen von selbstverwalteten und gehosteten OpenShift-Angeboten

OpenShift 4 bietet je nach Kundenanforderungen eine Vielzahl von Installationsmethoden, einschließlich „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“, sowohl für On-Premises-Installationen als auch für Installationen bei IaaS-Cloud-Anbietern. Diese Installationsmethoden, der Cluster-Betrieb und die Wartung erfordern mehr oder weniger Fachwissen und Arbeitsaufwand von den Cluster-Administratoren.

Kunden, die den Einsatz von OpenShift in Produktionsumgebungen planen, aber nicht bereit sind, die erforderliche Zeit und Ressourcen zu investieren, können sich an Red Hat und andere Cloud-Anbieter wenden, die gehostete OpenShift-Services anbieten.

- Mit gehosteten Services können OpenShift-Benutzer die Bereitstellung, Installation und Konfiguration der erforderlichen Infrastruktur für die Ausführung ihrer Cluster auslagern.
- Gehostete Services entlasten Kunden auch von der Verwaltung der Cluster, einschließlich Upgrades auf die neuesten unterstützten Versionen von OpenShift.
- Darüber hinaus können Kunden, die gehostete Services nutzen, von der Erfahrung von Red Hat bei der Verwaltung von OpenShift in einer Cloud-Umgebung profitieren.

Allerdings ist die Verwendung gehosteter OpenShift-Services im Vergleich zu selbstverwalteten OpenShift-Installationen auch mit einigen Einschränkungen verbunden:

- Gehostete Services bieten keine Benutzerkonten mit Administratorberechtigungen zur Verwaltung der zugrunde liegenden Infrastruktur.
- Die Anzahl der Anpassungen während der Cluster-Installation ist im Vergleich zu einer selbstverwalteten Installation begrenzt.
- Spezifische Anpassungen auf Knotenebene sind möglicherweise nicht verfügbar (wie die Anzahl der Pods pro Knoten).
- Features der OpenShift-Technologievorschau sind möglicherweise nicht verfügbar.
- Eine kuratierte Liste von Operatoren ist verfügbar. Administratoren können Operatoren nur in bestimmten Namespaces installieren.
- Anwendungen, die spezielle Hardware-Features erfordern, können nur begrenzt bereitgestellt werden.

Einführung in gehostete OpenShift-Angebote

Wie bereits erwähnt, bietet Red Hat in Partnerschaft mit anderen Anbietern eine Vielzahl gehosteter (oder verwalteter) OpenShift-Services als Alternative zu selbstverwalteten OpenShift-Installationen an.

In der folgenden Tabelle finden Sie einen allgemeinen Vergleich der unterschiedlichen gehosteten OpenShift-Angebote:

Gehostetes Angebot	In der Cloud gehostet	In Rechnung gestellt von	Verwaltet von	Unterstützt von
Microsoft Azure Red Hat OpenShift	Azure	Microsoft	Red Hat und Microsoft	Red Hat und Microsoft
Red Hat OpenShift Dedicated	AWS oder GCP	Red Hat (OpenShift), AWS oder GCP (Infrastruktur)	Red Hat	Red Hat
Red Hat OpenShift auf IBM Cloud	IBM Cloud	IBM	IBM	Red Hat und IBM
Red Hat OpenShift Online	Red Hat	Red Hat	Red Hat	Red Hat
Red Hat OpenShift Service auf AWS	AWS	AWS	Red Hat und AWS	Red Hat und AWS

Weitere Informationen zu den einzelnen gehosteten Services finden Sie hier:

- Microsoft Azure Red Hat OpenShift: <https://www.openshift.com/products/azure-openshift>
- Red Hat OpenShift Dedicated: <https://www.openshift.com/products/dedicated/>
- Red Hat OpenShift auf IBM Cloud: <https://www.openshift.com/products/openshift-ibm-cloud/>
- Red Hat OpenShift Online: <https://www.openshift.com/products/online/>
- Red Hat OpenShift Service auf AWS: <https://www.openshift.com/products/amazon-openshift/>

Vergleich der gehosteten OpenShift-Angebote mit den Managed-Kubernetes-Angeboten von Drittanbietern

Gehostete OpenShift-Angebote bieten Kunden die Möglichkeit, Workloads innerhalb von Minuten nach dem Kauf des Service bereitzustellen. Neben den von Kubernetes bereitgestellten Funktionen bietet OpenShift eine vollständige Plattform für Anwendungsentwicklung, -bereitstellung und Laufzeit. Es umfasst auch Features wie Pipelines, Monitoring, Logging, Sicherheit und das Red Hat Service Mesh.

Da OpenShift auf einer Vielzahl von Cloud-Anbietern ausgeführt werden kann, bietet es überall eine einheitliche Plattformerfahrung, unabhängig davon, wo es ausgeführt wird. In der folgenden

Kapitel 1 | Beschreiben des OpenShift-Installationsprozesses

Tabelle werden allgemeine Aspekte von gehosteten OpenShift-Angeboten und von Drittanbietern verwaltete Kubernetes-Angebote verglichen:

	Von Red Hat gehostete Services	Von Drittanbietern verwaltete Kubernetes-Services
Betriebssystem	RHEL/RHCOS	Variiert
Plattform	Kubernetes + OCP	Kubernetes
Cloud-übergreifende Portierbarkeit	Portierbar auf andere Betriebssysteme oder Kubernetes	Nur Kubernetes, andere Services sind anbieterspezifisch
Identität und Authentifizierung	LDAP, Google, OpenID, GitHub, Gitlab	Anbieterspezifisch (IAM, Azure AD, Google/Cloud IAM)
Protokolle	EFK, Protokollweiterleitung (OpenShift Dedicated) oder Azure Monitor	Keine Standardangebote
Metriken	Prometheus und Grafana oder Azure Monitor	Keine Standardangebote
CLIs und APIs	oc und kubectl	kubectl + anbieterspezifisch für Infrastruktur
Cluster-Netzwerk	Von Red Hat verwaltet	Im Allgemeinen selbstverwaltet
CI/CD	S2I, Image Streams, Jenkins, Tekton	Keine Standardangebote
Katalog	OperatorHub, Red Hat Marketplace	Anbieterspezifische oder Community-Quellen
Container-Registry	Enthalten	Keine Standardangebote
Support	24x7 Premium-Support	Basis-Support, Erwerb höherer Support-Levels möglich
Updates	Von Red Hat verwaltet	Mischung aus automatisierten Anbieter- und manuellen Kundenaktivitäten
Snapshots	Von Red Hat verwaltet für Plattform und PVs	Vom Kunden verwaltet
Workload-Benachrichtigungen	Grafana und Prometheus (OpenShift Dedicated), OCP Web Console	Keine Standardangebote

	Von Red Hat gehostete Services	Von Drittanbietern verwaltete Kubernetes-Services
Plattform-Benachrichtigungen	Status-Portal	Anbieterspezifische Dashboards



Literaturhinweise

Verwaltete Red Hat OpenShift-Services

<https://www.openshift.com/learn/topics/managed-services>

Seite mit OpenShift-Produkten

<https://www.openshift.com/products>

► Quiz

Einführung in gehostete OpenShift-Angebote

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche vier der folgenden Cloud-Anbieter bieten einen gehosteten OpenShift-Service an? (Wählen Sie vier Antworten aus.)
- a. Microsoft Azure
 - b. Alibaba Cloud
 - c. Amazon Web Services
 - d. IBM Cloud
 - e. Google Cloud
 - f. OVHcloud
- 2. Welche der folgenden Aussagen in Bezug auf gehostete OpenShift-Services ist richtig?
- a. Red Hat beteiligt sich nicht an der Unterstützung des OpenShift-Service auf AWS.
 - b. Red Hat beteiligt sich an der Unterstützung gehosteter OpenShift-Services auf jedem Cloud-Anbieter.
 - c. Alle gehosteten OpenShift-Services werden von Red Hat in Rechnung gestellt.
 - d. Red Hat stellt keinen gehosteten OpenShift-Service in Partnerschaft mit Microsoft zur Verfügung.
- 3. Welche zwei der folgenden Aussagen sind Vorteile der Verwendung eines gehosteten OpenShift-Service? (Wählen Sie zwei Antworten aus.)
- a. Kunden erhalten Administratorkonten zum Verwalten der zugrunde liegenden Infrastruktur.
 - b. Kunden müssen weder Zeit noch Ressourcen in Cluster-Upgrades investieren.
 - c. Kunden müssen nicht wissen, wie die Cloud-Anbieterinfrastruktur für die Ausführung von OpenShift konfiguriert wird.
 - d. Kunden können jede Anpassung auf Knotenebene durchführen.

► Lösung

Einführung in gehostete OpenShift-Angebote

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche vier der folgenden Cloud-Anbieter bieten einen gehosteten OpenShift-Service an? (Wählen Sie vier Antworten aus.)
- a. Microsoft Azure
 - b. Alibaba Cloud
 - c. Amazon Web Services
 - d. IBM Cloud
 - e. Google Cloud
 - f. OVHcloud
- 2. Welche der folgenden Aussagen in Bezug auf gehostete OpenShift-Services ist richtig?
- a. Red Hat beteiligt sich nicht an der Unterstützung des OpenShift-Service auf AWS.
 - b. Red Hat beteiligt sich an der Unterstützung gehosteter OpenShift-Services auf jedem Cloud-Anbieter.
 - c. Alle gehosteten OpenShift-Services werden von Red Hat in Rechnung gestellt.
 - d. Red Hat stellt keinen gehosteten OpenShift-Service in Partnerschaft mit Microsoft zur Verfügung.
- 3. Welche zwei der folgenden Aussagen sind Vorteile der Verwendung eines gehosteten OpenShift-Service? (Wählen Sie zwei Antworten aus.)
- a. Kunden erhalten Administratorkonten zum Verwalten der zugrunde liegenden Infrastruktur.
 - b. Kunden müssen weder Zeit noch Ressourcen in Cluster-Upgrades investieren.
 - c. Kunden müssen nicht wissen, wie die Cloud-Anbieterinfrastruktur für die Ausführung von OpenShift konfiguriert wird.
 - d. Kunden können jede Anpassung auf Knotenebene durchführen.

► Quiz

Kapitelwiederholung: Beschreiben des OpenShift-Installationsprozesses

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ **1. Welche der folgenden Aktionen wird bei Verwendung der Installationsmethode „Full-Stack-Automatisierung“ vom OpenShift-Installationsprogramm ausgeführt?**
 - a. Installieren von RHEL 7 auf den Control Plane-Knoten
 - b. Erstellen des Benutzers developer
 - c. Einrichten von Load Balancers
 - d. Konfigurieren eines NFS-Servers für persistenten Storage
- ▶ **2. Welche der folgenden Aktionen wird bei Verwendung der Installationsmethode „Bereits vorhandene Infrastruktur“ vom OpenShift-Installationsprogramm ausgeführt?**
 - a. Starten der Installation von RHCOS auf den Knoten
 - b. Generieren der Ignition-Konfigurationsdateien
 - c. Erstellen aller für die Installation erforderlichen Netzwerkressourcen
 - d. Einrichten eines Load Balancers für den API-Server
- ▶ **3. Welche OpenShift-Installationsmethode wird für die Installation von OpenShift empfohlen, wenn der Administrator volle Flexibilität und Anpassbarkeit der Infrastruktur wünscht?**
 - a. Installationsmethode „Bereits vorhandene Infrastruktur“
 - b. Getrennter Installationsmodus
 - c. Installationsmethode „Full-Stack-Automatisierung“
 - d. Verbundener Installationsmodus
- ▶ **4. Welche Option ist am besten für OpenShift-Benutzer geeignet, die schnellen Zugriff auf einen Cluster wünschen und außerdem Support-, Konfigurations- und Wartungsdienste durch einen zertifizierten Partner benötigen?**
 - a. Selbstverwalteter OpenShift-Cluster unter Verwendung der Installationsmethode „Bereits vorhandene Infrastruktur“
 - b. Selbstverwalteter OpenShift-Cluster unter Verwendung der Installationsmethode „Full-Stack-Automatisierung“
 - c. Gehosteter OpenShift-Cluster
 - d. Selbstverwalteter OpenShift-Cluster unter Verwendung des verbundenen Installationsmodus

► 5. Welche drei Unternehmen bieten gehostete OpenShift-Services an? (Wählen Sie drei

Antworten aus.)

- a. Red Hat
- b. Microsoft (Azure)
- c. VMware
- d. Amazon Web Services
- e. Oracle

► Lösung

Kapitelwiederholung: Beschreiben des OpenShift-Installationsprozesses

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ **1. Welche der folgenden Aktionen wird bei Verwendung der Installationsmethode „Full-Stack-Automatisierung“ vom OpenShift-Installationsprogramm ausgeführt?**
 - a. Installieren von RHEL 7 auf den Control Plane-Knoten
 - b. Erstellen des Benutzers developer
 - c. Einrichten von Load Balancers
 - d. Konfigurieren eines NFS-Servers für persistenten Storage
- ▶ **2. Welche der folgenden Aktionen wird bei Verwendung der Installationsmethode „Bereits vorhandene Infrastruktur“ vom OpenShift-Installationsprogramm ausgeführt?**
 - a. Starten der Installation von RHCOS auf den Knoten
 - b. Generieren der Ignition-Konfigurationsdateien
 - c. Erstellen aller für die Installation erforderlichen Netzwerkressourcen
 - d. Einrichten eines Load Balancers für den API-Server
- ▶ **3. Welche OpenShift-Installationsmethode wird für die Installation von OpenShift empfohlen, wenn der Administrator volle Flexibilität und Anpassbarkeit der Infrastruktur wünscht?**
 - a. Installationsmethode „Bereits vorhandene Infrastruktur“
 - b. Getrennter Installationsmodus
 - c. Installationsmethode „Full-Stack-Automatisierung“
 - d. Verbundener Installationsmodus
- ▶ **4. Welche Option ist am besten für OpenShift-Benutzer geeignet, die schnellen Zugriff auf einen Cluster wünschen und außerdem Support-, Konfigurations- und Wartungsdienste durch einen zertifizierten Partner benötigen?**
 - a. Selbstverwalteter OpenShift-Cluster unter Verwendung der Installationsmethode „Bereits vorhandene Infrastruktur“
 - b. Selbstverwalteter OpenShift-Cluster unter Verwendung der Installationsmethode „Full-Stack-Automatisierung“
 - c. Gehosteter OpenShift-Cluster
 - d. Selbstverwalteter OpenShift-Cluster unter Verwendung des verbundenen Installationsmodus

► 5. Welche drei Unternehmen bieten gehostete OpenShift-Services an? (Wählen Sie drei

Antworten aus.)

- a. Red Hat
- b. Microsoft (Azure)
- c. VMware
- d. Amazon Web Services
- e. Oracle

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Details des OpenShift-Installationsprozesses
- Die OpenShift-Installationsmethoden:
 - Verwenden Sie die Installationsmethode „Full-Stack-Automatisierung“, um den Cluster in einer Infrastruktur bereitzustellen, die vom Installationsprogramm bereitgestellt und vom Cluster verwaltet wird.
 - Verwenden Sie die Installationsmethode „Bereits vorhandene Infrastruktur“, um einen Cluster in einer Infrastruktur bereitzustellen, die von Administratoren selbst vorbereitet und verwaltet wird.
- Die OpenShift-Installationsmodi:
 - Verwenden Sie den verbundenen Installationsmodus, wenn die Cluster-Knoten Zugriff auf das Internet haben.
 - Verwenden Sie den getrennten Modus, wenn sich die Cluster-Knoten in einem eingeschränkten Netzwerk befinden.
- Die wichtigsten Ressourcen, die vom OpenShift-Installationsprogramm erstellt werden:
 - Die Installationskonfigurationsdatei `install-config.yaml`
 - Die Kubernetes-Manifeste
 - Die Ignition-Konfigurationsdateien
- Die Voraussetzungen, die die Infrastrukturumgebung erfüllen muss, damit OpenShift installiert werden kann
- Die Schritte, mit den überprüft werden kann, ob ein OpenShift-Cluster fehlerfrei und für die Day 2-Aufgaben zum Onboarding von Benutzern und Anwendungen bereit ist.
- Wie die Konfigurationsdatei `install-config.yaml` bearbeitet werden muss, um die Cluster-Installation anzupassen
- Wie die Ausführung, Überwachung und Fehlerbehebung für die einzelnen Phasen des OpenShift-Installationsprozesses zu erfolgen hat
- Wie Informationen von einem OpenShift-Cluster erfasst werden
- Wie Red Hat gehostete OpenShift-Angebote in Partnerschaft mit anderen Anbietern bereitstellt

Kapitel 2

Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

Ziel

Bereitstellen von OpenShift-Clustern auf IaaS-Cloud-Anbieter-Plattformen mit allgemeinen Anpassungen unter Verwendung der Installationsmethode „Full-Stack-Automatisierung“

Ziele

- Beschreiben der Architektur und des Workflows zur Installation von OpenShift auf einer IaaS-Cloud-Anbieter-Plattform mit der Full-Stack-Automatisierungsmethode
- Erfüllen der Voraussetzungen für die Installation von OpenShift auf Amazon Web Services (AWS) mit Full-Stack-Automatisierung
- Installieren von OpenShift auf Amazon Web Services (AWS) unter Verwendung von Full-Stack-Automatisierung mit allgemeinen Anpassungen
- Bewerten des Erfolgs einer Installation von OpenShift auf AWS

Abschnitte

- Einführung in die OpenShift-Installation mit Full-Stack-Automatisierung auf einer Cloud-Anbieter-Plattform (und Test)
- Beschreiben der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode (und Test)
- Demonstrieren der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode (und Test)
- Überprüfen der Installation von OpenShift auf AWS (und Test)
- Kapitelwiederholung: Installieren von OpenShift auf einer Cloud-Anbieter-Plattform (Test)

Einführung in die OpenShift-Installation mit Full-Stack-Automatisierung auf einer Cloud-Anbieter-Plattform

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die Architektur und den Workflow der Installation von OpenShift auf einer IaaS-Cloud-Anbieter-Plattform mit der Full-Stack-Automatisierungsmethode beschreiben können.

Einführung in OpenShift-Installation auf einer Cloud-Anbieter-Plattform

Wenn Sie OpenShift unter Verwendung der Full-Stack-Automatisierungsmethode auf einer unterstützten IaaS-Cloud-Anbieter-Plattform installieren, stellt das Installationsprogramm automatisch die erforderliche Infrastruktur und die benötigten Ressourcen bereit.

- Betriebssystem-Images
- VMs oder Cloud-Instanzen
- Load Balancer
- Storage
- Netzwerkfunktionen

Mit dieser Methode können Administratoren OpenShift mit minimalem Eingriff ihrerseits auf den folgenden unterstützten IaaS-Cloud-Anbieter-Plattformen installieren:

- Amazon Web Services (AWS)
- Red Hat OpenStack Platform (RHOSP)
- Microsoft Azure (MS Azure)
- Google Cloud Platform (GCP)

Gründe für die Installation von OpenShift auf einer Cloud-Anbieter-Plattform mit Full-Stack-Automatisierung

Für unterstützte Cloud-Anbieter empfiehlt Red Hat aus den folgenden Gründen die Verwendung der Installationsmethode „Full-Stack-Automatisierung“:

- Administratoren installieren OpenShift mit minimalem manuellen Eingriff mit einer „eigenwilligen“ Methode.
- Das OpenShift-Installationsprogramm erstellt automatisch die erforderlichen Ressourcen.
- Das OpenShift-Installationsprogramm integriert die OpenShift Machine API-Ressource vollständig in die Cloud-Anbieter-Services. Das Installationsprogramm erstellt die OpenShift-Ressource **MachineSets**, sodass die automatische Knotenbereitstellung und die automatische Cluster-Skalierung direkt nach Abschluss der Installation verwendet werden können.

- Red Hat unterstützt die OpenShift-Installation auf den relevantesten Cloud-Anbieter-Plattformen.
- Red Hat hat sich mit den relevantesten Cloud-Anbietern zusammengetan, um die Qualität der Installationsmethode „Full-Stack-Automatisierung“ sicherzustellen.

**Anmerkung**

Administratoren können OpenShift auch mithilfe der Methode „Bereits vorhandene Infrastruktur“ auf einer Cloud-Anbieter-Plattform installieren. Bei diesem Ansatz müssen Administratoren die erforderlichen Cloud-Ressourcen vor der Installation von OpenShift manuell erstellen.

Beschreiben von OpenShift-Installationen auf einer Cloud-Anbieter-Plattform

In der folgenden Liste werden die wichtigsten Schritte zur Installation von OpenShift auf einer Cloud-Anbieter-Plattform unter Verwendung der Full-Stack-Automatisierungsmethode zusammengefasst. Dieser Prozess wird im Abschnitt *Einführung in OpenShift-Installationsmethoden* ausführlicher beschrieben.

1. Erfüllen der in der Lektion *Beschreiben der OpenShift-Installationsvoraussetzungen* aufgeführten allgemeinen Voraussetzungen und Voraussetzungen für die Full-Stack-Automatisierung
2. Überprüfen der Infrastrukturvoraussetzungen, der Berechtigungen des Cloud-Benutzerkontos und der Kontingente
3. Ausführen des OpenShift-Installationsprogramms auf dem Bastion-Host
4. Das OpenShift-Installationsprogramm erstellt die erforderlichen Cloud-Ressourcen und startet die Cluster-Installation.
5. Überwachen des OpenShift-Installationsprozesses
6. Überprüfen der Cluster-Integrität nach Abschluss der OpenShift-Installation
7. Durchführen von Aufgaben nach der Installation (optional)

Der Schritt der Cluster-Installation besteht aus den folgenden Phasen:

Phase der Ressourcenerstellung

Zunächst erstellt und konfiguriert das OpenShift-Installationsprogramm die folgenden Cloud-Ressourcen:

- Cluster-Netzwerk
- Cluster-Netzwerkservices
- Cluster-Netzwerk-Gateway
- Cluster-Load Balancer
- Cluster-Storage
- Bootstrap-Knoten

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

- Control Plane-Knoten

Das folgende Diagramm zeigt die Beziehung zwischen OpenShift-Ressourcen und den Cloud-Anbieter-Services, die zum Erstellen der Ressourcen verwendet wurden.

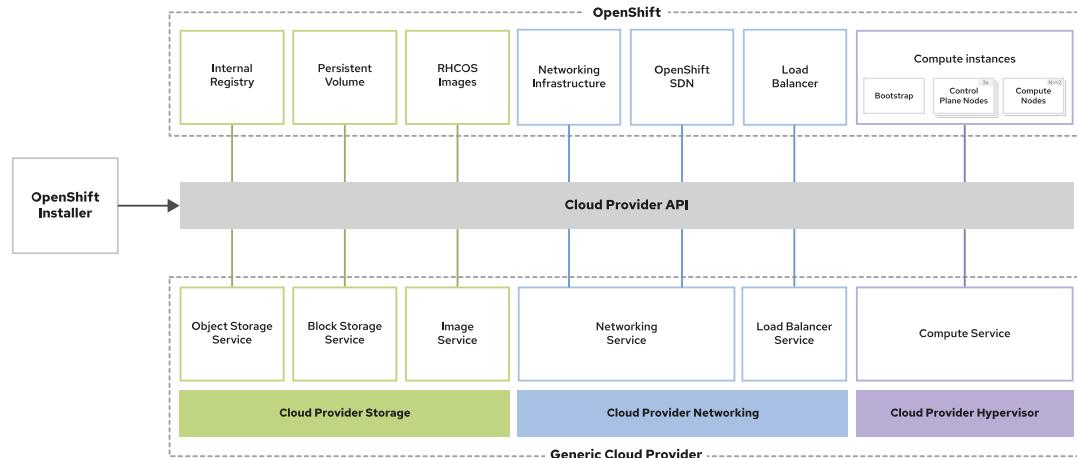


Abbildung 2.1: Ressourcen der OpenShift-Installation mit Full-Stack-Automatisierung auf einer generischen Cloud-Anbieter-Plattform

Bootstrap-Phase

In dieser Phase geschieht Folgendes:

- Der Bootstrap-Knoten startet die Kubernetes-API.
- Die Control Plane-Knoten versuchen, ihre Ignition-Konfigurationsdateien von der Kubernetes-API abzurufen.
- Wenn die Kubernetes-API auf dem Bootstrap-Knoten verfügbar wird, rufen die Control Plane-Knoten ihre Ignition-Konfigurationsdateien erfolgreich ab und beenden die Installation.

Im folgenden Diagramm wird die Architektur einer OpenShift-Bereitstellung auf einer generischen Cloud-Anbieter-Plattform in der Bootstrap-Phase erläutert.

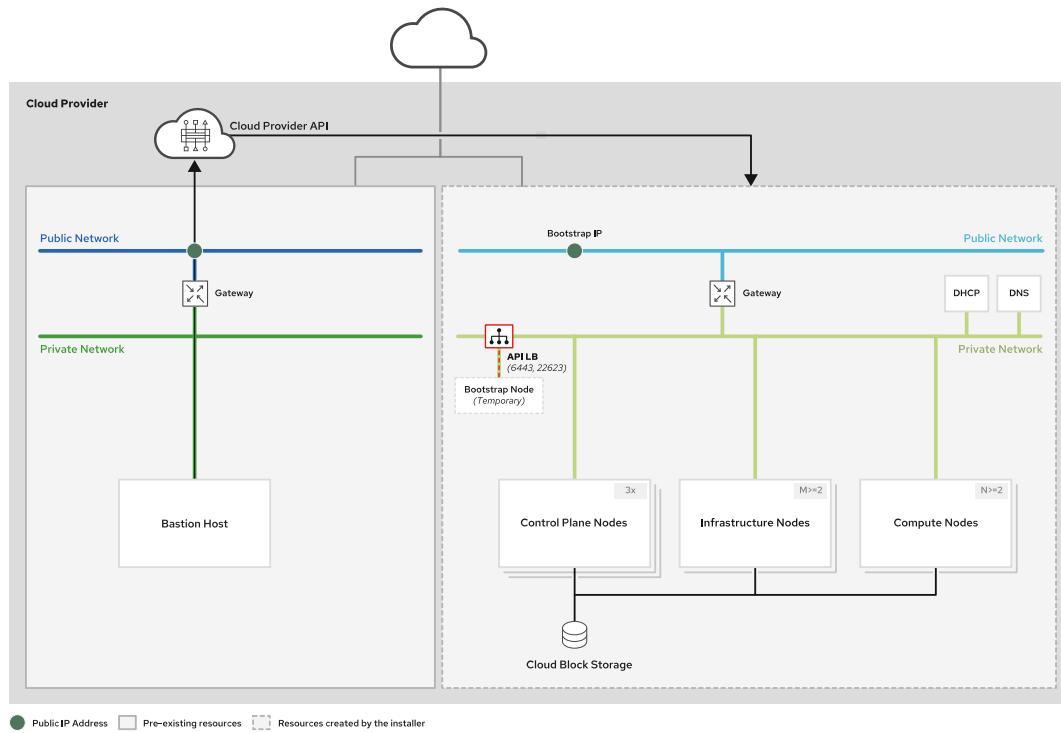


Abbildung 2.2: OpenShift-Installation mit Full-Stack-Automatisierung auf einer generischen Cloud-Anbieter-Plattform – Bootstrap-Phase

Produktions-Control Plane-Phase

Sobald der Bootstrap-Knoten die temporäre Control Plane auf die Control Plane-Knoten übertragen hat, wird sie nicht mehr benötigt.

In dieser Phase geschieht Folgendes:

- Die Produktions-Control Plane wird auf den Control Plane-Knoten ausgeführt.
- Der OpenShift-API-Load Balancer (OpenShift-API-LB) verwendet die Control Plane-Knoten als Back-End-Pool-Mitglieder.
- Das Installationsprogramm erstellt und installiert die Server-Knoten. Die Server-Knoten verbinden sich mit dem Cluster.
- Der OpenShift-Application Ingress-Load Balancer (OpenShift-App-Ingress-LB) verwendet die Server-Knoten als Back-End-Pool-Mitglieder.
- Das Installationsprogramm veröffentlicht die IP-Adressen der internen Load Balancer im öffentlichen Netzwerk des Cloud-Anbieters (optional).
- Gleichzeitig installiert der Cluster Version Operator (CVO), der auf der Produktions-Control Plane ausgeführt wird, die Operatoren, die den Cluster erstellen, und beendet dann die Installation.
- Das Installationsprogramm konfiguriert die interne OpenShift-Registry für die Verwendung des Cloud-Objekt-Storage.
- Das Installationsprogramm konfiguriert einen dynamischen Storage-Anbieter, um mithilfe des Cloud-Storage-Service persistenten Storage für die containerisierten Anwendungen bereitzustellen.

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

- Wenn die Cloud-Anbieter-Services die OpenShift Machine API-Integration unterstützen, konfiguriert das Installationsprogramm die Knotenbereitstellung und die automatische Cluster-Skalierung.

Das folgende Diagramm zeigt die Architektur einer OpenShift-Bereitstellung auf einer generischen Cloud-Anbieter-Plattform nach Abschluss der Installation.

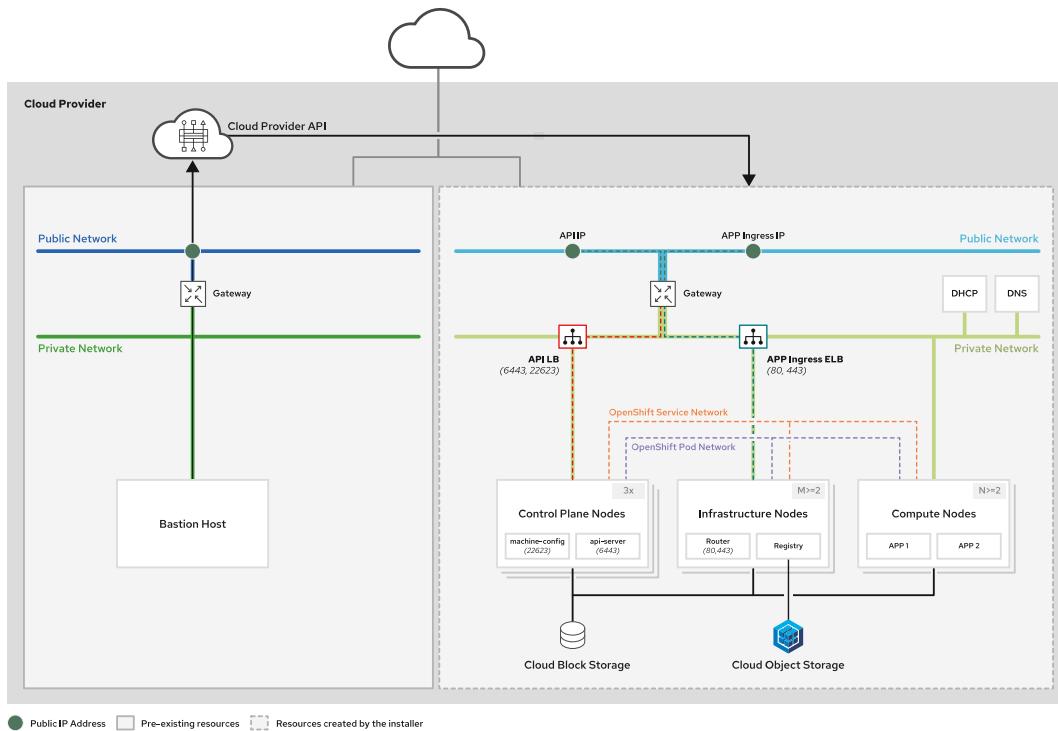


Abbildung 2.3: OpenShift-Installation mit Full-Stack-Automatisierung auf einer generischen Cloud-Anbieter-Plattform – letzte Phase

Während der OpenShift-Installation auf einer generischen Cloud-Anbieter-Plattform:

- Der Bastion-Host benötigt eine Netzwerkverbindung zur Cloud-Anbieter-API, um OpenShift zu installieren, und zur OpenShift-API, um OpenShift nach der Installation zu verwalten. Es ist nicht obligatorisch (wird jedoch empfohlen), den Bastion-Host auf derselben Cloud-Anbieter-Plattform zu installieren, die für die Installation von OpenShift verwendet wird.
- Bei der OpenShift-Installation werden keine Infrastrukturknoten erstellt, sondern nur Server-Knoten. Bei den Infrastrukturknoten handelt es sich um Server-Knoten, auf denen Router- oder Registry-Pods ausgeführt werden.
- Der OpenShift-Installationsprozess erstellt die Cloud-Ressource „APP Ingress-LB“, die alle Server-Knoten als Back-End-Pool-Mitglieder für die Ports 80/TCP und 443/TCP enthält. Nur Server-Knoten, auf denen Router-Pods ausgeführt werden, die diese Ports überwachen, bestehen die LB-Integritätsprüfung. Daher sendet der APP Ingress-LB Netzwerkdatenverkehr nur an die Server-Knoten, auf denen die Router-Pods ausgeführt werden.
- Die DHCP-Konfiguration stellt auch die NTP-Konfiguration bereit.
- Standardmäßig verwendet der auf jedem Cluster-Knoten ausgeführte systemd-Service `chronyd` den öffentlichen NTP-Pool `rhel.pool.ntp.org` zum Synchronisieren der

Systemuhr Nach der Installation können Administratoren die Cluster-Knoten mit dem Machine Config-Operator für die Verwendung eines lokalen NTP-Servers konfigurieren, wie im Abschnitt *Configuring chrony time service* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/post-installation_configuration erläutert.

Vergleichen der Terminologie von IaaS-Cloud-Anbietern

Jeder Cloud-Anbieter verwendet eine bestimmte Terminologie, um verschiedene Services zu referenzieren, die während der Installation verwendet werden.

In der folgenden Tabelle wird die unterschiedliche Cloud-Terminologie für die gängigsten Cloud-Anbieter aufgeführt.

Cloud-Anbieter-Services, die während einer OpenShift-Installation mit der Full-Stack-Automatisierungsmethode verwendet werden

OpenShift-Ressourcen	AWS-Anbieter	RHOSP-Anbieter	Azure-Anbieter	GCP-Anbieter
Interner Registry-Storage	Amazon S3	Swift	Azure Blob	Google Cloud-Objekt-Storage
Persistentes Volume	Amazon EBS	Cinder	Azure Disk	Google Cloud-Blockspeicher
RHCOS-Images	Amazon AMI	Glance	Azure-Images (VHD)	Google Compute Engine-Images
Netzwerkfunktionen	Amazon VPC	Neutron	Azure VNet	Google VPC
SDN/OVN	Amazon VPC	Neutron + Kurry	Azure VNet	Google VPC
Load Balancer	Amazon ELB	Statische Pods/Octavia LBaaS	Azure LB	Google-Netzwerk-Suites
DNS	Amazon Route53	Extern	Azure DNS	Google Cloud-DNS
Server-Instanzen	Amazon EC2	Nova	Azure Compute	Google Compute Engine

Installieren von OpenShift mit Full-Stack-Automatisierung auf Microsoft Azure

Microsoft Azure (MS Azure) wird als Option bei der Installation von OpenShift bei einem Cloud-Anbieter mit der Full-Stack-Automatisierungsmethode unterstützt. Mit dieser Installationsmethode können Administratoren einen OpenShift-Cluster mit minimalem Eingriff auf „eigenwillige“ Weise auf MS Azure installieren.

Voraussetzungen für die Installation von OpenShift auf MS Azure

Vor der Installation von OpenShift auf MS Azure müssen Administratoren die folgenden Aktionen ausführen:

- Überprüfen der in der Lektion *Beschreiben der OpenShift-Installationsvoraussetzungen* beschriebenen allgemeinen Voraussetzungen und Voraussetzungen für die Full-Stack-Automatisierung.
- Erstellen einer öffentlichen DNS-gehosteten Zone für den Cluster im Azure DNS-Service, sofern sie noch nicht vorhanden ist. Diese Zone muss für die Domain autoritativ sein.
- Verwenden eines Azure IAM-Benutzerkontos (Identity and Access Management, Identitäts- und Zugriffsverwaltung) zum Ausführen der Installation. Dieses Benutzerkonto muss über die Rollenberechtigung `User Access Administrator` verfügen.
- Sicherstellen, dass das Azure IAM-Benutzerkonto die zum Installieren von OpenShift erforderlichen Kontingent- und Berechtigungsvoraussetzungen erfüllt.
- Erstellen eines Dienstprinzipals, der das OpenShift-Installationsprogramm in Azure Resource Manager repräsentiert.
- Überprüfen der unterstützten Azure-Regionen. Das OpenShift-Installationsprogramm generiert auf Basis des MS Azure-Kontoabonnements dynamisch eine Liste der verfügbaren Azure-Regionen.
- Überprüfen der Einschränkungen des Azure-Ressourcennamens.



Anmerkung

- Weitere Informationen finden Sie im Abschnitt *Configuring an Azure account* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_azure

Dimensionieren von OpenShift-Installationen auf MS Azure

Administratoren müssen die Kontingente der MS Azure-Kontoressourcen überprüfen, um die folgenden Mindestanforderungen zu gewährleisten.

Azure-Beschränkungen, die sich auf die Installation und Ausführung von OpenShift-Clustern auswirken können

Ressource	Erforderlich	Azure-Standardlimit
vCPU	40	20 pro Region
VNet	1	1.000 pro Region
Netzwerkschnittstellen	6	65.536
Netzwerk-Sicherheitsgruppen	2	5.000
Load Balancer im Netzwerk	3	1.000 pro Region

Ressource	Erforderlich	Azure-Standardlimit
Öffentliche IP-Adressen	3	-
Private IP-Adressen	7	-



Anmerkung

Die in dieser Tabelle beschriebenen OpenShift-Ressourcenanforderungen sind für die Installation eines kleinen OpenShift-Clusters ausreichend. Die auf dem OpenShift-Cluster ausgeführten Anwendungs-Workloads können sich auf die tatsächlichen Anforderungen auswirken.



Anmerkung

Da für die OpenShift-Installation mindestens 40 vCPUs erforderlich sind und das Azure-Standardlimit pro Region 20 beträgt, müssen Sie dieses Limit wahrscheinlich erhöhen.

- Weitere Informationen finden Sie im Leitfaden *Increasing Azure account limits* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_azure

Anpassen von OpenShift-Installationen auf MS Azure

Für eine OpenShift-Installation auf Azure werden standardmäßig die folgenden VM-Größen (virtuelle Rechner) verwendet: Standard_D8s_v3 für Control Plane-Knoten und Standard_D4s_v3 für Server-Knoten. Diese Azure VM-Standardgrößen sind für Proof of Concept (PoC)- und andere Nichtproduktionsszenarien geeignet.

Größen von MS Azure-VMs der Dsv3-Serie (Referenz)

Größe	vCPU	RAM (GiB)	Max. SSD-IOPS	Max. SSD-Durchsatz
Standard_D4s_v3	4	16	8.000	100 MiB/s
Standard_D8s_v3	8	32	16.000	200 MiB/s
Standard_D16s_v3	16	64	32.000	400 MiB/s
Standard_D32s_v3	32	128	64.000	800 MiB/s

Verwenden Sie für Produktions-Cluster eine größere VM-Größe als die Standardgröße Standard_D8s_v3 für die Control Plane-Knoten, um eine angemessene Storage-Leistung für den etcd-Cluster sicherzustellen, der auf ihnen ausgeführt wird.

Verwenden Sie für Produktions-Cluster außerdem eine größere VM-Größe als die Standardgröße Standard_D4s_v3 für die Server-Knoten, um eine angemessene Leistung für die Cluster-Infrastruktur services sicherzustellen, die auf ihnen ausgeführt werden.



Anmerkung

Weitere Informationen finden Sie im Abschnitt *Installing a cluster on Azure with customizations* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_azure

Weitere Informationen zu OpenShift-Installationen auf MS Azure finden Sie hier:
<https://www.openshift.com/blog/openshift-4-2-on-azure-preview>

Beschreiben von OpenShift-Installationen auf MS Azure

Die zum Ausführen einer OpenShift-Installation auf einer generischen Cloud-Provider-Plattform erforderlichen Schritte werden im Abschnitt *Beschreiben von OpenShift-Installationen auf einer Cloud-Anbieter-Plattform* in dieser Lektion erläutert. In diesem Abschnitt werden die Besonderheiten der OpenShift-Installation auf der Cloud-Anbieter-Plattform MS Azure in den einzelnen Installationsphasen beschrieben.

Phase der Ressourcenerstellung

Zu Beginn der Installation erstellt und konfiguriert das OpenShift-Installationsprogramm die erforderlichen Cloud-Ressourcen.

Das OpenShift-Installationsprogramm verwendet Folgendes:

- Den Service „Azure Virtual Network (VNet)“ zum Erstellen des Cluster-Netzwerks und der Cluster-Netzwerkservices
- Den Service „Azure NAT Gateway“ zum Erstellen des Cluster-Netzwerk-Gateways
- Den Service „Azure Load Balancer (LB)“ zum Erstellen der Cluster-Load Balancer
- Den Service „Azure Compute“ zum Erstellen der Cluster-Server-Knoten
- Die Services „Azure Disk“ und „Azure Blob“ für den Cluster-Storage

Das folgende Diagramm zeigt die Beziehung zwischen OpenShift-Ressourcen und den MS Azure-Services, die zum Erstellen der Ressourcen verwendet werden.

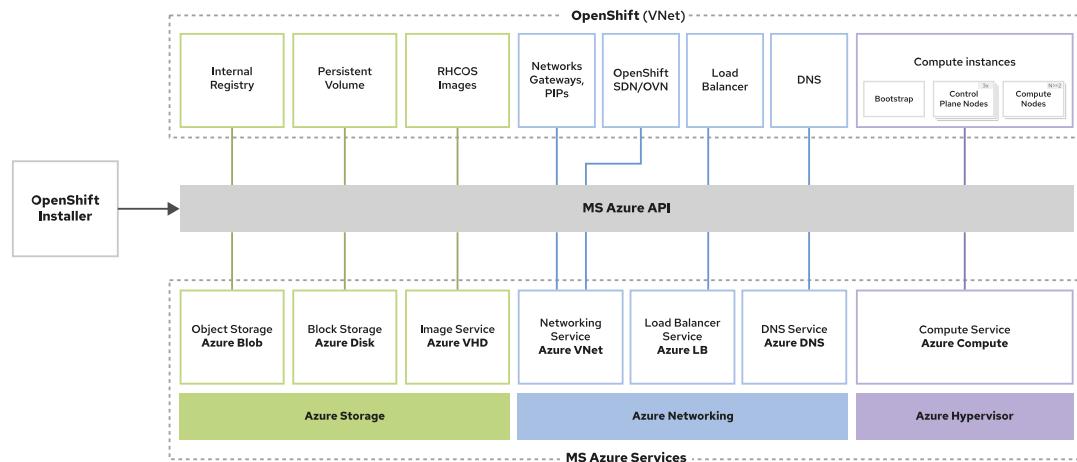


Abbildung 2.4: Ressourcen der OpenShift-Installation mit Full-Stack-Automatisierung auf MS Azure

Bootstrap-Phase

In dieser Phase werden die Kubernetes-API und die temporäre Control Plane auf dem Bootstrap-Knoten ausgeführt. Das OpenShift-Installationsprogramm beginnt mit der Installation der Cluster-Knoten.

Im folgenden Diagramm wird die Architektur einer OpenShift-Bereitstellung auf MS Azure in der Bootstrap-Phase erläutert.

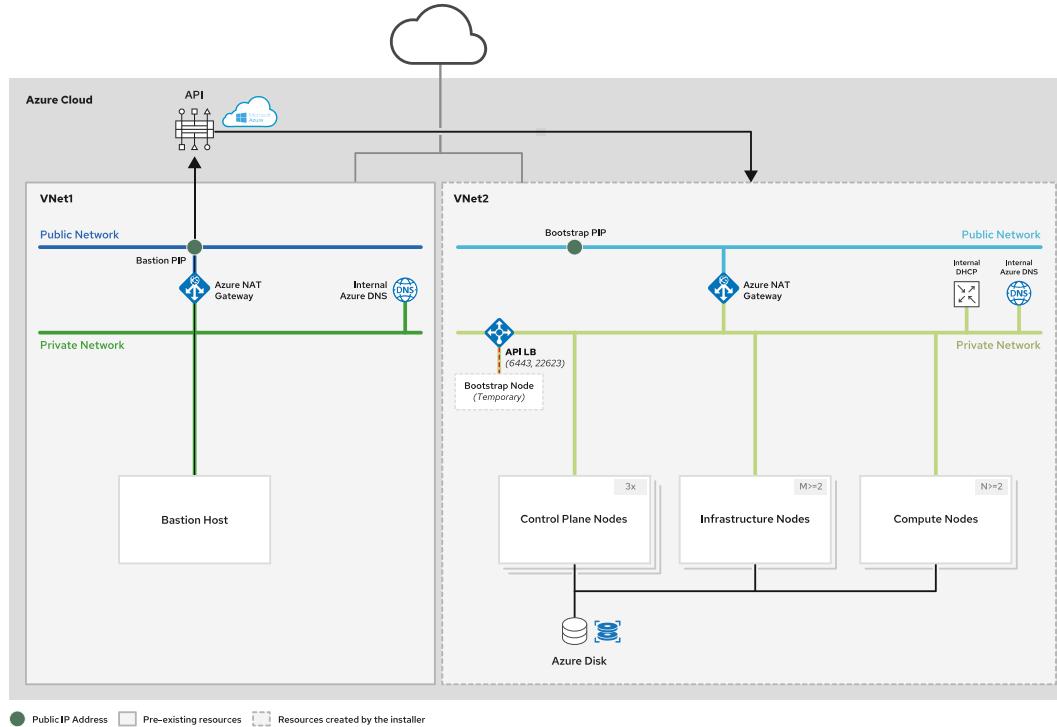


Abbildung 2.5: OpenShift-Installation mit Full-Stack-Automatisierung auf MS Azure – Bootstrap-Phase

Produktions-Control Plane-Phase

In dieser Phase werden die Kubernetes-API und die Produktions-Control Plane auf den Control Plane-Knoten ausgeführt. Der Cluster Version Operator (CVO), der auf der Produktions-Control Plane ausgeführt wird, installiert alle Operatoren, die den Cluster erstellen, und beendet anschließend die Installation.

Das OpenShift-Installationsprogramm führt Folgendes aus:

- Veröffentlichen der IP-Adressen des OpenShift-API-LB und des APP Ingress-LB als öffentliche IP-Adressen im öffentlichen Azure-Netzwerk

Bei der Installation von OpenShift auf MS Azure veröffentlicht das Installationsprogramm die OpenShift-Load Balancer-Endpunkte unter Verwendung öffentlicher IP-Adressen. Dieser Schritt ist bei Installation von OpenShift auf anderen Cloud-Anbietern wie RHOSP oder AWS optional, bei Installation von OpenShift auf dem Cloud-Anbieter MS Azure ist er jedoch obligatorisch.

Um zu vermeiden, dass bei der Installation von OpenShift auf MS Azure öffentliche IP-Adressen für die Cluster-Load Balancer-Endpunkte verwendet werden, müssen Administratoren den Cluster so konfigurieren, dass *benutzerdefiniertes ausgehendes Routing* verwendet wird, wie im Abschnitt *Installing a private cluster on Azure* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter <https://>

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_azure erläutert.

- Konfigurieren der internen OpenShift-Registry für die Verwendung von Azure Blob-Objekt-Storage
- Konfigurieren eines dynamischen Storage-Anbieters zur Bereitstellung von persistentem Storage für die containerisierten Anwendungen mithilfe des Azure Disk-Storage-Service

Das folgende Diagramm beschreibt die Architektur einer OpenShift-Bereitstellung auf MS Azure nach Abschluss der Installation.

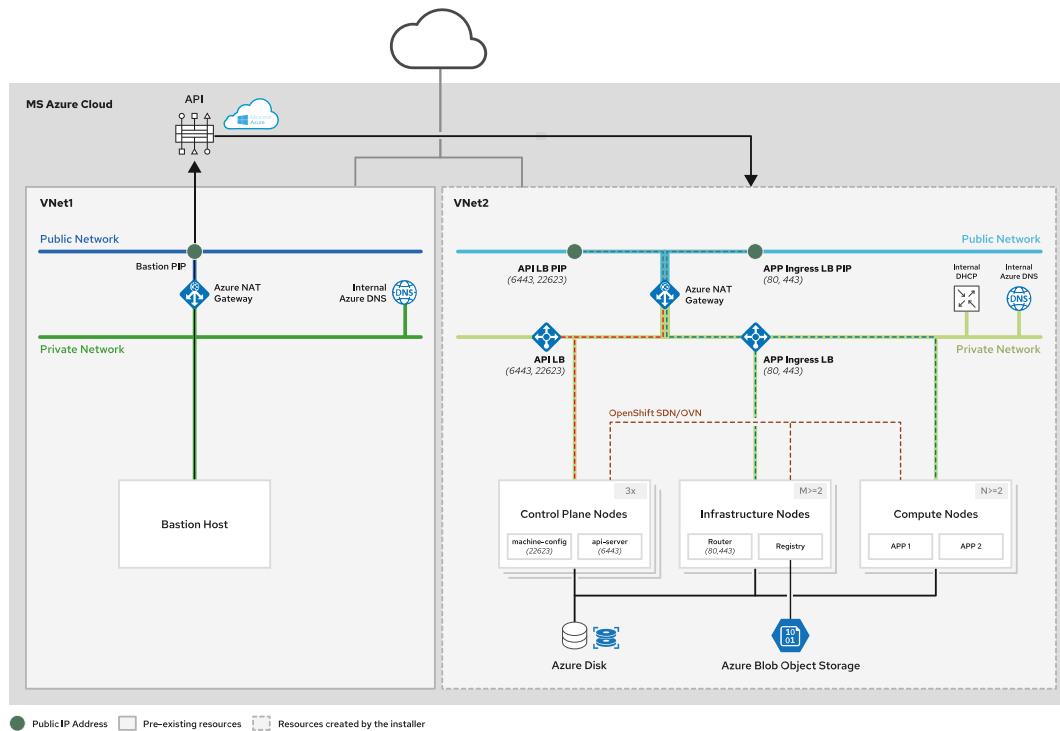


Abbildung 2.6: OpenShift-Installation mit Full-Stack-Automatisierung auf MS Azure – letzte Phase

Installieren von OpenShift mit Full-Stack-Automatisierung auf MS Azure

Videoergebnis

Installieren eines minimalen OpenShift-Clusters auf MS Azure unter Verwendung der Standardeinstellungen für die Full-Stack-Automatisierung

Installieren von OpenShift mit Full-Stack-Automatisierung auf Red Hat OpenStack Platform

Bei Installation von OpenShift auf Red Hat OpenStack Platform (RHOSP) empfiehlt Red Hat die Verwendung von Kuryr als SDN. Kuryr ist eine Container Network Interface (CNI)-Plug-in-Lösung. Sie verwendet die RHOSP-Services Neutron und Octavia zur Bereitstellung von Netzwerkfunktionen für Pods und Services.

Die Verwendung des Kuryr-SDN für OpenShift-Bereitstellungen auf RHOSP bietet die folgenden Vorteile:

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

- Verbessert die Netzwerkleistung, indem OpenShift-Pods direkt in das RHOSP-SDN eingebunden werden
- Bietet interne Konnektivität zwischen Pods und virtuellen RHOSP-Instanzen
- Vermeidet Doppelkapselung des Netzwerkdatenverkehrs

Voraussetzungen für die Installation von OpenShift auf RHOSP mit Kuryr

Vor der Installation von OpenShift auf RHOSP mit Kuryr müssen Administratoren die folgenden Aktionen ausführen:

- Überprüfen der in der Lektion *Beschreiben der OpenShift-Installationsvoraussetzungen* beschriebenen allgemeinen Voraussetzungen und Voraussetzungen für die Full-Stack-Automatisierung.
- Überprüfen Sie die Versionskompatibilität zwischen OpenShift und RHOSP anhand der folgenden Dokumentationsseite [https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/architecture/index#architecture-installation].
 - Verwenden Sie für die Installation von OpenShift 4.6 auf RHOSP mit Kuryr RHOSP Version 16.1.
- Stellen Sie sicher, dass die Limits und Berechtigungen von RHOSP-Benutzerkonten für die Installation von OpenShift ausreichend sind.
- Stellen Sie sicher, dass auf RHOSP der Cinder-Block-Storage und die Swift-Objekt-Storage-Services (optional, aber empfohlen) installiert sind.
- Stellen Sie sicher, dass der RHOSP-Metadatenservice aktiviert ist.
- Erstellen Sie die DNS-Einträge für die Floating-IP-Adressen der API- und Ingress-Load Balancer (optional).
- Bei Verwendung von Kuryr müssen Administratoren die Kontingente erhöhen, um den RHOSP-Ressourcenbedarf von Pods, Services, Namespaces und Netzwerkrichtlinien zu erfüllen.

```
[user@demo ~]$ sudo openstack quota set --secgroups 250 --secgroup-rules 1000 \
> --ports 1500 --subnets 250 --networks 250 <project>
```

Dimensionieren von OpenShift-Installationen auf RHOSP mit Kuryr

Administratoren müssen die Kontingente der RHOSP-Kontoressourcen überprüfen, um die folgenden Mindestanforderungen zu gewährleisten:

Empfohlene Ressourcen für die Installation eines Standard-OpenShift-Clusters auf RHOSP mit Kuryr

Ressource	Wert
Floating-IP-Adressen	3 – plus der erwarteten Anzahl an Services des Typs LoadBalancer
Ports	1.500 – 1 pro Pod erforderlich

Ressource	Wert
Router	1
Subnetze	250 – 1 pro Namespace/Projekt erforderlich
Netzwerke	250 – 1 pro Namespace/Projekt erforderlich
RAM	112 GB
vCPUs	28
Volume-Storage	275 GB
Instanzen	7
Sicherheitsgruppen	250 – 1 pro Service und NetworkPolicy erforderlich
Sicherheitsgruppenregeln	1.000
Load Balancer	100 – 1 Service erforderlich
Load Balancer-Listener	500 – 1 pro von Service verwendetem Port erforderlich
Load Balancer-Pools	500 – 1 pro von Service verwendetem Port erforderlich



Anmerkung

Die in dieser Tabelle beschriebenen OpenShift-Ressourcenanforderungen sind für die Installation eines kleinen OpenShift-Clusters ausreichend. Die auf dem OpenShift-Cluster ausgeführten Anwendungs-Workloads können sich auf die tatsächlichen Anforderungen auswirken.

Anpassen von OpenShift-Installationen auf RHOSP mit Kuryr

Der für eine OpenShift-Installation auf RHOSP verwendete RHOSP-Instanz-Flavor (4 vCPUs und 8 GiB RAM) ist standardmäßig für Proof of Concept (PoC)- und andere Nichtproduktionsszenarien geeignet. Verwenden Sie für Produktions-Cluster einen Instanz-Flavor mit mindestens 32 GB RAM, 8 vCPUs und 100 GB Speicherplatz für die Installation der Cluster-Knoten.

Vor der Installation von OpenShift auf RHOSP können Sie auch benutzerdefinierte Instanz-Flavors erstellen:

```
[user@demo ~]$ sudo openstack flavor create --disk 200 \
> --ram 32384 --vcpu 8 --public m1.master
[user@demo ~]$ sudo openstack flavor create --disk 200 \
> --ram 64384 --vcpu 16 --public m1.worker
```

Nachdem sie erstellt wurden, können Sie die benutzerdefinierten Flavors in die Datei `install-config.yaml` einschließen und den OpenShift-Cluster installieren.



Anmerkung

Weitere Informationen finden Sie im Abschnitt *Installing a cluster on OpenStack with customizations* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_openstack

Beschreiben von OpenShift-Installationen auf RHOSP mit Kuryr

Die zum Ausführen einer OpenShift-Installation auf RHOSP erforderlichen Schritte werden im Abschnitt *Beschreiben von OpenShift-Installationen auf einer Cloud-Anbieter-Plattform* in diesem Dokument erläutert. Diese Schritte gelten für die OpenShift-Installation auf RHOSP.

In diesem Abschnitt werden die Besonderheiten der OpenShift-Installation auf RHOSP in den einzelnen Installationsphasen beschrieben.

Phase der Ressourcenerstellung

Zu Beginn der Installation erstellt und konfiguriert das OpenShift-Installationsprogramm die erforderlichen Cloud-Ressourcen.

Das OpenShift-Installationsprogramm verwendet Folgendes:

- Den RHOSP-Service "Neutron" zum Erstellen des Cluster-Netzwerks und der Cluster-Netzwerkselemente
- Den RHOSP-Service "External Router" zum Erstellen des Cluster-Netzwerk-Gateways
- Statische Pods, die auf Cluster-Knoten ausgeführt werden, zum Erstellen der Cluster-Load Balancer
- Den RHOSP-Service "Nova" zum Erstellen der Cluster-Server-Knoten
- Die RHOSP-Services "Cinder" und "Swift" für den Cluster-Storage

Das folgende Diagramm zeigt die Beziehung zwischen OpenShift-Ressourcen und den RHOSP-Services, die zum Erstellen der Ressourcen verwendet werden.

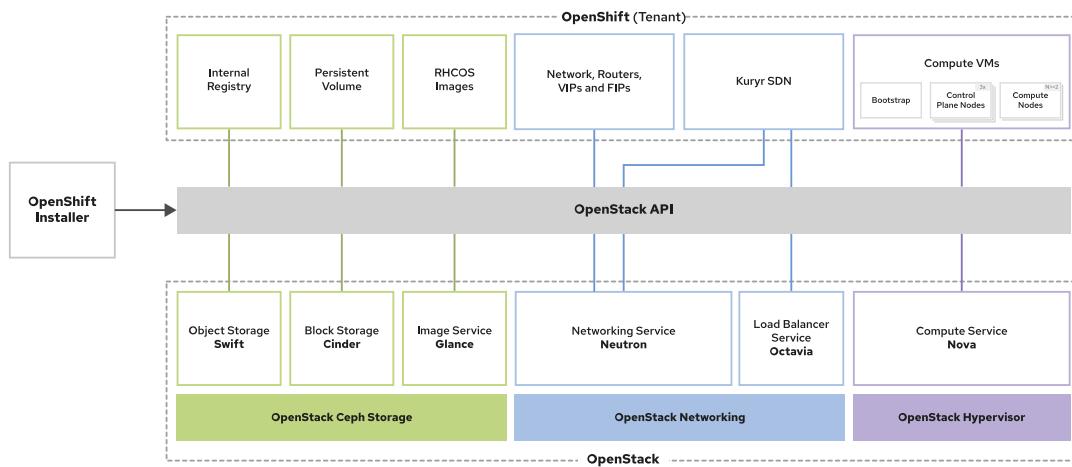


Abbildung 2.7: Ressourcen der OpenShift-Installation mit Full-Stack-Automatisierung auf RHOSP

Bootstrap-Phase

In dieser Phase werden die Kubernetes-API und die temporäre Control Plane auf dem Bootstrap-Knoten ausgeführt. Das OpenShift-Installationsprogramm beginnt mit der Installation der Cluster-Knoten.

Der Bootstrap-Knoten hostet die Remote-Ressourcen, die zum Booten von Control Plane-Knoten erforderlich sind (Ignition-Konfigurationsdateien), auf dem Machine Configuration-Server (MCS). Er führt auch eine einzelne Instanz des Clusters etcd aus.

Im folgenden Diagramm wird die Architektur einer OpenShift-Bereitstellung auf RHOSP in der Bootstrap-Phase erläutert.

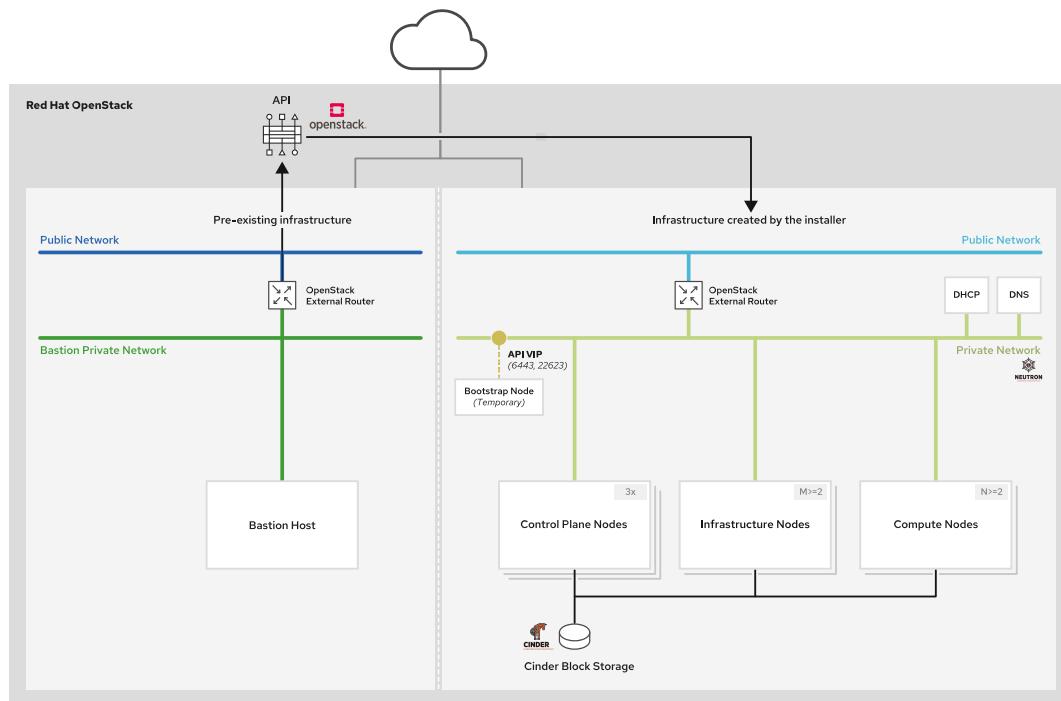


Abbildung 2.8: OpenShift-Installation mit Full-Stack-Automatisierung auf RHOSP – Bootstrap-Phase

Produktions-Control Plane-Phase

In dieser Phase werden die Kubernetes-API und die Produktions-Control Plane auf den Control Plane-Knoten ausgeführt. Der Cluster Version Operator (CVO), der auf der Produktions-Control Plane ausgeführt wird, installiert alle Operatoren, die den Cluster erstellen, und beendet anschließend die Installation.

Das OpenShift-Installationsprogramm führt Folgendes aus:

- Veröffentlichen der IP-Adressen des OpenShift-API-LB und des APP Ingress-LB als Floating-IP-Adressen im öffentlichen RHOSP-Netzwerk (optional)
- Konfigurieren der internen OpenShift-Registry für die Verwendung von RHOSP Swift-Objekt-Storage Wenn die RHOSP-Umgebung nicht den Swift-Objekt-Storage-Service verwendet, konfiguriert das Installationsprogramm die interne OpenShift-Registry für die Verwendung von RHOSP Cinder-Block-Storage.
- Konfigurieren eines dynamischen Storage-Anbieters zur Bereitstellung von persistentem Storage für die containerisierten Anwendungen mithilfe des RHOSP Cinder-Storage-Service

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

Das folgende Diagramm beschreibt die Architektur einer OpenShift-Bereitstellung auf RHOSP mit Kuryr nach Abschluss der Installation.

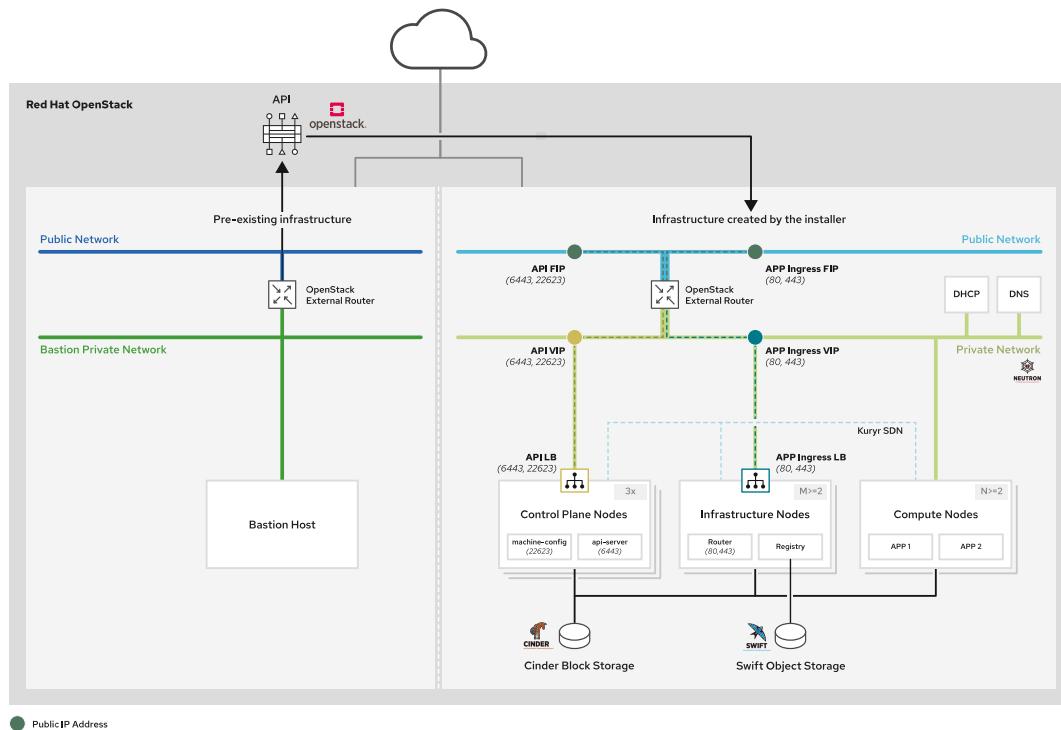


Abbildung 2.9: OpenShift-Installation mit Full-Stack-Automatisierung auf RHOSP – letzte Phase

Aufgaben nach der Installation

Bei umfangreichen OpenShift-Cluster-Bereitstellungen auf RHOSP mit Kuryr empfiehlt Red Hat den Octavia LBaaS zum Hochskalierung des Ingress-Load Balancers.

Standardmäßig stellt das OpenShift-Installationsprogramm die API- und Ingress-Load Balancer auf RHOSP mithilfe der statischen Pods `keepalived` und `haproxy` bereit. Diese statischen Pods werden auf der Control Plane für den API-Load Balancer und auf den Server-Knoten für den Ingress-Load Balancer in einem Aktiv-Passiv-Modus ausgeführt. Für jeden Load Balancer (API und Ingress) stellt nur ein statischer Pod vom Typ `keepalived Service` bereit.

Der aktive `haproxy`-Pod leitet den eingehenden Netzwerkdatenverkehr an die entsprechenden OpenShift-Pods weiter. Um Engpassprobleme auf dem Ingress-Load Balancer zu vermeiden, empfiehlt Red Hat die Verwendung des RHOSP Octavia LBaaS, wie in diesem Diagramm beschrieben.

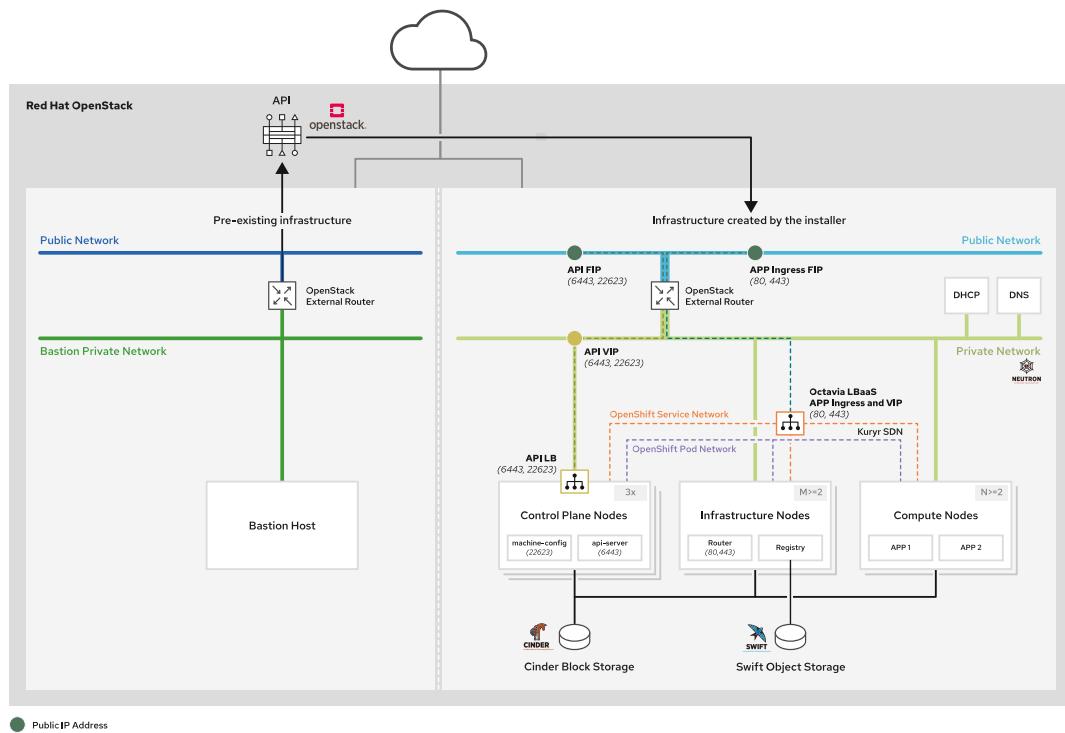


Abbildung 2.10: OpenShift-Installation mit Full-Stack-Automatisierung auf RHOSP – Skalieren des Ingress-Load Balancers mit Octavia LBaaS



Literaturhinweise

- Weitere Informationen finden Sie im Leitfaden *Installing on Azure* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_azure
- Weitere Informationen finden Sie im Leitfaden *Installing on OpenStack* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_openstack

► Quiz

Einführung in die OpenShift-Installation mit Full-Stack-Automatisierung auf einer Cloud-Anbieter-Plattform

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Bei der Installation von OpenShift auf einer generischen Cloud-Anbieter-Plattform mit der Full-Stack-Automatisierungsmethode werden welche drei der folgenden Ressourcen vom Installationsprogramm erstellt? (Wählen Sie drei Antworten aus.)
 - a. Cluster-Knoteninstanzen
 - b. Cluster-Netzwerk
 - c. Benutzerkonto beim Cloud-Anbieter
 - d. Bootstrap-Knoteninstanz
 - e. Der Bastion-Host

- ▶ 2. Der Bootstrap-Knoten wird bei der Installation von OpenShift auf einer generischen Cloud-Anbieter-Plattform mit der Full-Stack-Automatisierungsmethode verwendet. (Richtig oder falsch)
 - a. Richtig
 - b. Falsch

- ▶ 3. Für welche drei der folgenden IaaS-Cloud-Anbieter unterstützt Red Hat die Full-Stack-Automatisierungsmethode für die Installation von OpenShift? (Wählen Sie drei Antworten aus.)
 - a. Amazon Web Services (AWS)
 - b. Red Hat OpenStack Platform (RHOSP)
 - c. Oracle Cloud
 - d. Google Cloud Platform (GCP)

- ▶ 4. Bei der Installation von OpenShift auf MS Azure mit der Full-Stack-Automatisierungsmethode werden welche zwei der folgenden Cloud-Storage-Ressourcen vom Installationsprogramm erstellt? (Wählen Sie zwei Antworten aus.)
 - a. Azure Blob
 - b. Azure Swift
 - c. Azure EBS
 - d. Azure Disk

► **5. Welche zwei der folgenden Aussagen sind Vorteile der Verwendung des Kuryr-SDN für OpenShift-Bereitstellungen auf RHOSP? (Wählen Sie zwei Antworten aus.)**

- a. Verbessert die Netzwerkleistung
- b. Skaliert den Ingress-Load Balancer hoch
- c. Bietet interne Konnektivität zwischen Pods und virtuellen RHOSP-Instanzen
- d. Ermöglicht Doppelkapselung des Netzwerkdatenverkehrs

► Lösung

Einführung in die OpenShift-Installation mit Full-Stack-Automatisierung auf einer Cloud-Anbieter-Plattform

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Bei der Installation von OpenShift auf einer generischen Cloud-Anbieter-Plattform mit der Full-Stack-Automatisierungsmethode werden welche drei der folgenden Ressourcen vom Installationsprogramm erstellt? (Wählen Sie drei Antworten aus.)
 - a. Cluster-Knoteninstanzen
 - b. Cluster-Netzwerk
 - c. Benutzerkonto beim Cloud-Anbieter
 - d. Bootstrap-Knoteninstanz
 - e. Der Bastion-Host

- ▶ 2. Der Bootstrap-Knoten wird bei der Installation von OpenShift auf einer generischen Cloud-Anbieter-Plattform mit der Full-Stack-Automatisierungsmethode verwendet. (Richtig oder falsch)
 - a. Richtig
 - b. Falsch

- ▶ 3. Für welche drei der folgenden IaaS-Cloud-Anbieter unterstützt Red Hat die Full-Stack-Automatisierungsmethode für die Installation von OpenShift? (Wählen Sie drei Antworten aus.)
 - a. Amazon Web Services (AWS)
 - b. Red Hat OpenStack Platform (RHOSP)
 - c. Oracle Cloud
 - d. Google Cloud Platform (GCP)

- ▶ 4. Bei der Installation von OpenShift auf MS Azure mit der Full-Stack-Automatisierungsmethode werden welche zwei der folgenden Cloud-Storage-Ressourcen vom Installationsprogramm erstellt? (Wählen Sie zwei Antworten aus.)
 - a. Azure Blob
 - b. Azure Swift
 - c. Azure EBS
 - d. Azure Disk

► **5. Welche zwei der folgenden Aussagen sind Vorteile der Verwendung des Kuryr-SDN für OpenShift-Bereitstellungen auf RHOSP? (Wählen Sie zwei Antworten aus.)**

- a. Verbessert die Netzwerkleistung
- b. Skaliert den Ingress-Load Balancer hoch
- c. Bietet interne Konnektivität zwischen Pods und virtuellen RHOSP-Instanzen
- d. Ermöglicht Doppelkapselung des Netzwerkdatenverkehrs

Beschreiben der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die Voraussetzungen für die Installation von OpenShift auf Amazon Web Services (AWS) mit Full-Stack-Automatisierung erfüllen können.

Installieren von OpenShift mit Full-Stack-Automatisierung auf Amazon Web Services (AWS)

Mit der Full-Stack-Automatisierungsmethode können Administratoren OpenShift problemlos auf Amazon Web Services (AWS) installieren.

Administratoren müssen alle Voraussetzungen erfüllen, bevor sie eine OpenShift-Installation ausführen. Anschließend fordert das OpenShift-Installationsprogramm zur Eingabe der erforderlichen Informationen auf und erstellt die Datei `install-config.yaml`. Administratoren können die Installation optional anpassen, indem sie die Datei `install-config.yaml` ändern. Schließlich erstellt das OpenShift-Installationsprogramm die erforderlichen AWS-Ressourcen und installiert einen OpenShift-Cluster in etwa 30 Minuten.

Voraussetzungen für die Installation von OpenShift auf AWS

Vor der Installation von OpenShift auf AWS müssen Administratoren die folgenden Aktionen ausführen:

- Überprüfen der in der Lektion *Beschreiben der OpenShift-Installationsvoraussetzungen* beschriebenen allgemeinen Voraussetzungen und Voraussetzungen für die Full-Stack-Automatisierung.
- Erstellen einer öffentlichen DNS-gehosteten Zone für den Cluster im AWS Route53-Service, sofern sie noch nicht vorhanden ist. Diese Zone muss für die Domain autoritativ sein.
- Verwenden eines AWS IAM-Benutzerkontos (Identity and Access Management, Identitäts- und Zugriffsverwaltung) zum Ausführen der Installation. Red Hat empfiehlt, zu diesem Zweck einen zweiten IAM-Administratorbenutzer zu erstellen.
- Sicherstellen, dass das IAM AWS-Benutzerkonto die zum Installieren von OpenShift erforderlichen Kontingent- und Berechtigungsvoraussetzungen erfüllt.

Konfigurieren der Voraussetzungen für die Installation von OpenShift auf AWS

In der folgenden Demonstration werden die detaillierten Schritte gezeigt, die erforderlich sind, um die erforderlichen Voraussetzungen für die Installation von OpenShift mit der Full-Stack-Automatisierungsmethode auf AWS zu erfüllen.

In diesem Szenario wird Folgendes angenommen:

- Der Bastion-Host mit dem Betriebssystem Red Hat Enterprise Linux 8 ist installiert.

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

- Der Bastion-Host hat Zugriff auf die AWS-API, um die Installation durchzuführen.
- Sie verfügen über den SCHLÜSSEL und das SECRET eines AWS-Benutzerkontos:
- Sie müssen den OpenShift-Cluster in der AWS-Region us-east-2 installieren.
- Ihre Cluster-Domain ist example.com.
- Sie verfügen über ein Benutzerkonto namens user auf dem Bastion-Host.
- Der Benutzer user kann mit dem Befehl sudo auf das Konto root zugreifen.

Führen Sie alle Demonstrationsschritte auf dem Bastion-Host aus.

- Installieren und konfigurieren Sie das AWS-CLI-Tool aws auf dem Bastion-Host.

```
[user@bastion ~]$ export AWSKEY=***** && export AWSSECRETKEY=*****
[user@bastion ~]$ export REGION=us-east-2 && mkdir -p $HOME/.aws
[user@bastion ~]$ cat << EOF >> $HOME/.aws/credentials
> [default]
> aws_access_key_id = ${AWSKEY}
> aws_secret_access_key = ${AWSSECRETKEY}
> region = $REGION
> EOF
```

```
[user@bastion ~]$ sudo -i
[root@bastion ~]# curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" \
> -o "awscli-bundle.zip"
[root@bastion ~]# unzip awscli-bundle.zip
[root@bastion ~]# ./awscli-bundle/install -i /usr/local/aws -b /bin/aws
[root@bastion ~]# rm -rf /root/awscli-bundle /root/awscli-bundle.zip
```

- Überprüfen Sie die Installation und Konfiguration der AWS-CLI aws.

```
[root@bastion ~]# su - user
[user@bastion ~]$ aws sts get-caller-identity
{
  "Account": "*****",
  "UserId": "*****",
  "Arn": "arn:aws:iam::*****user/do322"
}
```

- Installieren Sie die Tools openshift-install und oc auf dem Bastion-Host.

```
[user@bastion ~]$ sudo -i
[root@bastion ~]# OCP_VERSION=4.6.4
[root@bastion ~]# MIRROR=mirror.openshift.com/pub/openshift-v4/clients
[root@bastion ~]# wget \
> https://${MIRROR}/ocp/${OCP_VERSION}/openshift-install-linux-${OCP_VERSION}.tar.gz
[root@bastion ~]# tar zxvf openshift-install-linux-${OCP_VERSION}.tar.gz \
> -C /usr/bin
[root@bastion ~]# rm -f openshift-install-linux-${OCP_VERSION}.tar.gz
[root@bastion ~]# chmod +x /usr/bin/openshift-install
```

```
[root@bastion ~]# wget \
> https://$[MIRROR]/ocp/${OCP_VERSION}/openshift-client-linux-${OCP_VERSION}.tar.gz
[root@bastion ~]# tar zxvf openshift-client-linux-${OCP_VERSION}.tar.gz \
> -C /usr/bin
[root@bastion ~]# rm -f openshift-client-linux-${OCP_VERSION}.tar.gz
[root@bastion ~]# chmod +x /usr/bin/oc
[root@bastion ~]# oc completion bash >/etc/bash_completion.d/openshift
```

- Generieren Sie einen SSH-Schlüssel auf dem Bastion-Host.

```
[root@bastion ~]# su - user
[user@bastion ~]$ ssh-keygen -f ${HOME}/.ssh/ocp4-aws-key -N ''
```

- Erstellen Sie bei Bedarf eine öffentliche DNS-gehostete Zone im AWS Route53-Service. Diese Zone muss für die Domain autoritativ sein.

```
[user@bastion ~]$ aws route53 create-hosted-zone \
> --name example.com --caller-reference 2020-12-24-18:02
{
    "HostedZone": {
        "ResourceRecordSetCount": 2,
        "CallerReference": "2020-12-24-18:02",
        "Config": {
            "PrivateZone": false
        },
        "Id": "/hostedzone/Z0...NE2",
        "Name": "example.com."
    },
    "DelegationSet": {
        "NameServers": [
            "ns-1843.awsdns-38.co.uk",
            "ns-453.awsdns-56.com",
            "ns-1516.awsdns-61.org"
        ]
    },
    "Location": "https://route53.amazonaws.com/...",
    "ChangeInfo": {
        "Status": "PENDING",
        "SubmittedAt": "2021-02-08T10:25:20.795Z",
        "Id": "/change/C09...M4B"
    }
}
```

Die Zeichenfolge 2020-12-24-18:02 ist ein eindeutiger Bezeichner, den Sie für diese Ausführung festgelegt haben. Die Befehlsausgabe zeigt, dass der DNS-Server ns-1843.awsdns-38.co.uk für die neu erstellte Domain-Zone autoritativ ist.

Wenn Ihre DNS-Root-Domain nicht von AWS verwaltet wird, müssen Sie die relevanten Delegierungseinträge so konfigurieren, dass das AWS-DNS der gehosteten Domain als autoritativ festgelegt wird. Die Weitergabe neuer gehosteter Domain-Zonen auf AWS- und Internet-DNS-Servern kann bis zu 48 Stunden dauern.

- Überprüfen Sie die Propagation der DNS-gehosteten Domain.

```
[user@bastion ~]$ dig example.com
; <>> DiG 9.11.4-P2-RedHat-9.11.4-9.P2.el7 <>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52429
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com. IN A

;; AUTHORITY SECTION:
example.com. 300 IN SOA ns-1843.awsdns-38.co.uk. awsdns-hostmaster.amazon.com. 1
7200 900 1209600 86400

;; Query time: 848 msec
;; SERVER: 192.168.0.2#53(192.168.0.2)
;; WHEN: Tue Dec 29 13:45:44 UTC 2020
;; MSG SIZE rcvd: 134
```

Die Ausgabe des Befehls `dig` bestätigt, dass der DNS-Server `ns-1843.awsdns-38.co.uk` für die erstellte gehostete Domain-Zone autoritativ ist.

- Überprüfen Sie den Status der neuen gehosteten Domain-Zone im AWS Route53-Service.

```
[user@bastion ~]$ aws route53 list-hosted-zones
{
  "HostedZones": [
    {
      "ResourceRecordSetCount": 3,
      "CallerReference": "example.com.-15...62.53",
      "Config": {
        "PrivateZone": false
      },
      "Id": "/hostedzone/Z24...BD2",
      "Name": "example.com."
    },
    ...
  ...output omitted...
```

- Stellen Sie sich, dass das AWS-Benutzerkonto über die zum Installieren von OpenShift erforderlichen Berechtigungen verfügt.

```
[user@bastion ~]$ aws iam get-account-authorization-details
[user@bastion ~]$ aws iam get-account-authorization-details --filter=Role
...output omitted...
```

- Stellen Sie sich, dass das AWS-Benutzerkonto über die zum Installieren von OpenShift erforderlichen Kontingente und Beschränkungen verfügt.

```
[user@bastion ~]$ aws service-quotas list-service-quotas --service-code=ec2
...output omitted...
```

```
[user@bastion ~]$ aws service-quotas list-service-quotas --service-code=vpc
{
    "Quotas": [
        {
            "QuotaName": "Active VPC peering connections per VPC",
            "Adjustable": true,
            "QuotaArn": "arn:aws:servicequotas:us-east-2:743309855567:vpc/L-7E9ECCDB",
            "Value": 50.0,
            "ServiceName": "Amazon Virtual Private Cloud (Amazon VPC)",
            "GlobalQuota": false,
            "ServiceCode": "vpc",
            "QuotaCode": "L-7E9ECCDB",
            "Unit": "None"
        },
        ...
    ]
}
```

- Rufen Sie das **pull-secret** der Registry ab.

Öffnen Sie Ihren Webbrowser und navigieren Sie zu <https://cloud.redhat.com/openshift/install/aws/installer-provisioned>. Melden Sie sich mit Ihrem Red Hat-Benutzerkonto an.

Klicken Sie auf die Schaltfläche **Download pull secret**, und wählen Sie **Save file** aus. Die Datei wird standardmäßig unter /home/user/Downloads/pull-secret gespeichert.

Dimensionieren von OpenShift-Installationen auf AWS

Administratoren müssen die Kontingente der AWS-Kontoressourcen überprüfen, um die folgenden Mindestanforderungen zu gewährleisten.

AWS-Beschränkungen, die sich auf die Installation und Ausführung von OpenShift-Clustern auswirken können

Ressource	Erforderlich	AWS-Standardlimit
Instanzen	7	Variiert
Elastic IPs (EIPs)	1	5 EIPs pro Benutzerkonto
Virtual Private Clouds (VPCs)	5	5 VPCs pro Region
Elastic Load Balancing (ELB)	3	20 pro Region
NAT-Gateways	5	5 pro Verfügbarkeitszone
Elastic Network Interfaces (ENIs)	Mindestens 12	350 pro Region
VPC-Gateway	20	20 pro Konto
S3-Buckets	99	100 Buckets pro Benutzerkonto
Sicherheitsgruppen	250	2.500 pro Benutzerkonto



Anmerkung

Die in dieser Tabelle beschriebenen OpenShift-Ressourcenanforderungen sind für die Installation eines kleinen OpenShift-Clusters ausreichend. Die auf dem OpenShift-Cluster ausgeführten Anwendungs-Workloads können sich auf die tatsächlichen Anforderungen auswirken.

Anpassen von OpenShift-Installationen auf AWS

Die vom OpenShift-Installationsprogramm generierte Datei `install-config.yaml` enthält die folgenden Standardwerte:

```
[user@demo ~]$ cat ${HOME}/ocp4-aws-cluster/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform: {}
  replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: ocp4-aws
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
    networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  aws:
    region: us-east-2
publish: External
pullSecret: '{"auths":{"cloud.openshift.com": ...}}
sshKey: |
  ssh-rsa AAAA....
```

Administratoren können die Datei `install-config.yaml` ändern, um die OpenShift-Installation auf AWS anzupassen. Die folgenden Anpassungen werden am häufigsten vorgenommen:

- Erhöhen der EC2-Instanzgrößen für die Cluster-Knoten
- Erhöhen der EBS-Festplattengrößen für die Cluster-Knoten

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

- Verwenden von Hochleistungs-Storage für den etcd-Cluster
- Verteilen der Cluster-Knoten auf verschiedene AWS-Verfügbarkeitszonen (VZs)

Anpassen von EC2-Instanzgrößen

Die für eine OpenShift-Installation auf AWS verwendeten Standardgrößen für AWS EC2-Instanzen sind m4.xlarge für die Control Plane-Knoten und m4.large für die Server-Knoten. Diese Standardgrößen für AWS EC2-Instanzen sind für Proof of Concept (PoC)- und andere Nichtproduktionsszenarien geeignet.

Verwenden Sie für Produktions-Cluster größere AWS EC2-Instanzgrößen als die Standardgrößen. Verwenden Sie die folgenden AWS EC2-Instanzgrößenwerte als Referenz.

AWS-Instanzgrößen für einen Proof-of-Concept-OpenShift-Cluster

Knoten	Flavor	vCPU	RAM (GiB)
Control Plane-Knoten (etcd)	m4.xlarge	4	16
Server-Knoten	m4.large	2	8

AWS-Instanzgrößen für einen kleinen OpenShift-Cluster

Knoten	Flavor	vCPU	RAM (GiB)
Control Plane-Knoten (etcd)	m5.2xlarge	8	32
Server-Knoten	m5.4xlarge	16	64

AWS-Instanzgrößen für einen mittelgroßen OpenShift-Cluster

Knoten	Flavor	vCPU	RAM (GiB)
Control Plane-Knoten (etcd)	r5.4xlarge	16	128
Server-Knoten	m5.8xlarge	32	128

AWS-Instanzgrößen für einen großen OpenShift-Cluster

Knoten	Flavor	vCPU	RAM (GiB)
Control Plane-Knoten (etcd)	r5.8xlarge	32	256
Server-Knoten	m5.12xlarge	48	192

Verwenden Sie die Daten in den vorherigen Tabellen nur als Referenz. Die geeigneten AWS EC2-Instanzgrößen für die Verwendung in OpenShift-Produktions-Clustern hängen von vielen Variablen ab, beispielsweise:

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

- Größe des OpenShift-Clusters
- Welche Infrastruktur-Services auf dem OpenShift-Cluster ausgeführt werden müssen
- Welche Arten von Anwendungs-Workloads auf dem OpenShift-Cluster ausgeführt werden müssen
- Wie viele Anwendungsanforderungen der OpenShift-Cluster maximal unterstützen muss

**Anmerkung**

Weitere Informationen finden Sie im Abschnitt *Planning your environment according to object maximums* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/scalability_and_performance.

Das folgende Beispiel zeigt die Anpassung der AWS EC2-Instanzgrößen für einen kleinen Produktions-Cluster.

```
[user@demo ~]$ cat ${HOME}/ocp4-aws-cluster/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform:
    aws:
      type: m5.4xlarge
    replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      type: m5.2xlarge
    replicas: 3
...output omitted...
platform:
  aws:
    region: us-east-2
publish: External
pullSecret: '{"auths":{"cloud.openshift.com": ...}}
sshKey: |
  ssh-rsa AAAA....
```

**Anmerkung**

Weitere Informationen finden Sie im Abschnitt *Amazon EC2-Instance-Typen* der Dokumentation zu Amazon Web Services unter <https://aws.amazon.com/ec2/instance-types>.

Verwenden von Hochleistungs-Storage für etcd

Das OpenShift-Installationsprogramm stellt auf jedem Cluster-Knoten standardmäßig ein EBS-gp2-Volume mit 120 GiB für die Installation von RHCOS bereit. Bei einem großen OpenShift-Cluster, in dem etcd eine hohe I/O-Leistung erfordert, reicht das für den etcd-Storage verwendete gp2-Volume nicht aus. Um bestmögliche etcd-Zuverlässigkeit zu erzielen, sollte bevorzugt die Storage-Technologie mit der niedrigsten konsistenten Latenz verwendet werden.

In der folgenden Tabelle werden den Amazon EBS-Volume-Typen verglichen.

Amazon EBS-Volume-Typen

Volume-Typ	gp2	gp3	io1	io2 Block Express
Kategorie	Allzweck-SSD	Allzweck-SSD	Bereitgestellte IOPS-SSD	Bereitgestellte IOPS-SSD
Anwendungsfall	Interaktive Apps mit geringer Latenz	Interaktive Apps mit geringer Latenz	Latenz unter einer Millisekunde und anhaltende IOPS-Leistung	Latenz unter einer Millisekunde und anhaltende IOPS-Leistung
Volume-Größe	1 GiB – 16 TiB	1 GiB – 16 TiB	4 GiB – 16 TiB	4 GiB – 64 TiB
Max. IOPS pro Volume	16.000	16.000	64.000	256.000
Max. Durchsatz pro Volume	250 MiB/s	1.000 MiB/s	1.000 MiB/s	4.000 MiB/s



Anmerkung

Weitere Informationen finden Sie im Abschnitt *Amazon EBS-Volume-Typen* der Dokumentation zu Amazon Web Services unter <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html>.

Um schnelleren Storage für etcd zu konfigurieren, müssen Sie den Storage-Typ minimal auf **io1** und den IOPS-Wert auf 4.000 auf Control Plane-Knoten festlegen. Erhöhen Sie außerdem die EBS-Volume-Größe für alle Cluster-Knoten.

```
[user@demo ~]$ cat ${HOME}/ocp4-aws-cluster/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 250
```

```
type: io1
type: m5.4xlarge
replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      rootVolume:
        iops: 4000
        size: 250
        type: io1
      type: m5.2xlarge
    replicas: 3
...output omitted...
platform:
  aws:
    region: us-east-2
publish: External
pullSecret: '{"auths":{"cloud.openshift.com": ...}}
sshKey: |
  ssh-rsa AAAA....
```



Anmerkung

Der Parameter `iops` konfiguriert die Eingabe- und Ausgabevorgänge pro Sekunde (IOPS), die für das Volume reserviert sind.

Verteilen von Cluster-Knoten auf AWS-Verfügbarkeitszonen

Standardmäßig verteilt das OpenShift-Installationsprogramm die Cluster-Knoten auf verschiedene AWS-Verfügbarkeitszonen (VZs) in derselben AWS-Region. Bei dieser Konfiguration kombiniert der installierte OpenShift-Cluster die Resilienz- und Hochverfügbarkeitsmechanismen, die von Red Hat OpenShift und AWS bereitgestellt werden.

Red Hat empfiehlt, alle Cluster-Knoten auf mehreren VZs zu verteilen, um sich vor Problemen zu schützen, die eine einzelne VZ betreffen würden. Administratoren können die VZs auswählen, in denen die Cluster-Knoten installiert werden.

Die folgende `install-config.yaml`-Konfigurationsdatei enthält die in dieser Lektion beschriebene Anpassungen von AWS-VZs.

```
[user@demo ~]$ cat ${HOME}/ocp4-aws-cluster/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform:
    aws:
      zones:
      - us-east-2a
```

```
- us-east-2b
- us-east-2c
rootVolume:
  iops: 2000
  size: 250
  type: io1
  type: m5.4xlarge
replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      zones:
        - us-east-2a
        - us-east-2b
        - us-east-2c
      rootVolume:
        iops: 4000
        size: 250
        type: io1
        type: m5.2xlarge
      replicas: 3
    metadata:
      creationTimestamp: null
      name: ocp4-aws
  networking:
    clusterNetwork:
      - cidr: 10.128.0.0/14
      hostPrefix: 23
    machineNetwork:
      - cidr: 10.0.0.0/16
    networkType: OpenShiftSDN
    serviceNetwork:
      - 172.30.0.0/16
  platform:
    aws:
      region: us-east-2
  publish: External
  pullSecret: '{"auths":{"cloud.openshift.com": ...'
  sshKey: |
    ssh-rsa AAAA....
```

Beschreiben von OpenShift-Installationen auf AWS

Die zum Ausführen einer OpenShift-Installation auf AWS erforderlichen Schritte werden im Abschnitt *Beschreiben von OpenShift-Installationen auf einer Cloud-Anbieter-Plattform* in diesem Dokument erläutert. In diesem Abschnitt haben Sie die Schritte zum Ausführen einer OpenShift-Installation auf einer generischen Cloud-Anbieter-Plattform kennen gelernt. Diese Schritte gelten für die OpenShift-Installation auf der AWS Cloud-Anbieter-Plattform.

In diesem Abschnitt werden die Besonderheiten der OpenShift-Installation auf der Cloud-Anbieter-Plattform AWS in den einzelnen Installationsphasen beschrieben.

Phase der Ressourcenerstellung

Zu Beginn der Installation erstellt und konfiguriert das OpenShift-Installationsprogramm die erforderlichen Cloud-Ressourcen.

Das OpenShift-Installationsprogramm verwendet Folgendes:

- Den Service „Amazon Virtual Private Cloud (VPC)“ zum Erstellen des Cluster-Netzwerks und der Cluster-Netzwerkservices
- Den Service „Amazon NAT Gateway“ zum Erstellen des Cluster-Netzwerk-Gateways
- Den Service „Amazon Elastic Load Balancer (ELB)“ zum Erstellen der Cluster-Load Balancer
- Den Service „Amazon EC2“ zum Erstellen der Cluster-Server-Knoten
- Die Services „Amazon EBS“ und „Amazon S3“ für den Cluster-Storage

Das folgende Diagramm zeigt die Beziehung zwischen OpenShift-Ressourcen und den AWS-Services, die zum Erstellen der Ressourcen verwendet werden.

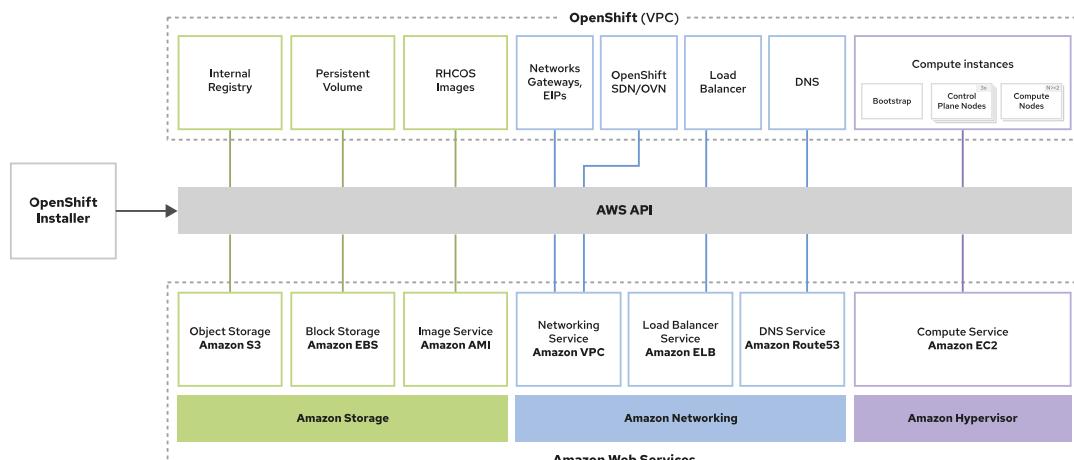


Abbildung 2.11: Ressourcen der OpenShift-Installation mit Full-Stack-Automatisierung auf AWS

Bootstrap-Phase

In dieser Phase werden die Kubernetes-API und die temporäre Control Plane auf dem Bootstrap-Knoten ausgeführt. Das OpenShift-Installationsprogramm beginnt mit der Installation der Cluster-Knoten.

Im folgenden Diagramm wird die Architektur einer OpenShift-Bereitstellung auf AWS in der Bootstrap-Phase erläutert.

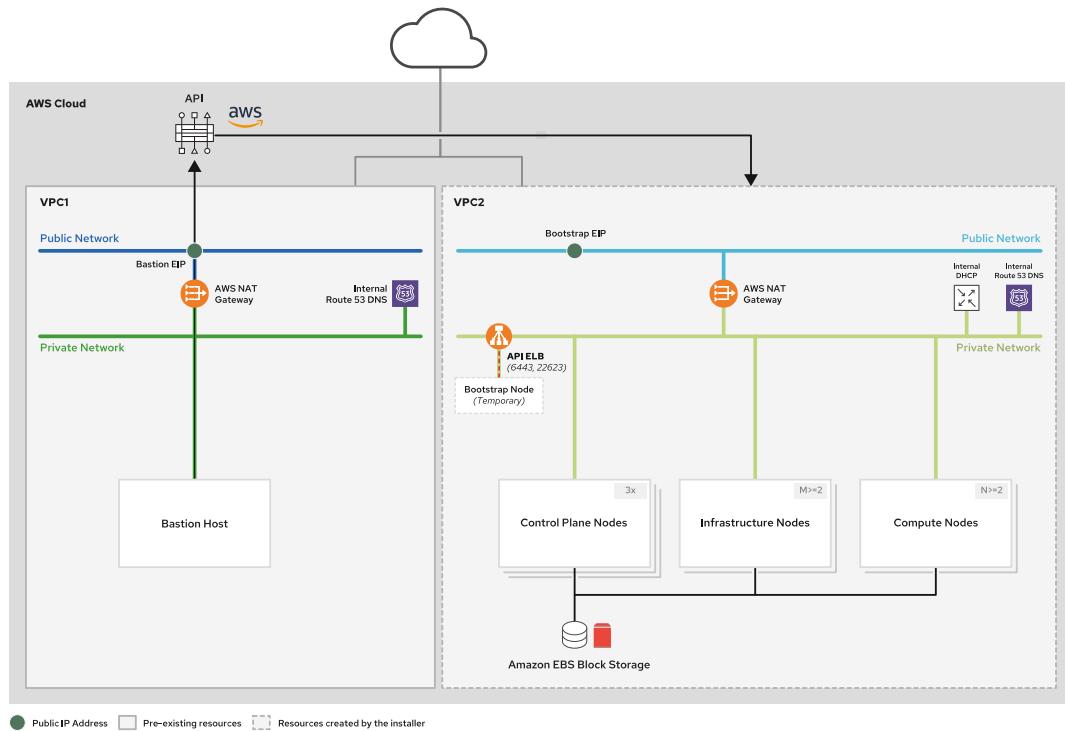


Abbildung 2.12: OpenShift-Installation mit Full-Stack-Automatisierung auf AWS – Bootstrap-Phase

Produktions-Control Plane-Phase

In dieser Phase werden die Kubernetes-API und die Produktions-Control Plane auf den Control Plane-Knoten ausgeführt. Der Cluster Version Operator (CVO), der auf der Produktions-Control Plane ausgeführt wird, installiert alle Operatoren, die den Cluster erstellen, und beendet anschließend die Installation.

Das OpenShift-Installationsprogramm führt Folgendes aus:

- Veröffentlichen der IP-Adressen des OpenShift-API-ELB und des APP Ingress-ELB als öffentliche IP-Adressen im öffentlichen AWS-Netzwerk (optional)
- Konfigurieren der internen OpenShift-Registry für die Verwendung von AWS S3-Objekt-Storage
- Konfigurieren eines dynamischen Storage-Anbieters zur Bereitstellung von persistentem Storage für die containerisierten Anwendungen mithilfe des AWS EBS-Storage-Service

Das folgende Diagramm beschreibt die Architektur einer OpenShift-Bereitstellung auf AWS nach Abschluss der Installation.

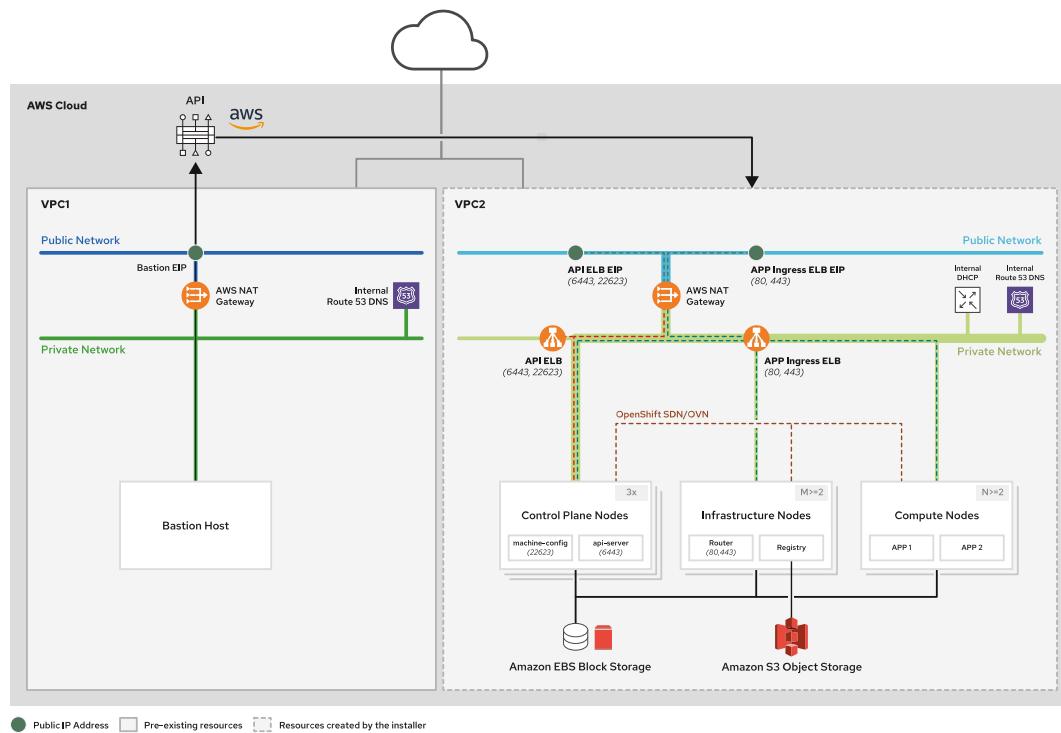


Abbildung 2.13: OpenShift-Installation mit Full-Stack-Automatisierung auf AWS – letzte Phase

Währen der OpenShift-Installation auf AWS:

- Der Bastion-Host benötigt eine Netzwerkverbindung zur Cloud-Anbieter-API, um OpenShift zu installieren, und zur OpenShift-API, um OpenShift nach der Installation zu verwalten. Es ist nicht obligatorisch (wird jedoch empfohlen), den Bastion-Host auf derselben Cloud-Anbieter-Plattform zu installieren, die für die Installation von OpenShift verwendet wird.
- Bei der OpenShift-Installation werden keine Infrastrukturknoten erstellt, sondern nur Server-Knoten. Bei den Infrastrukturknoten handelt es sich um Server-Knoten, auf denen Router- oder Registry-Pods ausgeführt werden.
- Der OpenShift-Installationsprozess erstellt die Cloud-Ressource „APP Ingress-LB“, die alle Server-Knoten als Back-End-Pool-Mitglieder für die Ports 80/TCP und 443/TCP enthält. Nur Server-Knoten, auf denen Router-Pods ausgeführt werden, die diese Ports überwachen, bestehen die LB-Integritätsprüfung. Daher sendet der APP Ingress-LB Netzwerksdatenverkehr nur an die Server-Knoten, auf denen die Router-Pods ausgeführt werden.
- Die DHCP-Konfiguration stellt auch die NTP-Konfiguration bereit.
- Standardmäßig verwendet der auf jedem Cluster-Knoten ausgeführte systemd-Service `chrony` den öffentlichen NTP-Pool `rhel.pool.ntp.org` zum Synchronisieren der Systemuhr. Nach der Installation können Administratoren die Cluster-Knoten mit dem Machine Config-Operator für die Verwendung eines lokalen NTP-Servers konfigurieren, wie im Abschnitt *Configuring chrony time service* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/post-installation_configuration erläutert.

Konfigurieren der Voraussetzungen für die Installation von OpenShift auf AWS

Videoergebnis:

Konfigurieren der Voraussetzungen für die Installation eines OpenShift-Clusters mithilfe der Full-Stack-Automatisierung auf AWS



Literaturhinweise

- Weitere Informationen finden Sie im Leitfaden *Installing on AWS* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter
https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_aws
- Weitere Informationen zu AWS-Benutzerkonten finden Sie im Abschnitt *Creating an IAM user* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter
https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_aws

► Quiz

Beschreiben der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Welche drei der folgenden Ressourcen werden bei Verwendung der Full-Stack-Automatisierungsmethode auf AWS vom Installationsprogramm erstellt? (Wählen Sie drei Antworten aus.)
 - a. Eine dedizierte VPC für die Installation von OpenShift
 - b. Eine DNS-gehostete Zone im AWS Route53-Service
 - c. Die API- und Ingress-Load Balancer
 - d. Der als persistenter Storage für die interne Registry verwendete AWS S3-Objekt-Storage
 - e. Ein zweiter IAM-Benutzer

- ▶ 2. Administratoren können OpenShift mithilfe der Installationsmethode „Bereits vorhandene Infrastruktur“ auf AWS installieren. (Richtig oder falsch)
 - a. Richtig
 - b. Falsch

- ▶ 3. Mit welchen zwei der folgenden AWS-CLI-Tool-Befehle können Berechtigungen und Kontingente von AWS-Benutzerkonten überprüft werden? (Wählen Sie zwei Antworten aus.)
 - a. aws sts get-caller-identity
 - b. aws iam get-account-authorization-details
 - c. aws iam policy identity
 - d. aws service-quotas list-service-quotas

- ▶ 4. Welche zwei der folgenden Antworten sind gängige Anpassungen für große OpenShift-Bereitstellungen auf AWS? (Wählen Sie zwei Antworten aus.)
 - a. Vergrößern der EC2-Instanzgrößen
 - b. Hochskalieren der Anzahl der Control Plane-Knoten
 - c. Verwenden von Hochleistungs-Storage für den etcd-Cluster
 - d. Verwenden des EBS-gp2-Volumes als persistenten Storage für die interne Registry

► Lösung

Beschreiben der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Welche drei der folgenden Ressourcen werden bei Verwendung der Full-Stack-Automatisierungsmethode auf AWS vom Installationsprogramm erstellt? (Wählen Sie drei Antworten aus.)
 - a. Eine dedizierte VPC für die Installation von OpenShift
 - b. Eine DNS-gehostete Zone im AWS Route53-Service
 - c. Die API- und Ingress-Load Balancer
 - d. Der als persistenter Storage für die interne Registry verwendete AWS S3-Objekt-Storage
 - e. Ein zweiter IAM-Benutzer

- ▶ 2. Administratoren können OpenShift mithilfe der Installationsmethode „Bereits vorhandene Infrastruktur“ auf AWS installieren. (Richtig oder falsch)
 - a. Richtig
 - b. Falsch

- ▶ 3. Mit welchen zwei der folgenden AWS-CLI-Tool-Befehle können Berechtigungen und Kontingente von AWS-Benutzerkonten überprüft werden? (Wählen Sie zwei Antworten aus.)
 - a. aws sts get-caller-identity
 - b. aws iam get-account-authorization-details
 - c. aws iam policy identity
 - d. aws service-quotas list-service-quotas

- ▶ 4. Welche zwei der folgenden Antworten sind gängige Anpassungen für große OpenShift-Bereitstellungen auf AWS? (Wählen Sie zwei Antworten aus.)
 - a. Vergrößern der EC2-Instanzgrößen
 - b. Hochskalieren der Anzahl der Control Plane-Knoten
 - c. Verwenden von Hochleistungs-Storage für den etcd-Cluster
 - d. Verwenden des EBS-gp2-Volumes als persistenten Storage für die interne Registry

Demonstrieren der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode

Ziele

Nach Abschluss dieses Abschnitts sollten Sie OpenShift auf Amazon Web Services (AWS) unter Verwendung von Full-Stack-Automatisierung mit allgemeinen Anpassungen installieren können.

Ausführen der OpenShift-Installation mit Full-Stack-Automatisierung auf Amazon Web Services (AWS)

Im vorherigen Abschnitt haben Sie erfahren, wie Sie eine OpenShift-Installation auf AWS mit der Full-Stack-Automatisierungsmethode vorbereiten. Darüber hinaus haben Sie erfahren, welche Voraussetzungen Sie erfüllen müssen, bevor Sie OpenShift auf AWS installieren können.

Im Anschluss an den im Abschnitt *Einführung in OpenShift-Installationsmethoden* erläuterten OpenShift-Installations-Workflow besteht der nächste Schritt nach Erfüllung der Voraussetzungen darin, die OpenShift-Installation mit der Binärdatei `openshift-installer` auf dem Bastion-Host auszuführen.

Weitere Informationen zu den unterstützten Konfigurationsfeldern für die Installation von OpenShift auf AWS können Sie mit der Option `explain` des OpenShift-Installationsprogramms aufrufen:

```
[user@demo ~]$ openshift-install explain installconfig.platform.aws
KIND:     InstallConfig
VERSION:  v1

RESOURCE: <object>
    AWS is the configuration used when installing on AWS.

FIELDS:
    amiID <string>
        AMIID is the AMI that should be used to boot machines for the cluster. If set, the AMI should belong to the same region as the cluster.

    defaultMachinePlatform <object>
        DefaultMachinePlatform is the default configuration used when installing on AWS for machine pools which do not define their own platform configuration.

    region <string> -required-
        Region specifies the AWS region where the cluster will be created.

    serviceEndpoints <[]object>
        ServiceEndpoints list contains custom endpoints which override the default service endpoint of AWS Services. A service can have only one ServiceEndpoint.
        ServiceEndpoint stores the configuration for services to override existing AWS Services defaults.

    subnets <[]string>
```

```
Subnets specifies existing subnets (by ID) where cluster resources will be created. Leave unset to have the installer create subnets in a new VPC on your behalf.
```

```
userTags <object>
```

```
UserTags additional keys and values that the installer will add as tags to all resources that it creates. Resources created by the cluster itself may not include these tags.
```

Dies sind die unterstützten Felder, die Sie im Abschnitt `platform.aws` der Installationskonfigurationsdatei `install-config.yaml` verwenden können.

```
[user@demo ~]$ cat ${HOME}/ocp4-aws-cluster/install-config.yaml
apiVersion: v1
baseDomain: example.com
...output omitted...
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  aws:
    region: us-east-2
publish: External
pullSecret: '{"auths":...}'
sshKey: |
  ssh-rsa AA...
```

Erstellen von OpenShift-Installationsressourcen

Wie an anderer Stelle in diesem Kurs erläutert, sind die wichtigsten vom OpenShift-Installationsprogramm erstellten Ressourcen die Installationskonfigurationsdatei, die Kubernetes-Manifeste und die Ignition-Konfigurationsdateien.

Das OpenShift-Installationsprogramm transformiert die Installationskonfigurationsdatei in Kubernetes-Manifeste und bindet die Manifeste dann in Ignition-Konfigurationsdateien ein. Anhand dieser Ignition-Konfigurationsdateien erstellt das OpenShift-Installationsprogramm OpenShift-Cluster.

Erstellen des Installationsverzeichnisses

Erstellen Sie vor dem Ausführen des OpenShift-Installationsprogramms ein Installationsverzeichnis, in dem alle Installationsressourcen gespeichert werden.

```
[user@demo ~]$ mkdir ${HOME}/ocp4-aws-cluster
```

Generieren der Installationskonfigurationsdatei

Führen Sie nach dem Erstellen des Installationsverzeichnisses die OpenShift-Installationsprogramm-Binärdatei aus, um die erste Installationsressource zu erstellen: die Installationskonfigurationsdatei `install-config.yaml`. Das Installationsprogramm fordert Sie zur Eingabe der erforderlichen Cluster-Informationen auf und erstellt dann die Datei `install-config.yaml` entsprechend.

```
[user@demo ~]$ openshift-install create install-config \
> --dir=${HOME}/ocp4-aws-cluster
? SSH Public Key /home/user/.ssh/ocp4-aws-key.pub
? Platform aws
INFO Credentials loaded from the "default" profile in file "/home/user/.aws/
credentials"
? Region us-east-2
? Base Domain example.com
? Cluster Name ocp4-aws
? Pull Secret [? for help] ++++++
INFO Install-Config created in: /home/user/ocp4-aws-cluster
```

```
[user@demo ~]$ find ${HOME}/ocp4-aws-cluster
/home/user/ocp4-aws-cluster
/home/user/ocp4-aws-cluster/.openshift_install.log
/home/user/ocp4-aws-cluster/.openshift_install_state.json
/home/user/ocp4-aws-cluster/install-config.yaml
...output omitted...
```

Das Installationsprogramm erstellt die Datei `install-config.yaml` einschließlich der angegebenen Informationen und einiger Cluster-Standardwerte.

```
[user@demo ~]$ cat ${HOME}/ocp4-aws-cluster/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform: {}
  replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: ocp4-aws
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
```

```
- cidr: 10.0.0.0/16
networkType: OpenShiftSDN
serviceNetwork:
- 172.30.0.0/16
platform:
aws:
region: us-east-2
publish: External
pullSecret: '{"auths":...}'
sshKey: |
ssh-rsa AA...
```

Anpassen der Installationskonfigurationsdatei

Sie können die Installation anpassen, indem Sie die Datei `install-config.yaml` ändern. Die folgenden Anpassungen werden bei der Installation von OpenShift-Produktions-Clustern auf AWS häufig vorgenommen:

- Erhöhen der EC2-Instanzgrößen für die Cluster-Knoten
- Erhöhen der EBS-Festplattengrößen für die Cluster-Knoten
- Verwenden von Hochleistungs-Storage für den etcd-Cluster
- Verteilen der Cluster-Knoten auf verschiedene AWS-Verfügbarkeitszonen (VZs)

In der folgenden Datei `install-config.yaml` werden einige Anpassungen hervorgehoben:

```
[user@demo ~]$ cat ${HOME}/ocp4-aws-cluster/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- architecture: amd64
hyperthreading: Enabled
name: worker
platform:
aws:
zones:
- us-east-2a
- us-east-2b
- us-east-2c
rootVolume:
iops: 2000
size: 250
type: io1
type: m5.4xlarge
replicas: 3
controlPlane:
architecture: amd64
hyperthreading: Enabled
name: master
platform:
aws:
zones:
- us-east-2a
- us-east-2b
```

```
- us-east-2c
rootVolume:
  iops: 4000
  size: 250
  type: io1
  type: m5.2xlarge
replicas: 3
metadata:
  creationTimestamp: null
  name: ocp4-aws
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/12
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  aws:
    region: us-east-2
publish: External
pullSecret: '{"auths":{"cloud.openshift.com": ...}}
sshKey: |
  ssh-rsa AAAA....
```

Sie können auch das OpenShift-Cluster-Netzwerk anpassen. Nachfolgend sehen Sie die Standardkonfiguration für Cluster-Netzwerke.

- Das Cluster-Netzwerk ist das Netzwerk **10.128.0.0/14** (**10.128.0.0-10.131.255.255**).
- Die Cluster-Knoten werden in /23-Subnetzen des Cluster-Netzwerks (**10.128.0.0/23**, **10.128.2.0/23**,...) zugeordnet.

Dies bedeutet Folgendes:

- Das Cluster-Netzwerk hat ($23-14 = 9$ Bit = $2^9 =$) 512 Subnetze, die Cluster-Knoten zugeordnet werden können.
- Jedem Cluster-Knoten wird ein /23-Subnetz zugeordnet, das bis zu ($32-23 = 9$ Bit = $2^9 - 2 =$) 510 IP-Adressen für die darauf ausgeführten Pods hosten kann.

Mithilfe der standardmäßigen Cluster-Netzwerkkonfiguration können Sie einen OpenShift-Cluster mit maximal 512 Knoten erstellen. Jeder Knoten kann bis zu 510 Pods hosten.



Anmerkung

In diesem Fall kann jeder Clusterknoten bis zu 510 IP-Adressen hosten, da Sie die Netzwerk- und Broadcast-Adressen aus dem Subnetz subtrahieren müssen.

In der zuvor gezeigten Datei `install-config.yaml` wird die benutzerdefinierte Cluster-Netzwerkkonfiguration wie folgt verwendet:

- Das Cluster-Netzwerk ist das Netzwerk **10.128.0.0/12** (**10.128.0.0-10.143.255.255**).

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

- Die Cluster-Knoten werden in /23-Subnetzen des Cluster-Netzwerks (10.128.0.0/23, 10.128.2.0/23,...) zugeordnet.

Mithilfe dieser benutzerdefinierten Cluster-Netzwerkkonfiguration können Sie einen OpenShift-Cluster mit maximal 2048 Knoten erstellen. Jeder Knoten kann bis zu 510 Pods hosten. Red Hat hat maximal 500 Server-Knoten auf einem Red Hat OpenShift Container Platform 4.6-Cluster getestet. Wenden Sie sich vor der Installation eines größeren Clusters an den Red Hat Support.

Generieren der Kubernetes-Manifeste

Zu diesem Zeitpunkt haben Sie alle Anpassungen Ihrer Cluster-Installation in die Datei `install-config.yaml` aufgenommen. Vor dem Ausführen des OpenShift-Installationsprogramms zum Erstellen der Kubernetes-Manifeste aus der Datei `install-config.yaml` wird empfohlen, die Datei `install-config.yaml` zu sichern. Das OpenShift-Installationsprogramm entfernt sie automatisch, sobald die Kubernetes-Manifeste erstellt wurden.

```
[user@demo ~]$ cp ${HOME}/ocp4-aws-cluster/install-config.yaml ${HOME}/
```

```
[user@demo ~]$ openshift-install create manifests \
> --dir=${HOME}/ocp4-aws-cluster
INFO Credentials loaded from the "default" profile in file "/home/user/.aws/
credentials"
INFO Consuming Install Config from target directory
INFO Manifests created in: /home/user/ocp4-aws-cluster/manifests and /home/user/
ocp4-aws-cluster/openshift
```

```
[user@demo ~]$ find ${HOME}/ocp4-aws-cluster/manifests
/home/user/ocp4-aws-cluster/manifests
/home/user/ocp4-aws-cluster/manifests/04-openshift-machine-config-operator.yaml
/home/user/ocp4-aws-cluster/manifests/cluster-config.yaml
...output omitted...
```

Für die Installation von OpenShift auf AWS mit der Full-Stack-Automatisierungsmethode müssen Sie alle Cluster-Anpassungen mit der Datei `install-config.yaml` durchführen. Sie dürfen die Kubernetes-Manifeste nach dem Erstellen nicht ändern.



Warnung

Das Ändern von Kubernetes-Manifesten wird nur unterstützt, wenn Sie dokumentierte Red Hat-Verfahren oder Anweisungen des Red Hat-Supports befolgen. Andernfalls wird dies nicht unterstützt.

Generieren der Ignition-Konfigurationsdateien

Nachdem das Installationsprogramm die Kubernetes-Manifeste aus der benutzerdefinierten Datei `install-config.yaml` erstellt hat, bindet das Installationsprogramm die Manifeste in Ignition-Konfigurationsdateien ein. Die Ignition-Konfigurationsdateien sind die letzten vom Installationsprogramm erstellten Installationsressourcen.

Anhand dieser Ignition-Konfigurationsdateien erstellt das OpenShift-Installationsprogramm abschließend den OpenShift-Cluster.

```
[user@demo ~]$ openshift-install create ignition-configs \
> --dir=${HOME}/ocp4-aws-cluster
INFO Consuming Master Machines from target directory
INFO Consuming OpenShift Install (Manifests) from target directory
INFO Consuming Openshift Manifests from target directory
INFO Consuming Worker Machines from target directory
INFO Consuming Common Manifests from target directory
INFO Ignition-Configs created in: /home/user/ocp4-aws-cluster and /home/user/ocp4-aws-cluster/auth
```

```
[user@demo ~]$ find ${HOME}/ocp4-aws-cluster -name '*.ign' | xargs ls -lrt
-rw-r----- 1 user user 1732 Dec 27 19:35 /home/user/ocp4-aws-cluster/master.ign
-rw-r----- 1 user user 1732 Dec 27 19:35 /home/user/ocp4-aws-cluster/worker.ign
-rw-r----- 1 user user 307594 Dec 27 19:35 /home/user/ocp4-aws-cluster/
bootstrap.ign
```

Wie bereits an anderer Stelle in diesem Kurs erläutert:

- Die Ignition-Datei `bootstrap.ign` enthält alle erforderlichen Anweisungen zum Installieren des Bootstrap-Knotens und zum Starten der temporären Control Plane. Der Bootstrap-Knoten ruft seine `bootstrap.ign`-Ignition-Datei aus einem AWS S3-Bucket ab, wendet die Konfiguration aus dieser Ignition-Datei an und führt dann die temporäre Control Plane aus.
- Beim ersten Booten rufen die Control Plane-Knoten und die Serverknoten auch ihre Ignition-Dateien aus einem AWS S3-Bucket ab. Diese Ignition-Dateien (`master.ign` und `worker.ign`) enthalten nur eine Umleitungsanweisung, um die entsprechenden Ignition-Dateien vom Machine Config Server (MCS) zu erhalten, der auf der (temporären) OpenShift-Control-Plane ausgeführt wird.



Warnung

Das Ändern von Ignition-Konfigurationsdateien wird nur unterstützt, wenn Sie dokumentierte Red Hat-Verfahren oder Anweisungen des Red Hat-Supports befolgen. Andernfalls wird dies nicht unterstützt.

Ausführen des OpenShift-Installationsprozesses

Nach dem Abrufen der Ignition-Konfigurationsdateien können Sie die Cluster-Bereitstellung mit dem OpenShift-Installationsprogramm starten. Die Installationsprotokolle finden Sie in der Datei `${install_dir} / .openshift_install.log`. In den OpenShift-Installationsprozessprotokollen können Sie die folgenden Phasen unterscheiden:

```
[user@demo ~]$ openshift-install create cluster \
> --dir=${HOME}/ocp4-aws-cluster --log-level=debug
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
...output omitted...
INFO Consuming Bootstrap Ignition Config from target directory
INFO Consuming Worker Ignition Config from target directory
INFO Consuming Master Ignition Config from target directory
...output omitted...
INFO Creating infrastructure resources... ①
DEBUG module.vpc.aws_vpc.new_vpc[0]: Creating...
```

```

DEBUG module.bootstrap.aws_iam_role.bootstrap: Creating...
DEBUG module.masters.aws_iam_role.master_role: Creating...
DEBUG module.bootstrap.aws_s3_bucket.ignition: Creating...
DEBUG module.vpc.aws_vpc_dhcp_options.main[0]: Creating...
...output omitted...
DEBUG Apply complete! Resources: 122 added, 0 changed, 0 destroyed.
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
INFO Waiting up to 20m0s for the Kubernetes API at
https://api.ocp4-aws.example.com:6443... 2
INFO API v1.19.0+9f84db3 up
INFO Waiting up to 30m0s for bootstrapping to complete... 3
...output omitted...
DEBUG Bootstrap status: complete
INFO Destroying the bootstrap resources...
...output omitted...
INFO Waiting up to 40m0s for the cluster at https://api.ocp4-aws.example.com:6443
to initialize... 4
DEBUG Still waiting for the cluster to initialize: Working towards 4.6.4: 82%
complete
...output omitted...
DEBUG Cluster is initialized
INFO Waiting up to 10m0s for the openshift-console route to be created...
...output omitted...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/user/ocp4-aws-cluster/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-console.apps.ocp4-aws.example.com
INFO Login to the console with user: "kubeadmin", and password: "xxxx"
...output omitted...
INFO Time elapsed: 34m11s

```

- ① Phase der AWS-Ressourcenerstellung
- ② Bootstrap-Phase (Bootkube)
- ③ Bootstrap-Phase (temporäre Control Plane)
- ④ Produktions-Control Plane-Phase

Überwachen des OpenShift-Installationsprozesses

Um eine OpenShift-Installation zu überwachen, müssen Sie das im Kapitel *Beschreiben des OpenShift-Installationsprozesses* erläuterte Verfahren befolgen.

Nachfolgend finden Sie eine nützliche Sequenz von oc-Befehlen zur Überwachung des Installationsprozesses:

```
[user@demo ~]$ export KUBECONFIG=${HOME}/ocp4-aws-cluster/auth/kubeconfig
```

```
[user@demo ~]$ watch 'oc get clusterversio; oc get clusteroperators; \
> oc get pods --all-namespaces | grep -v -E "Running|Completed"; oc get nodes'
NAME      VERSION AVAILABLE PROGRESSING SINCE STATUS
```

version	False	True	13m	Working towards 4.6.4: 98% complete				
NAME				VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication				4.6.4	False	False	True	7m59s
cloud-credential					True	False	False	11m
cluster-autoscaler				4.6.4	True	False	False	6m34s
config-operator				4.6.4	True	False	False	8m4s
console				4.6.4	False	True	False	60s
csi-snapshot-controller				4.6.4	True	False	False	7m55s
dns				4.6.4	True	False	False	7m18s
etcd				4.6.4	True	True	False	6m39s
<i>...output omitted...</i>								
NAMESPACE		NAME		READY	STATUS	RESTARTS	AGE	
openshift-apiserver		apiserver-5c4b794d87	0/2	Pending	0	0	2m13s	
openshift-authentication		oauth Openshift-7bfcc	0/1	Terminating	0	0	58s	
openshift-kube-apiserver		kube-apiserver-ip-10	0/5	Init:0/1	0	0	61s	
NAME		STATUS	ROLES	AGE	VERSION			
ip-10-0-143-122.us-east-2		Ready	master	8m53s	v1.19.0+9f84db3			
ip-10-0-148-117.us-east-2		Ready	worker	99s	v1.19.0+9f84db3			
ip-10-0-167-115.us-east-2		Ready	worker	98s	v1.19.0+9f84db3			
ip-10-0-186-189.us-east-2		Ready	master	8m59s	v1.19.0+9f84db3			
ip-10-0-208-252.us-east-2		Ready	master	8m58s	v1.19.0+9f84db3			
ip-10-0-220-24.us-east-2		Ready	worker	98s	v1.19.0+9f84db3			

Der Befehl „oc get events“ gibt die OpenShift-Ereignismeldungen in Echtzeit aus, daher ist er hilfreich, um den Installationsfortschritt und Hintergrundereignisse detailliert zu überwachen.

```
[user@demo ~]$ oc get events -A -w
...output omitted...
openshift-infra      0s      Warning   NetworkNotReady
```

Fehlersuche beim OpenShift-Installationsvorgang

Der Fehlerbehebungsprozess hängt von der Phase ab, in der die OpenShift-Installation fehlschlägt.

Fehlerbehebung in der Phase der AWS-Ressourcenerstellung

In dieser Phase werden beim OpenShift-Installationsprozess AWS-Ressourcen erstellt. Nachfolgend finden Sie die relevanten Installationsprotokolle in dieser Phase.

```
[user@demo ~]$ openshift-install create cluster \
> --dir=${HOME}/ocp4-aws-cluster --log-level=debug
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
...output omitted...
INFO Consuming Bootstrap Ignition Config from target directory
INFO Consuming Worker Ignition Config from target directory
INFO Consuming Master Ignition Config from target directory
...output omitted...
```

```
INFO Creating infrastructure resources...
❶
...output omitted...
```

- ❶ Ort der Fehlermeldung im Protokollschnitt `Creating infrastructure resources`.

Sie können das AWS CLI-Tool `aws` verwenden, um die erstellten Ressourcen zu überprüfen.

```
[user@demo ~]$ aws ec2 describe-instances \
> --filters Name=tag:Name,Values="*ocp4-aws*" --output table
```

Sie können die AWS Web Console verwenden, um die erstellten Ressourcen zu überprüfen.

Fehler in dieser Phase hängen in der Regel mit den AWS-Voraussetzungen zusammen, die vor dem Starten des OpenShift-Installationsprozesses erforderlich sind:

- Router53-DNS-Domain
- AWS-Kontoanmeldedaten
- Beschränkungen und Berechtigungen von AWS-Konten

Beheben von Fehlern in der Bootstrap-Phase (Bootkube)

In dieser Phase geschieht Folgendes:

- Der OpenShift-Installationsprozess hat den Bootstrap-Knoten bereitgestellt.
- Der auf dem Bootstrap-Knoten ausgeführte systemd-Service `release-image` lädt die Container-Images herunter, die zum Starten der temporären Control Plane erforderlich sind.
- Der auf dem Bootstrap-Knoten ausgeführte systemd-Service `bootkube` startet die temporäre Control Plane.
- Der OpenShift-Installationsprozess wartet, bis die Kubernetes-API, die auf dem Bootstrap-Knoten ausgeführt wird, verfügbar wird.

Nachfolgend finden Sie die relevanten Installationsprotokolle in dieser Phase.

```
[user@demo ~]$ openshift-install create cluster \
> --dir=${HOME}/ocp4-aws-cluster --log-level=debug
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
...output omitted...
INFO Consuming Bootstrap Ignition Config from target directory
INFO Consuming Worker Ignition Config from target directory
INFO Consuming Master Ignition Config from target directory
...output omitted...
INFO Creating infrastructure resources...
DEBUG module.vpc.aws_vpc.new_vpc[0]: Creating...
DEBUG module.bootstrap.aws_iam_role.bootstrap: Creating...
DEBUG module.masters.aws_iam_role.master_role: Creating...
DEBUG module.bootstrap.aws_s3_bucket.ignition: Creating...
DEBUG module.vpc.aws_vpc_dhcp_options.main[0]: Creating...
...output omitted...
DEBUG Apply complete! Resources: 122 added, 0 changed, 0 destroyed.
DEBUG OpenShift Installer 4.6.4
```

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

```
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
INFO Waiting up to 20m0s for the Kubernetes API at
https://api.ocp4-aws.example.com:6443...
①
...output omitted...
```

- ① Ort der Fehlermeldung im Protokollschnitt `Waiting for the Kubernetes API`.

Zur Fehlerbehebung in dieser Phase müssen Sie die Protokolle vom Bootstrap-Knoten abrufen. Sie können sich beim Bootstrap-Knoten mit SSH über den Bastion-Host anmelden und die Journal- und Container-Protokolle überprüfen. Sie können die öffentliche IP des Bootstrap-Knotens mit dem AWS-CLI-Tool `aws` abrufen. Anschließend können Sie mit dem für den Installationsprozess generierten SSH-Schlüssel eine Verbindung zum Bootstrap-Knoten herstellen.

```
[user@demo ~]$ aws ec2 describe-instances \
> --filters Name>tag:Name,Values="*ocp4-aws*" --output table \
> | grep -B30 -i bootstrap | grep PublicDnsName
PublicDnsName      | ec2-3-21-104-144.us-east-2.compute.amazonaws.com
```

```
[user@demo ~]$ ssh -i ${HOME}/.ssh/ocp4-aws-key
core@ec2-3-17-184-44.us-east-2.compute.amazonaws.com
```

**Anmerkung**

Im Rahmen des Installationsprozesses wird der Bootstrap-Knoten mit einer öffentlichen AWS-IP (EIP) bereitgestellt. Nach Abschluss der OpenShift-Installation entfernt das Installationsprogramm den Bootstrap-Knoten.

Führen Sie nach der Anmeldung beim Bootstrap-Knoten den folgenden Befehl aus, um die Services `release-image` und `bootkube` zu debuggen:

```
sh-4.2# journalctl -b -f -u release-image.service -u bootkube.service
```

Führen Sie außerdem den Befehl `crictl` auf dem Bootstrap-Knoten aus, um nach fehlgeschlagenen Containern zu suchen und ihre Protokolle auszugeben:

```
sh-4.2# sudo bash
sh-4.2# crictl ps -a
sh-4.2# crictl logs <container_id>
```

Beheben von Fehlern in der Bootstrap-Phase (temporäre Control Plane)

In dieser Phase geschieht Folgendes:

- Die Kubernetes-API und die temporäre Control Plane werden auf dem Bootstrap-Knoten ausgeführt.
- Die Knoten der Control Plane beenden ihre Installation mit den vom MCS abgerufenen Ignition-Dateien, die auf der temporären Kontrollsicht ausgeführt werden.

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

- Der Bootstrap-Knoten überträgt die temporäre Control Plane auf die Produktions-Control Plane, die auf den Control Plane-Knoten ausgeführt wird.
- Der OpenShift-Installationsprozess wartet auf den Abschluss des Bootstrap-Prozesses.

Nachfolgend finden Sie die relevanten Installationsprotokolle in dieser Phase.

```
[user@demo ~]$ openshift-install create cluster \
> --dir=${HOME}/ocp4-aws-cluster --log-level=debug
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
...output omitted...
INFO Consuming Bootstrap Ignition Config from target directory
INFO Consuming Worker Ignition Config from target directory
INFO Consuming Master Ignition Config from target directory
...output omitted...
INFO Creating infrastructure resources...
DEBUG module.vpc.aws_vpc.new_vpc[0]: Creating...
DEBUG module.bootstrap.aws_iam_role.bootstrap: Creating...
DEBUG module.masters.aws_iam_role.master_role: Creating...
DEBUG module.bootstrap.aws_s3_bucket.ignition: Creating...
DEBUG module.vpc.aws_vpc_dhcp_options.main[0]: Creating...
...output omitted...
DEBUG Apply complete! Resources: 122 added, 0 changed, 0 destroyed.
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
INFO Waiting up to 20m0s for the Kubernetes API at https://api.ocp4-
aws.example.com:6443...
INFO API v1.19.0+9f84db3 up
INFO Waiting up to 30m0s for bootstrapping to complete...
1
...output omitted...
```

- ➊ Ort der Fehlermeldung im Protokollschnitt **Waiting for bootstrapping to complete**.

Zur Fehlerbehebung müssen Sie wie im vorherigen Abschnitt die Protokolle vom Bootstrap-Knoten abrufen. In dieser Phase können Sie auch das OpenShift-Installationsprogramm verwenden, um die Installationsprotokolle von den Bootstrap- und Control Plane-Knoten abzurufen.

```
[user@demo ~]$ export KUBECONFIG=${HOME}/ocp4-aws-cluster/auth/kubeconfig
```

```
[user@demo ~]$ openshift-install gather bootstrap \
> --dir=${HOME}/ocp4-aws-cluster
INFO Pulling debug logs from the bootstrap machine
INFO Bootstrap gather logs captured here "/home/user/ocp4-aws-cluster/log-
bundle-20210107135825.tar.gz"
```

```
[user@demo ~]$ cd ocp4-aws-cluster  
[user@demo ~]$ tar -xvzf log-bundle-20210107135825.tar.gz  
[user@demo ~]$ cd log-bundle-20210107135825  
[user@demo ~]$ ls  
bootstrap/ control-plane/ failed-units.txt rendered-assets/ resources/ unit-  
status/
```

Sie können auch auf die Kubernetes-API zugreifen, die auf dem Bootstrap-Knoten ausgeführt wird, um den Fortschritt der Cluster-Installation überwachen. Verwenden Sie zur Fehlerbehebung die oc-Standardbefehle auf dem Bastion-Host.

```
[user@demo ~]$ export KUBECONFIG=${HOME}/ocp4-aws-cluster/auth/kubeconfig
```

```
[user@demo ~]$ oc get nodes  
ip-10-0-143-122.us-east-2.compute.internal Ready master 7h43m  
v1.19.0+9f84db3  
ip-10-0-186-189.us-east-2.compute.internal Ready master 7h44m  
v1.19.0+9f84db3  
ip-10-0-208-252.us-east-2.compute.internal Ready master 7h44m  
v1.19.0+9f84db3
```

```
[user@demo ~]$ oc get clusterversion  
NAME VERSION AVAILABLE PROGRESSING SINCE STATUS  
version False True 8m32s Unable to apply 4.6.4: an unknown  
error has occurred: MultipleErrors
```

```
[user@demo ~]$ oc get clusteroperators  
NAME VERSION AVAILABLE PROGRESSING DEGRADED SINCE  
authentication  
cloud-credential True False False 10m  
cluster-autoscaler  
config-operator  
console  
csi-snapshot-controller  
dns  
etcd 4.6.4 Unknown Unknown False 1s  
...output omitted...
```

Sie können den Installationsprozess detailliert verfolgen, indem Sie die Cluster-Ereignisse beobachten.

```
[user@demo ~]$ oc get events -A -w  
...output omitted...  
openshift-infra 0s Warning NetworkNotReady
```

In dieser Phase können Sie auch den Befehl `oc adm node-logs` verwenden, um die Cluster-Knotenprotokolle zu erfassen.

```
[user@demo ~]$ oc adm node-logs ip-10-0-131-192.us-east-2.compute.internal
...output omitted...
Jan 06 22:31:58.320538 ip-10-0-131-192 hyperkube[1566]: I0106 22:31:58.320538
1566 prober.go:126] Liveness probe for "aws-ebs-csi-driver-node-hzcb8_openshift-
cluster-csi-drivers(96c559fe-67c7-4072-9311-450dfd2f8ddb):csi-driver" succeeded
```

Alternativ können Sie den Befehl `oc debug node` zur Fehlerbehebung auf den Control Plane-Knoten verwenden. Nach der Anmeldung können Sie die gleichen Befehle verwenden, die Sie für die Fehlerbehebung beim Bootstrap-Knoten verwendet haben.

```
[user@demo ~]$ oc get nodes
ip-10-0-143-122.us-east-2.compute.internal Ready master 7h43m
v1.19.0+9f84db3
ip-10-0-186-189.us-east-2.compute.internal NotReady master 7h44m
v1.19.0+9f84db3
ip-10-0-208-252.us-east-2.compute.internal Ready master 7h44m
v1.19.0+9f84db3
```

```
[user@demo ~]$ oc debug node/ip-10-0-186-189.us-east-2.compute.internal
...output omitted...
sh-4.2# chroot /host
sh-4.2# journalctl -b -f -u kubelet.service -u crio.service
sh-4.2# sudo bash
sh-4.2# crictl ps -a
sh-4.2# crictl logs <container_id>
```

Beheben von Fehlern in der Produktions-Control Plane-Phase

In dieser Phase geschieht Folgendes:

- Der OpenShift-Installationsprozess hat die temporäre Control Plane auf die Produktions-Control Plane übertragen, die auf den Control Plane-Knoten ausgeführt wird.
- Der OpenShift-Installationsprozess hat den Bootstrap-Knoten entfernt, und die Installation wird mithilfe der Produktions-Control Plane fortgesetzt.
- Der Cluster Version Operator (CVO), der auf der Produktions-Control Plane ausgeführt wird, installiert alle Operatoren, die den OpenShift-Cluster erstellen, und beendet anschließend die Installation.
- Der OpenShift-Installationsprozess wartet, bis der CVO den Abschluss der Cluster-Installation meldet.

Nachfolgend finden Sie die relevanten Installationsprotokolle in dieser Phase.

```
[user@demo ~]$ openshift-install create cluster \
> --dir=${HOME}/ocp4-aws-cluster --log-level=debug
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
...output omitted...
INFO Consuming Bootstrap Ignition Config from target directory
INFO Consuming Worker Ignition Config from target directory
INFO Consuming Master Ignition Config from target directory
```

```
...output omitted...
INFO Creating infrastructure resources...
DEBUG module.vpc.aws_vpc.new_vpc[0]: Creating...
DEBUG module.bootstrap.aws_iam_role.bootstrap: Creating...
DEBUG module.masters.aws_iam_role.master_role: Creating...
DEBUG module.bootstrap.aws_s3_bucket.ignition: Creating...
DEBUG module.vpc.aws_vpc_dhcp_options.main[0]: Creating...
...output omitted...
DEBUG Apply complete! Resources: 122 added, 0 changed, 0 destroyed.
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
INFO Waiting up to 20m0s for the Kubernetes API at https://api.ocp4-
aws.example.com:6443...
INFO API v1.19.0+9f84db3 up
INFO Waiting up to 30m0s for bootstrapping to complete...
...output omitted...
DEBUG Bootstrap status: complete
INFO Destroying the bootstrap resources...
...output omitted...
INFO Waiting up to 40m0s for the cluster at https://api.ocp4-aws.example.com:6443
to initialize...
DEBUG Still waiting for the cluster to initialize: Working towards 4.6.4: 82%
complete
...output omitted...
①
...output omitted...
```

- ① Ort der Fehlermeldung im Protokollschnitt **Waiting for the cluster to initialize**.

In dieser Phase können Sie dieselben Techniken zur Fehlerbehebung anwenden, die im vorherigen Abschnitt erläutert wurden. In diesem Fall werden die Cluster-Control Plane und die Kubernetes-API auf den Control Plane-Knoten ausgeführt.



Anmerkung

Weitere Informationen finden Sie im Abschnitt *Troubleshooting installation issues* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing.

Installieren von OpenShift mit Full-Stack-Automatisierung auf AWS

Videoergebnis:

Installieren von OpenShift mit Anpassungen unter Verwendung von Full-Stack-Automatisierung auf AWS



Literaturhinweise

- Weitere Informationen finden Sie im Abschnitt *Installing a cluster on AWS with customizations* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter
https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_aws

► Quiz

Demonstrieren der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Wenn Sie die Installationskonfigurationsdatei `install-config.yaml` für eine OpenShift-Installation auf AWS mit dem Befehl `openshift-install` generieren, welche drei der folgenden Cluster-Informationselemente müssen Sie bereitstellen? (Wählen Sie drei Antworten aus.)
 - a. Die IP-Adresse des Bastionhosts
 - b. Die AWS-Region des Clusters
 - c. AWS VZ des Clusters
 - d. Die Basis-Domain des Clusters
 - e. Cluster-Name

- ▶ 2. Welche drei der folgenden Phasen sind Phasen bei der Installation von OpenShift auf AWS? (Wählen Sie drei Antworten aus.)
 - a. Phase der AWS-Ressourcenerstellung
 - b. Bootstrap-Phase
 - c. Berechtigungsphase für AWS-Konten
 - d. Produktions-Control Plane-Phase
 - e. Authentifizierungsphase für AWS-Konten

- ▶ 3. Welche zwei der folgenden Befehle sind hilfreich für die Überwachung des OpenShift-Installationsprozesses? (Wählen Sie zwei Antworten aus.)
 - a. `watch 'oc get clusterversion; oc get clusteroperators; oc get pods --all-namespaces | grep -v -E "Running|Completed"; oc get nodes'`
 - b. `aws ec2 describe-instances --output table`
 - c. `oc get events -A -w`
 - d. `oc monitor cluster`

- ▶ 4. Welche der folgenden Befehle sind bei der Fehlerbehebung bei einer OpenShift-Installation nützlich für die Anmeldung bei den Cluster-Knoten?
 - a. `openshift-install gather bootstrap`
 - b. `oc adm node-logs`
 - c. `oc debug node`

► **5. Welche zwei der folgenden Standardkonfigurationen werden vom
Installationsprogramm bei der Installation von OpenShift auf AWS mit der Full-Stack-
Automatisierungsmethode verwendet? (Wählen Sie zwei Antworten aus.)**

- a. Die Cluster-Knoteninstanzen verwenden AWS EBS gp2-Volumes für die System-Disks.
- b. Die Cluster-Knoteninstanzen verwenden AWS EBS io2-Volumes für die System-Disks.
- c. Das Cluster-Netzwerk hostet bis zu 512 Knoten.
- d. Das Cluster-Netzwerk hostet bis zu 2048 Knoten.

► Lösung

Demonstrieren der Installation von OpenShift auf AWS mithilfe der Full-Stack-Automatisierungsmethode

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Wenn Sie die Installationskonfigurationsdatei `install-config.yaml` für eine OpenShift-Installation auf AWS mit dem Befehl `openshift-install` generieren, welche drei der folgenden Cluster-Informationselemente müssen Sie bereitstellen? (Wählen Sie drei Antworten aus.)
 - a. Die IP-Adresse des Bastionhosts
 - b. Die AWS-Region des Clusters
 - c. AWS VZ des Clusters
 - d. Die Basis-Domain des Clusters
 - e. Cluster-Name

- ▶ 2. Welche drei der folgenden Phasen sind Phasen bei der Installation von OpenShift auf AWS? (Wählen Sie drei Antworten aus.)
 - a. Phase der AWS-Ressourcenerstellung
 - b. Bootstrap-Phase
 - c. Berechtigungsphase für AWS-Konten
 - d. Produktions-Control Plane-Phase
 - e. Authentifizierungsphase für AWS-Konten

- ▶ 3. Welche zwei der folgenden Befehle sind hilfreich für die Überwachung des OpenShift-Installationsprozesses? (Wählen Sie zwei Antworten aus.)
 - a. `watch 'oc get clusterversion; oc get clusteroperators; oc get pods --all-namespaces | grep -v -E "Running|Completed"; oc get nodes'`
 - b. `aws ec2 describe-instances --output table`
 - c. `oc get events -A -w`
 - d. `oc monitor cluster`

- ▶ 4. Welche der folgenden Befehle sind bei der Fehlerbehebung bei einer OpenShift-Installation nützlich für die Anmeldung bei den Cluster-Knoten?
 - a. `openshift-install gather bootstrap`
 - b. `oc adm node-logs`
 - c. `oc debug node`

► **5. Welche zwei der folgenden Standardkonfigurationen werden vom
Installationsprogramm bei der Installation von OpenShift auf AWS mit der Full-Stack-
Automatisierungsmethode verwendet? (Wählen Sie zwei Antworten aus.)**

- a. Die Cluster-Knoteninstanzen verwenden AWS EBS gp2-Volumes für die System-Disks.
- b. Die Cluster-Knoteninstanzen verwenden AWS EBS io2-Volumes für die System-Disks.
- c. Das Cluster-Netzwerk hostet bis zu 512 Knoten.
- d. Das Cluster-Netzwerk hostet bis zu 2048 Knoten.

Überprüfen der Installation von OpenShift auf AWS

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, den Erfolg einer Installation von OpenShift auf AWS zu bewerten.

Überprüfen von OpenShift-Installationen auf AWS

Nachdem die OpenShift-Installation auf AWS erfolgreich abgeschlossen wurde, müssen Administratoren sicherstellen, dass der installierte Cluster fehlerfrei ist und für Day 2-Aufgaben zum Onboarding von Benutzern und Anwendungen bereit ist. Administratoren müssen die im Kapitel *Beschreiben des OpenShift-Installationsprozesses* erläuterten Schritte für die Cluster-Überprüfung befolgen.

Integrität des OpenShift-Clusters

Auf dem Bastion-Host können Sie mit dem Befehl `oc` eine einfache Integritätsprüfung durchführen.

- Konfigurieren Sie die Umgebungsvariable `KUBECONFIG` so, dass sie sich mit `cluster-admin`-Berechtigungen bei der OpenShift-API authentifiziert.

```
[user@demo ~]$ export KUBECONFIG=${HOME}/ocp4-aws-cluster/auth/kubeconfig
```

- Überprüfen Sie, ob die Systemuhr aller Cluster-Knoten mit einem NTP-Server (Network Time Protocol) synchronisiert wird.

```
[user@demo ~]$ oc debug node/ip-10-0-151-66.us-east-2.compute.internal
...output omitted...
sh-4.4# chroot /host
sh-4.4# cat /etc/chrony.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
pool 2.rhel.pool.ntp.org iburst
...output omitted...

sh-4.4# sudo chronyc tracking
Reference ID      : 2D574C03 (ntp.devrandom.be)
Stratum          : 3
Ref time (UTC)   : Sun Feb  7 18:36:05 2021
System time      : 0.000073610 seconds slow of NTP time
Last offset      : -0.000169116 seconds
RMS offset       : 0.000708059 seconds
Frequency        : 12.811 ppm fast
Residual freq    : +0.025 ppm
Skew             : 0.708 ppm
Root delay       : 0.091646813 seconds
```

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

```
Root dispersion : 0.001100251 seconds
Update interval : 65.1 seconds
Leap status      : Normal
```

Der auf dem Cluster-Knoten ausgeführte systemd-Service `chrony` verwendet den NTP-Pool `2.rhel.pool.ntp.org`. Die Systemuhr wird mit dem NTP-Server `ntp.devrandom.be` synchronisiert.

Wiederholen Sie diesen Vorgang auf allen Cluster-Knoten.

- Überprüfen Sie, ob sich alle Cluster-Knoten im Status Ready befinden.

Wenn ein Cluster-Knoten sich nicht im Status Ready befindet, kann er nicht mit der OpenShift-Control Plane kommunizieren und ist für den Cluster nicht verfügbar.

```
[user@demo ~]$ oc get nodes
NAME                               STATUS   ROLES   AGE VERSION
ip-10-0-153-185.us-east-2.compute.internal  Ready   worker  22h v1.19.0+9f84db3
ip-10-0-156-69.us-east-2.compute.internal  Ready   master  22h v1.19.0+9f84db3
ip-10-0-175-189.us-east-2.compute.internal  Ready   master  22h v1.19.0+9f84db3
ip-10-0-184-156.us-east-2.compute.internal  Ready   worker  22h v1.19.0+9f84db3
ip-10-0-195-237.us-east-2.compute.internal  Ready   worker  22h v1.19.0+9f84db3
ip-10-0-220-19.us-east-2.compute.internal  Ready   master  22h v1.19.0+9f84db3
```

- Überprüfen Sie, ob alle Cluster-Knoten Nutzungsmetriken melden.

```
[user@demo ~]$ oc adm top node
NAME                               CPU(cores)  CPU%  MEMORY(bytes)  MEMORY%
ip-10-0-153-185.us-east-2.compute.internal  130m        0%    1982Mi        3%
ip-10-0-156-69.us-east-2.compute.internal  742m        9%    5457Mi       17%
ip-10-0-175-189.us-east-2.compute.internal  597m        7%    4004Mi       13%
ip-10-0-184-156.us-east-2.compute.internal  146m        0%    2574Mi        4%
ip-10-0-195-237.us-east-2.compute.internal  851m        5%    6135Mi       9%
ip-10-0-220-19.us-east-2.compute.internal  618m        8%    4445Mi      14%
```

- Stellen Sie sicher, dass keine Anforderungen für die Signierung des Zertifikats (Certificate Signing Requests, CSRs) auf Genehmigung warten.

```
[user@demo ~]$ oc get csr | grep Pending
```

- Überprüfen Sie, ob der OpenShift-Cluster im CVO-Bericht (Cluster Version Operator) als verfügbar und bereit aufgeführt wird.

```
[user@demo ~]$ oc get clusterversion
NAME      VERSION  AVAILABLE  PROGRESSING  SINCE  STATUS
version   4.6.4    True       False        22h    Cluster version is 4.6.4
```

- Überprüfen Sie, ob alle Cluster-Operatoren verfügbar und bereit sind.

Wenn der Cluster fehlerfrei ist, sollten alle Cluster-Operatoren verfügbar sein und nicht weiter ausgeführt werden, es sei denn, seine Konfiguration wird noch von mindestens einem Operator weiterhin angewendet.

```
[user@demo ~]$ oc get clusteroperators
NAME          VERSION AVAILABLE PROGRESSING DEGRADED SINCE
authentication 4.6.4   True    False     False    22h
cloud-credential 4.6.4   True    False     False    22h
cluster-autoscaler 4.6.4   True    False     False    22h
config-operator 4.6.4   True    False     False    22h
console        4.6.4   True    False     False    22h
csi-snapshot-controller 4.6.4   True    False     False    22h
dns            4.6.4   True    False     False    22h
etcd           4.6.4   True    False     False    22h
...output omitted...
```

- Stellen Sie sicher, dass der Cluster keine Pods mit Planungs- oder Ausführungsproblemen enthält.

```
[user@demo ~]$ oc get pods --all-namespaces | grep -v -E 'Running|Completed'
NAMESPACE  NAME      READY  STATUS    RESTARTS AGE
```

Integrität des OpenShift-etcd-Clusters

- Stellen Sie sicher, dass alle Mitglieder des etcd-Clusters fehlerfrei sind.

```
[user@demo ~]$ oc get pods -n openshift-etcd | grep etcd-ip
etcd-ip-10-0-156-69.us-east-2.compute.internal 3/3 Running 0 22h
etcd-ip-10-0-175-189.us-east-2.compute.internal 3/3 Running 0 22h
etcd-ip-10-0-220-19.us-east-2.compute.internal 3/3 Running 0 22h
```

```
[user@demo ~]$ oc rsh -n openshift-etcd \
> etcd-ip-10-0-156-69.us-east-2.compute.internal
sh-4.4# etcdctl endpoint health --cluster
https://10.0.156.69:2379 is healthy: successfully committed proposal: took=10.8ms
https://10.0.175.189:2379 is healthy: successfully committed proposal: took=11.8ms
https://10.0.220.19:2379 is healthy: successfully committed proposal: took=12.1ms
```

Integrität von OpenShift-API und -Konsole

- Überprüfen Sie, ob der OpenShift-API-DNS-Eintrag api.ocp4-aws.example.com für die Verwendung der öffentlichen OpenShift-API-ELB-IPs konfiguriert ist.

```
[user@demo ~]$ dig api.ocp4-aws.example.com
...output omitted...
;; ANSWER SECTION:
api.ocp4-aws.example.com. 60 IN A 3.139.205.90
api.ocp4-aws.example.com. 60 IN A 18.216.12.234
api.ocp4-aws.example.com. 60 IN A 3.131.198.170
...output omitted...
```

```
[user@demo ~]$ aws elbv2 describe-load-balancers | grep DNSName
"DNSName": "ocp4-aws-n6768-ext-09c36c5873bc4193.elb.us-east-2.amazonaws.com",
"DNSName": "ocp4-aws-n6768-int-dc74cad870d449a1.elb.us-east-2.amazonaws.com",
```

```
[user@demo ~]$ host \
> ocp4-aws-n6768-ext-09c36c5873bc4193.elb.us-east-2.amazonaws.com
ocp4-aws-n6768-ext-09c36c5873bc4193.elb.us-east-2.amazonaws.com has
  address 18.216.12.234
ocp4-aws-n6768-ext-09c36c5873bc4193.elb.us-east-2.amazonaws.com has
  address 3.131.198.170
ocp4-aws-n6768-ext-09c36c5873bc4193.elb.us-east-2.amazonaws.com has
  address 3.139.205.90
```

- Stellen Sie sicher, dass die OpenShift-API verfügbar ist, indem Sie die Kubernetes-Version anfordern.

```
[user@demo ~]$ curl -k https://api.ocp4-aws.example.com:6443/version
...output omitted...
"gitVersion": "v1.19.0+9f84db3",
...output omitted...
```

- Prüfen Sie, ob Sie eine Verbindung zur OpenShift-Konsole herstellen können.

```
[user@demo ~]$ curl -kIs \
> https://console-openshift-console.apps.ocp4-aws.example.com
...output omitted...
HTTP/1.1 200 OK
...output omitted...
```

```
[user@demo ~]$ firefox https://console-openshift-console.apps.ocp4-aws.example.com
```

Integrität der OpenShift-Registry

- Stellen Sie sicher, dass die Anzahl der internen Registry-Pods, die auf dem OpenShift-Cluster ausgeführt werden, mit der Bereitstellungskonfiguration übereinstimmt.

```
[user@demo ~]$ oc -n openshift-image-registry get deployment.apps/image-registry
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
image-registry 2/2     2           2           24h
```

- Wenn mehrere Server-Knoten vorhanden sind, überprüfen Sie, ob jeder Registry-Pod auf einem anderen Server-Knoten ausgeführt wird.

```
[user@demo ~]$ oc -n openshift-image-registry get pods -o wide | grep -i
^image-registry
image-registry-64889f9585-ngmkv           1/1     Running    0
6d20h  10.132.0.13  ip-10-0-215-243.us-east-2
image-registry-64889f9585-w6445           1/1     Running    0
6d20h  10.131.0.6  ip-10-0-133-26.us-east-2
```

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

- Überprüfen Sie auf jedem Cluster-Knoten die Integrität der internen Registry.

```
[user@demo ~]$ oc debug node/ip-10-0-184-156.us-east-2.compute.internal
sh-4.4# chroot /host
sh-4.4# curl -kIs \
> https://image-registry.openshift-image-registry.svc:5000/healthz
...output omitted...
HTTP/2 200
...output omitted...
```

- Überprüfen Sie, ob die Bereitstellung der internen Registry persistenten AWS S3-Storage verwendet. Stellen Sie außerdem sicher, dass sich der Image-Registry-Operator im Verwaltungsstatus Managed befindet.

```
[user@demo ~]$ oc get configs.imageregistry.operator.openshift.io cluster -o yaml
...output omitted...
spec:
  managementState: Managed
  ...output omitted...
  storage:
    s3:
      bucket: ocp4-aws-n6768-image-registry-us-east-2-uklufirxkteqjuxyjjiuae
      encrypt: true
      region: us-east-2
      virtualHostedStyle: false
  ...output omitted...
```

Integrität von OpenShift Ingress

- Überprüfen Sie, ob der DNS-Platzhaltereintrag für Anwendungen, *.apps.ocp4-aws.example.com, für die Verwendung der öffentlichen Ingress-ELB-IPs konfiguriert ist.

```
[user@demo ~]$ dig test.apps.ocp4-aws.example.com
...output omitted...
;; ANSWER SECTION:
test.apps.ocp4-aws.example.com. 60 IN A 3.129.28.60
test.apps.ocp4-aws.example.com. 60 IN A 18.221.163.32
...output omitted...
```

```
[user@demo ~]$ aws elb describe-load-balancers | grep DNSName
"DNSName":
  "ad3c569731b15486a8f31214c5a2b6be-1651370221.us-east-2.elb.amazonaws.com",

[user@demo ~]$ host \
> ad3c569731b15486a8f31214c5a2b6be-1651370221.us-east-2.elb.amazonaws.com
ad3c569731b15486a8f31214c5a2b6be-1651370221.us-east-2.elb.amazonaws.com has
  address 3.129.28.60
ad3c569731b15486a8f31214c5a2b6be-1651370221.us-east-2.elb.amazonaws.com has
  address 18.221.163.32
```

- Stellen Sie sicher, dass Sie auf eine über eine OpenShift Ingress-Route verfügbar gemachte Anwendung zugreifen können.

```
[user@demo ~]$ oc get routes -A | grep downloads
openshift-console downloads downloads-openshift-console.apps.ocp4-aws.example.com
```

```
[user@demo ~]$ curl -kIs \
> https://downloads-openshift-console.apps.ocp4-aws.example.com
...output omitted...
HTTP/1.0 200 OK
...output omitted...
```

- Stellen Sie sicher, dass die Anzahl der Router-Pods, die auf dem OpenShift-Cluster ausgeführt werden, mit der Bereitstellungskonfiguration übereinstimmt.

```
[user@demo ~]$ oc -n openshift-ingress get deployment.apps/router-default
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
router-default  2/2     2            2           24h
```

- Wenn mehrere Server-Knoten vorhanden sind, überprüfen Sie, ob jeder Router-Pod auf einem anderen Server-Knoten ausgeführt wird.

```
[user@demo ~]$ oc -n openshift-ingress get pods -o wide
NAME                           READY   STATUS    RESTARTS   AGE   IP
  NODE
router-default-6b5dcc7d88-5sjdp  1/1    Running   0          6d20h  10.131.0.4
  ip-10-0-133-26.us-east-2
router-default-6b5dcc7d88-bn62l  1/1    Running   0          6d20h  10.132.0.4
  ip-10-0-215-243.us-east-2
```

Integrität des dynamischen OpenShift-Storage-Anbieters

- Überprüfen Sie den Status der AWS EBS-Storage-Klasse gp2.

```
[user@demo ~]$ oc get sc
NAME          PROVISIONER          RECLAIMPOLICY  BINDINGMODE      EXPANSION  AGE
gp2 (default) kubernetes.io/aws-ebs  Delete        WaitForFirstConsumer  true       32m
gp2-csi       ebs.csi.aws.com      Delete        WaitForFirstConsumer  true       32m
```

Die Storage-Klasse „gp2“ verwendet den Volume-Bindungsmodus `WaitForFirstConsumer`. Dieser Volume-Bindungsmodus verzögert die Bindung und Bereitstellung eines `PersistentVolume`, bis ein Pod erstellt wird, der `PersistentVolumeClaim` verwendet. Diese Konfiguration ist für diese Storage-Klasse unveränderlich.

- Überprüfen Sie, ob die Storage-Klasse gp2 wie erwartet funktioniert.

Erstellen Sie eine einfache httpd-Anwendung, die persistenten Storage für das Verzeichnis `DocumentRoot` unter `/var/www/html` verwendet.

```
[user@demo ~]$ oc new-project httpd-persistent
[user@demo ~]$ cat /tmp/httpd-persistent.yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
```

```
metadata:
  name: httpd-claim
  namespace: httpd-persistent
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
---
apiVersion: v1
kind: Pod
metadata:
  name: httpd
  namespace: httpd-persistent
spec:
  containers:
  - image: registry.redhat.io/rhel8/httpd-24:latest
    name: httpd
    ports:
    - containerPort: 8080
      name: http
      protocol: TCP
    volumeMounts:
    - mountPath: /var/www/html
      name: httpd-claim
  volumes:
  - name: httpd-claim
    persistentVolumeClaim:
      claimName: httpd-claim

[user@demo ~]$ oc create -f /tmp/httpd-persistent.yaml
persistentvolumeclaim/httpd-claim created
pod/httpd created
```

Überprüfen Sie, ob die PVC-Erstellung automatisch die PV-Bereitstellung und -Bindung durch die standardmäßige Storage-Klasse gp2 auslöst.

```
[user@demo ~]$ oc get pvc
NAME      STATUS   VOLUME      CAPACITY   ACCESS MODES   STORAGECLASS   AGE
httpd-claim  Bound   pvc-d965cb1f  3Gi        RWO          gp2           29s

[user@demo ~]$ oc rsh httpd
sh-4.4$ df -h
Filesystem      Size  Used Avail Use% Mounted on
...output omitted...
/dev/nvme2n1  2.9G  9.0M  2.9G   1% /var/www/html
...output omitted...
```

Testen der OpenShift-Anwendungserstellung und -bereitstellung

- Erstellen Sie eine Testanwendung, und stellen Sie sie bereit, um den Build-Zyklus der OpenShift-Anwendung zu überprüfen.

```
[user@demo ~]$ oc new-project validate
[user@demo ~]$ oc new-app django-psql-example
[user@demo ~]$ oc get pods -n validate
NAME                  READY   STATUS    RESTARTS   AGE
django-psql-example-1-build  0/1     Completed  0          11m
django-psql-example-1-deploy 0/1     Completed  0          10m
django-psql-example-1-vfb5l  1/1     Running   0          10m
postgresql-1-bdgkk         1/1     Running   0          11m
postgresql-1-deploy        0/1     Completed  0          11m
```

```
[user@demo ~]$ oc logs -f django-psql-example-1-build
...output omitted...
Successfully pushed image-registry.openshift-image-registry.svc:5000/validate/
django-psql-example@sha256:b97b...ff82
Push successful
```

```
[user@demo ~]$ oc logs -f django-psql-example-1-deploy
...output omitted...
--> Scaling django-psql-example-1 to 1
--> Success
```

```
[user@demo ~]$ oc get routes -n validate
NAME           HOST/PORT                               PATH
SERVICES      PORT  TERMINATION WILDCARD
django-psql-example  django-psql-example-validate.apps.ocp4-aws.example.com
django-psql-example <all>                         None
```

```
[user@demo ~]$ curl -Is \
> http://django-psql-example-validate.apps.ocp4-aws.example.com
...output omitted...
HTTP/1.1 200 OK
...output omitted...
```

```
[user@demo ~]$ firefox
http://django-psql-example-validate.apps.ocp4-aws.example.com
```

Überprüfen von OpenShift-Installationsanpassungen auf AWS

Nach der Durchführung eines grundlegenden Health Checks für den OpenShift-Cluster müssen Sie sicherstellen, dass die während der OpenShift-Installation angewendeten Anpassungen umgesetzt wurden.

Verteilung von OpenShift-Cluster-Knoten

- Überprüfen Sie, ob beim OpenShift-Installationsprozess die Cluster-Knoten auf unterschiedlichen VZs (Verfügbarkeitszonen) verteilt wurden.

```
[user@demo ~]$ oc get nodes --label-columns \
> failure-domain.beta.kubernetes.io/region,failure-domain.beta.kubernetes.io/zone
NAME           STATUS   ROLES    AGE     VERSION
ip-10-0-131-192.us-east-2.compute.internal   Ready    master   6d20h
v1.19.0+9f84db3 us-east-2   us-east-2a
ip-10-0-133-26.us-east-2.compute.internal   Ready    worker   6d20h
v1.19.0+9f84db3 us-east-2   us-east-2a
ip-10-0-161-55.us-east-2.compute.internal   Ready    master   6d20h
v1.19.0+9f84db3 us-east-2   us-east-2b
ip-10-0-168-121.us-east-2.compute.internal   Ready    worker   6d20h
v1.19.0+9f84db3 us-east-2   us-east-2b
ip-10-0-213-142.us-east-2.compute.internal   Ready    master   6d20h
v1.19.0+9f84db3 us-east-2   us-east-2c
ip-10-0-215-243.us-east-2.compute.internal   Ready    worker   6d20h
v1.19.0+9f84db3 us-east-2   us-east-2c
```

Instanzengrößen für OpenShift-Cluster-Knoten

- Verifizieren Sie die AWS-Instanzgrößen der Cluster-Knoten.

```
[user@demo ~]$ oc get nodes --label-columns \
> beta.kubernetes.io/instance-type
NAME           STATUS   ROLES    AGE     VERSION
INSTANCE-TYPE
ip-10-0-131-192.us-east-2.compute.internal   Ready    master   6d20h
v1.19.0+9f84db3   m5.2xlarge
ip-10-0-133-26.us-east-2.compute.internal   Ready    worker   6d20h
v1.19.0+9f84db3   m5.4xlarge
ip-10-0-161-55.us-east-2.compute.internal   Ready    master   6d20h
v1.19.0+9f84db3   m5.2xlarge
ip-10-0-168-121.us-east-2.compute.internal   Ready    worker   6d20h
v1.19.0+9f84db3   m5.4xlarge
ip-10-0-213-142.us-east-2.compute.internal   Ready    master   6d20h
v1.19.0+9f84db3   m5.2xlarge
ip-10-0-215-243.us-east-2.compute.internal   Ready    worker   6d20h
v1.19.0+9f84db3   m5.4xlarge
```

OpenShift-Cluster-Netzwerk

- Überprüfen Sie die Konfiguration des OpenShift-Cluster-Netzwerks.

```
[user@demo ~]$ oc get network.config/cluster -o yaml
apiVersion: config.openshift.io/v1
kind: Network
metadata:
...output omitted...
spec:
```

```

clusterNetwork:
  - cidr: 10.128.0.0/12
    hostPrefix: 23
  externalIP:
    policy: {}
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
status:
  clusterNetwork:
    - cidr: 10.128.0.0/12
      hostPrefix: 23
  clusterNetworkMTU: 8951
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16

```

Storage-Leistung von OpenShift-etcd

- Überprüfen Sie die Storage-Leistung des etcd-Clusters.

```
[user@demo ~]$ oc get pods -n openshift-etcd | grep etcd-ip
etcd-ip-10-0-151-122.us-east-2.compute.internal          3/3     Running   0
  4h52m
etcd-ip-10-0-176-145.us-east-2.compute.internal          3/3     Running
  0           4h54m
etcd-ip-10-0-203-117.us-east-2.compute.internal          3/3     Running
  0           4h55m
```

```
[user@demo ~]$ oc rsh -n openshift-etcd \
> etcd-ip-10-0-151-122.us-east-2.compute.internal
...output omitted...

sh-4.4# etcdctl check perf --load="m"
60 / 60 Booooooooooooooooooooooo! 100.00% 1m0s
PASS: Throughput is 993 writes/s
PASS: Slowest request took 0.192837s
PASS: Stddev is 0.012513s
PASS

sh-4.4# etcdctl check perf --load="l"
60 / 60 Booooooooooooooooooooooo! 100.00% 1m0s
FAIL: Throughput too low: 6506 writes/s
PASS: Slowest request took 0.211567s
PASS: Stddev is 0.021653s
FAIL
```

Die Testergebnisse zeigen, dass der etcd-Cluster für einen mittelgroßen Cluster (–load="m") gut funktioniert und bei einem großen Cluster (–load="l") fehlschlägt. Weitere

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

Informationen zur etcd-Leistung finden Sie auf der Dokumentationsseite zu etcd unter <https://etcd.io/docs/current/op-guide/hardware/>

Verwenden Sie das Tool **fio** aus dem Container **etcd-perf**, um einen Leistungstest auf den Control Plane-Knoten durchzuführen und so detailliertere Informationen zur etcd-Storage-Leistung zu erhalten. Die Ausgabe des Leistungstests gibt an, ob die Festplatte schnell genug ist, um etcd zu hosten. Dabei wird das 99. Perzentil der aus der Ausführung erfassten fsync-Metrik verglichen, um festzustellen, ob es kleiner als **10 ms** ist.

```
[user@demo ~]$ oc debug node/ip-10-0-151-122.us-east-2.compute.internal
...output omitted...
sh-4.4# chroot /host
sh-4.4# podman run --volume /var/lib/etcd:/var/lib/etcd:z
quay.io/openshift-scale/etcd-perf
{
    "fio version" : "fio-3.7",
...output omitted...
    "global options" : {
        "rw" : "write",
        "ioengine" : "sync",
        "fdatasync" : "1",
        "directory" : "/var/lib/etcd",
        "size" : "22m",
        "bs" : "2300"
    },
...output omitted...
    "write" : {
...output omitted...
        "iops" : 1273.362113,
...output omitted...
    }
}
-----
99th percentile of fsync is 970752 ns
99th percentile of the fsync is within the recommended threshold - 10 ms, the disk can be used to host etcd
```

Der **fio**-Leistungstest gibt das folgende Ergebnis aus:

1. Dieser Test schreibt 22 MiB Daten in Blöcken von 2.300 Byte in das Verzeichnis `/var/lib/etcd`.
2. Das 99. Perzentil von fsync ist 970.752 ns, was einer Schreiblatenz von **1ms** entspricht.
3. Das Betriebssystem hat während des Tests durchschnittlich **1273 IOPS** erreicht.

OpenShift Machine API

- Verifizieren Sie die Bereitstellung und automatische Skalierung des Server-Knotens.

Bei der Installation von OpenShift auf AWS mit der Full-Stack-Automatisierungsmethode erstellt und konfiguriert das OpenShift-Installationsprogramm eine MachineSet-Ressource für jede Verfügbarkeitszone in der ausgewählten Region.

```
[user@demo ~]$ oc get machines -n openshift-machine-api
NAME                           PHASE   TYPE      REGION   ZONE
AGE
ocp4-aws-9r678-master-0       Running  m5.2xlarge us-east-2 us-
east-2a  16h
ocp4-aws-9r678-master-1       Running  m5.2xlarge us-east-2 us-
east-2b  16h
ocp4-aws-9r678-master-2       Running  m5.2xlarge us-east-2 us-
east-2c  16h
ocp4-aws-9r678-worker-us-east-2a-gq2ps Running  m5.4xlarge us-east-2 us-
east-2a  16h
ocp4-aws-9r678-worker-us-east-2b-slp7l  Running  m5.4xlarge us-east-2 us-
east-2b  16h
ocp4-aws-9r678-worker-us-east-2c-vj7pj  Running  m5.4xlarge us-east-2 us-
east-2c  16h
```

```
[user@demo ~]$ oc get machinesets -n openshift-machine-api
NAME          DESIRED  CURRENT  READY  AVAILABLE  AGE
ocp4-aws-9r678-worker-us-east-2a  1         1        1        1        16h
ocp4-aws-9r678-worker-us-east-2b  1         1        1        1        16h
ocp4-aws-9r678-worker-us-east-2c  1         1        1        1        16h
```

```
[user@demo ~]$ oc get nodes --label-columns \
> failure-domain.beta.kubernetes.io/region,failure-domain.beta.kubernetes.io/zone
NAME          STATUS  ROLES   AGE    VERSION
REGION        ZONE
ip-10-0-151-177.us-east-2.compute.internal  Ready  master  16h
v1.19.0+9f84db3  us-east-2  us-east-2a
ip-10-0-157-4.us-east-2.compute.internal  Ready  worker  16h
v1.19.0+9f84db3  us-east-2  us-east-2a
```

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

ip-10-0-166-182.us-east-2.compute.internal v1.19.0+9f84db3 us-east-2 us-east-2b	Ready	worker	16h
ip-10-0-180-27.us-east-2.compute.internal v1.19.0+9f84db3 us-east-2 us-east-2b	Ready	master	17h
ip-10-0-205-233.us-east-2.compute.internal v1.19.0+9f84db3 us-east-2 us-east-2c	Ready	master	17h
ip-10-0-217-153.us-east-2.compute.internal v1.19.0+9f84db3 us-east-2 us-east-2c	Ready	worker	16h

Mithilfe des Workers MachineSets können Sie die Anzahl der auf dem Cluster ausgeführten Server-Knoten hoch- oder herunterskalieren.

```
[user@demo ~]$ oc scale machineset ocp4-aws-9r678-worker-us-east-2c \  
> --replicas=2 -n openshift-machine-api  
machineset.machine.openshift.io/ocp4-aws-9r678-worker-us-east-2c scaled
```

[user@demo ~]\$ oc get machinesets -n openshift-machine-api					
NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
ocp4-aws-9r678-worker-us-east-2a	1	1	1	1	17h
ocp4-aws-9r678-worker-us-east-2b	1	1	1	1	17h
ocp4-aws-9r678-worker-us-east-2c	2	2	1	1	17h

Nach ein paar Minuten muss der neue Server-Knoten den Status Ready aufweisen.

[user@demo ~]\$ oc get machinesets -n openshift-machine-api					
NAME	DESIRED	CURRENT	READY	AVAILABLE	AGE
ocp4-aws-9r678-worker-us-east-2a	1	1	1	1	17h
ocp4-aws-9r678-worker-us-east-2b	1	1	1	1	17h
ocp4-aws-9r678-worker-us-east-2c	2	2	2	2	17h

[user@demo ~]\$ oc get machines -n openshift-machine-api					
NAME	PHASE	TYPE	REGION	ZONE	AGE
ocp4-aws-9r678-master-0	Running	m5.2xlarge	us-east-2	us-east-2a	17h
ocp4-aws-9r678-master-1	Running	m5.2xlarge	us-east-2	us-east-2b	17h
ocp4-aws-9r678-master-2	Running	m5.2xlarge	us-east-2	us-east-2c	17h

Kapitel 2 | Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

```
ocp4-aws-9r678-worker-us-east-2a-gq2ps    Running   m5.4xlarge   us-east-2   us-
east-2a  17h
ocp4-aws-9r678-worker-us-east-2b-slp7l    Running   m5.4xlarge   us-east-2   us-
east-2b  17h
ocp4-aws-9r678-worker-us-east-2c-65hln  Running   m5.4xlarge   us-east-2
us-east-2c  3m41s
ocp4-aws-9r678-worker-us-east-2c-vj7pj    Running   m5.4xlarge   us-east-2   us-
east-2c  17h
```

```
[user@demo ~]$ oc get nodes --label-columns \
> failure-domain.beta.kubernetes.io/region,failure-domain.beta.kubernetes.io/zone
NAME                               STATUS  ROLES   AGE     VERSION
REGION      ZONE
ip-10-0-151-177.us-east-2.compute.internal  Ready   master  17h
v1.19.0+9f84db3  us-east-2  us-east-2a
ip-10-0-157-4.us-east-2.compute.internal  Ready   worker  17h
v1.19.0+9f84db3  us-east-2  us-east-2a
ip-10-0-166-182.us-east-2.compute.internal  Ready   worker  17h
v1.19.0+9f84db3  us-east-2  us-east-2b
ip-10-0-180-27.us-east-2.compute.internal  Ready   master  17h
v1.19.0+9f84db3  us-east-2  us-east-2b
ip-10-0-196-32.us-east-2.compute.internal  Ready   worker  2m31s
v1.19.0+9f84db3  us-east-2  us-east-2c
ip-10-0-205-233.us-east-2.compute.internal  Ready   master  17h
v1.19.0+9f84db3  us-east-2  us-east-2c
ip-10-0-217-153.us-east-2.compute.internal  Ready   worker  17h
v1.19.0+9f84db3  us-east-2  us-east-2c
```

Mit der OpenShift Machine API wurde automatisch ein neuer Server-Knoten bereitgestellt (`ip-10-0-196-32.us-east-2.compute.internal`) und dem Cluster in der gewünschten AWS-VZ `us-east-2c` hinzugefügt.

Videoergebnis:

Überprüfen, ob das OpenShift-Cluster für die Day 2-Aufgaben zum Onboarding von Benutzern und Anwendungen bereit ist



Literaturhinweise

- Weitere Informationen finden Sie im Leitfaden *Installing on AWS* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_aws

► Quiz

Überprüfen der Installation von OpenShift auf AWS

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Welche drei der folgenden Health Checks müssen Administratoren nach einer OpenShift-Installation auf AWS durchführen? (Wählen Sie drei Antworten aus.)
 - a. Überprüfen, ob sich alle Cluster-Knoten im Status Ready befinden.
 - b. Überprüfen, ob der Bootstrap-Knoten ausgeführt wird.
 - c. Sicherstellen, dass der Cluster im CVO-Bericht (Cluster Version Operator) als verfügbar und bereit aufgeführt wird.
 - d. Überprüfen, ob alle Cluster-Operatoren verfügbar und bereit sind.
 - e. Überprüfen, ob der Cluster mithilfe eines sekundären IAM-Benutzers installiert wurde.

- ▶ 2. Welche der folgenden Befehle müssen bei der Verifizierung einer OpenShift-Installation verwendet werden, um die Systemuhrsynchronisierung der Cluster-Knoten zu überprüfen?
 - a. ntpstat
 - b. chronyc tracking
 - c. ntpq -p

- ▶ 3. Red Hat empfiehlt die Verwendung eines Debug-Pods zum Generieren einer sosreport-Datei aus einem OpenShift-Cluster-Knoten. (Richtig oder falsch)
 - a. Richtig
 - b. Falsch

- ▶ 4. Welche der folgenden Befehle müssen bei der Messung der etcd-Storage-Leistung verwendet werden?
 - a. etcdctl endpoint health
 - b. etcdctl check perf
 - c. etcdctl perf etcd

- ▶ 5. Welche der folgenden Befehle müssen nach der Installation von OpenShift auf AWS verwendet werden, um die OpenShift Machine API-Integration in die AWS-Services zu überprüfen?
 - a. oc scale machines
 - b. oc edit autoscaling -o yaml
 - c. oc scale machineset

► Lösung

Überprüfen der Installation von OpenShift auf AWS

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Welche drei der folgenden Health Checks müssen Administratoren nach einer OpenShift-Installation auf AWS durchführen? (Wählen Sie drei Antworten aus.)
 - a. Überprüfen, ob sich alle Cluster-Knoten im Status Ready befinden.
 - b. Überprüfen, ob der Bootstrap-Knoten ausgeführt wird.
 - c. Sicherstellen, dass der Cluster im CVO-Bericht (Cluster Version Operator) als verfügbar und bereit aufgeführt wird.
 - d. Überprüfen, ob alle Cluster-Operatoren verfügbar und bereit sind.
 - e. Überprüfen, ob der Cluster mithilfe eines sekundären IAM-Benutzers installiert wurde.
- ▶ 2. Welche der folgenden Befehle müssen bei der Verifizierung einer OpenShift-Installation verwendet werden, um die Systemuhrsynchronisierung der Cluster-Knoten zu überprüfen?
 - a. ntpstat
 - b. chronyc tracking
 - c. ntpq -p
- ▶ 3. Red Hat empfiehlt die Verwendung eines Debug-Pods zum Generieren einer sosreport-Datei aus einem OpenShift-Cluster-Knoten. (Richtig oder falsch)
 - a. Richtig
 - b. Falsch
- ▶ 4. Welche der folgenden Befehle müssen bei der Messung der etcd-Storage-Leistung verwendet werden?
 - a. etcdctl endpoint health
 - b. etcdctl check perf
 - c. etcdctl perf etcd
- ▶ 5. Welche der folgenden Befehle müssen nach der Installation von OpenShift auf AWS verwendet werden, um die OpenShift Machine API-Integration in die AWS-Services zu überprüfen?
 - a. oc scale machines
 - b. oc edit autoscaling -o yaml
 - c. oc scale machineset

► Quiz

Kapitelwiederholung: Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche drei der folgenden Vorteile ergeben sich bei der Installation von OpenShift bei einem unterstützten Cloud-Anbieter durch die Installation mit Full-Stack-Automatisierung? (Wählen Sie drei Antworten aus.)
- a. Administratoren installieren OpenShift mit minimalem manuellen Eingriff.
 - b. Die Installation mit Full-Stack-Automatisierung kann besser angepasst werden als die Installationsmethode in einer bereits vorhandenen Infrastruktur.
 - c. Das OpenShift-Installationsprogramm erstellt automatisch die erforderlichen Cloud-Ressourcen.
 - d. Das OpenShift-Installationsprogramm konfiguriert automatisch die OpenShift Machine API für die automatische Skalierung von Cluster-Knoten.
- 2. Während des gesamten OpenShift-Installationsvorgangs muss der Bootstrap-Knoten ausgeführt werden. (Richtig oder falsch)
- a. Richtig
 - b. Falsch
- 3. Bei der Installation von OpenShift auf MS Azure mit der Full-Stack-Automatisierungsmethode sind welche zwei der folgenden Aspekte Voraussetzungen? (Wählen Sie zwei Antworten aus.)
- a. Sicherstellen, dass das IAM Azure-Benutzerkonto die zum Installieren von OpenShift erforderlichen Kontingent- und Berechtigungsvoraussetzungen erfüllt.
 - b. Bereitstellen des Bastion-Hosts in der MS Azure-Cloud
 - c. Konfigurieren eines HTTPD-Servers zum Speichern der Ignition-Konfigurationsdateien.
 - d. Überprüfen, welche MS Azure-Regionen für die Installation von OpenShift unterstützt werden.
- 4. Welche der folgenden AWS EBS-Volume-Typen bieten die höhere maximale Anzahl an IOPS pro Volume?
- a. gp2
 - b. gp3
 - c. io1
 - d. io2

► **5. Welche drei der folgenden Aktionen sind automatische Aktionen, die vom OpenShift-Cluster ausgeführt werden, wenn Sie ein Worker-machineset in einem auf AWS bereitgestellten OpenShift-Cluster hochskalieren? (Wählen Sie drei Antworten aus.)**

- a. Die OpenShift Machine API stellt automatisch eine AWS EC2-Instanz für den neuen Server-Knoten bereit und startet sie.
- b. Der neue Server-Knoten ruft seine Ignition-Konfigurationsdatei ab und installiert RHCOS.
- c. Der neue Server-Knoten verbindet sich mit dem OpenShift-Cluster.
- d. Der neue Server-Knoten ist als Infrastruktur-Server-Koten konfiguriert.
- e. Der neue Server-Knoten ist heruntergefahren.

► Lösung

Kapitelwiederholung: Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche drei der folgenden Vorteile ergeben sich bei der Installation von OpenShift bei einem unterstützten Cloud-Anbieter durch die Installation mit Full-Stack-Automatisierung? (Wählen Sie drei Antworten aus.)
- a. Administratoren installieren OpenShift mit minimalem manuellen Eingriff.
 - b. Die Installation mit Full-Stack-Automatisierung kann besser angepasst werden als die Installationsmethode in einer bereits vorhandenen Infrastruktur.
 - c. Das OpenShift-Installationsprogramm erstellt automatisch die erforderlichen Cloud-Ressourcen.
 - d. Das OpenShift-Installationsprogramm konfiguriert automatisch die OpenShift Machine API für die automatische Skalierung von Cluster-Knoten.
- 2. Während des gesamten OpenShift-Installationsvorgangs muss der Bootstrap-Knoten ausgeführt werden. (Richtig oder falsch)
- a. Richtig
 - b. Falsch
- 3. Bei der Installation von OpenShift auf MS Azure mit der Full-Stack-Automatisierungsmethode sind welche zwei der folgenden Aspekte Voraussetzungen? (Wählen Sie zwei Antworten aus.)
- a. Sicherstellen, dass das IAM Azure-Benutzerkonto die zum Installieren von OpenShift erforderlichen Kontingent- und Berechtigungsvoraussetzungen erfüllt.
 - b. Bereitstellen des Bastion-Hosts in der MS Azure-Cloud
 - c. Konfigurieren eines HTTPD-Servers zum Speichern der Ignition-Konfigurationsdateien.
 - d. Überprüfen, welche MS Azure-Regionen für die Installation von OpenShift unterstützt werden.
- 4. Welche der folgenden AWS EBS-Volume-Typen bieten die höhere maximale Anzahl an IOPS pro Volume?
- a. gp2
 - b. gp3
 - c. io1
 - d. io2

► **5. Welche drei der folgenden Aktionen sind automatische Aktionen, die vom OpenShift-Cluster ausgeführt werden, wenn Sie ein Worker-machineset in einem auf AWS bereitgestellten OpenShift-Cluster hochskalieren? (Wählen Sie drei Antworten aus.)**

- a. Die OpenShift Machine API stellt automatisch eine AWS EC2-Instanz für den neuen Server-Knoten bereit und startet sie.
- b. Der neue Server-Knoten ruft seine Ignition-Konfigurationsdatei ab und installiert RHCOS.
- c. Der neue Server-Knoten verbindet sich mit dem OpenShift-Cluster.
- d. Der neue Server-Knoten ist als Infrastruktur-Server-Koten konfiguriert.
- e. Der neue Server-Knoten ist heruntergefahren.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Die Vorteile einer Installation mit Full-Stack-Automatisierung bei der Installation von OpenShift auf einer unterstützten IaaS-Cloud-Anbieter-Plattform.
- Die Nutzung der Full-Stack-Automatisierung bei der Installation von OpenShift auf einer unterstützten IaaS-Cloud-Provider-Plattform.
- Die Installation von OpenShift mit Full-Stack-Automatisierung auf Microsoft Azure.
- Die Installation von OpenShift mit Full-Stack-Automatisierung auf Red Hat OpenStack Platform
- Die ausführlichen Schritte bei der Installation von OpenShift mit Full-Stack-Automatisierung auf Amazon Web Services:
 - Vorbereiten der Voraussetzungen für die Installation von OpenShift.
 - Größe der OpenShift-Installation festlegen.
 - Anpassen der OpenShift-Installation.
 - Ausführen der OpenShift-Installation.
 - Überwachen der OpenShift-Installation.
 - Beheben von Fehlern bei der OpenShift-Installation.
 - Verifizieren der OpenShift-Installation durch Ausführung eines detaillierten OpenShift-Health Checks.

Kapitel 3

Installieren von OpenShift in einer virtualisierten Umgebung

Ziel

Bereitstellen von OpenShift-Clustern auf Hypervisors mit allgemeinen Anpassungen unter Verwendung der Installationsmethoden „Full-Stack-Automatisierung“ und „Installation auf bereits vorhandener Infrastruktur“

Ziele

- Beschreiben der Architektur und des Workflows zur Installation von OpenShift auf Hypervisors mit der Full-Stack-Automatisierungsmethode und der Methode bei bereits vorhandener Infrastruktur.
- Beschreiben der Installation von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierung mit allgemeinen Anpassungen.
- Beschreiben der Installation von OpenShift auf vSphere mithilfe der Installation bei bereits vorhandener Infrastruktur mit allgemeinen Anpassungen.

Abschnitte

- Einführung in OpenShift-Installationsmethoden auf Hypervisors (und Test)
- Beschreiben der Installation von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierungsmethode (und Test)
- Beschreiben der Installation von OpenShift auf vSphere bei bereits vorhandener Infrastruktur (und Test)
- Kapitelwiederholung: Installieren von OpenShift in einer virtualisierten Umgebung (Test)

Einführung in OpenShift-Installationsmethoden auf Hypervisors

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die Architektur und den Workflow zur Installation von OpenShift auf Hypervisors mit der Full-Stack-Automatisierungsmethode und der Methode bei bereits vorhandener Infrastruktur beschreiben können.

Einführung in OpenShift-Installationsmethoden auf Hypervisors

Vor der Einführung von Hypervisors nutzten physische Computer nur ein Betriebssystem. Obwohl dies einen stabilen physischen Server bot, der Anforderungen von einem einzelnen Betriebssystem effektiv und effizient verarbeiten konnte, wurden nicht verwendete Ressourcen verschwendet. Ein Hypervisor ist ein spezieller Software-Layer, der die parallele Ausführung mehrerer Betriebssysteme ermöglicht.

Ein Hypervisor ist eine spezifische Softwareschicht, die virtuelle Rechner erstellt und ausgeführt. Ein Hypervisor wird manchmal auch als "Monitor" für eine virtuelle Maschine (VM) bezeichnet, da er über Funktionen zur Isolierung des Betriebssystems verfügt.

VM-Ressourcen sind logisch voneinander isoliert. Ressourcen wie Arbeitsspeicher und CPU werden vom Hypervisor basierend auf den Anforderungen der VM und dem Pool der auf dem physischen Host verfügbaren Ressourcen den vorhandenen und neuen virtuellen Rechnern zugewiesen und geplant.

Der wichtigste Vorteil der Virtualisierung besteht in der möglichen Nutzung mehrerer Betriebssysteme, die die gleichen, von einem Hypervisor verwalteten virtualisierten Hardwareressourcen nutzen. Für eine erfolgreiche Bereitstellung von OpenShift Container Platform sind neben dem Hypervisor auch weitere Services erforderlich, z. B. ein Load Balancer für den Datenverkehr oder die DNS-Zonenverwaltung.

Einführung in Hypervisors

Virtualisierungs-Hypervisors werden entweder als Bare-Metal-Hypervisor oder als Hosted-Hypervisor kategorisiert.

Ein nativer Hypervisor wird allgemein als ein „Bare-Metal-Hypervisor“ definiert, da er das Host-Betriebssystem ersetzen, direkt auf der Hardware des Hosts ausgeführt werden, die Gastbetriebssysteme verwalten und VM-Ressourcen direkt in der Hardware planen kann. Die folgenden Bare-Metal-Hypervisors werden häufig in Rechenzentren von Unternehmen und in serverbasierten Umgebungen bereitgestellt.

- RHV
- Microsoft Hyper-V
- vSphere ESXi

Ein Software-Layer-Hypervisor wird als Hosted-Hypervisor bezeichnet, da er auf einem Standard-Betriebssystem als Software-Layer oder Anwendung ausgeführt werden kann.

Kapitel 3 | Installieren von OpenShift in einer virtualisierten Umgebung

Im Gegensatz zu Bare-Metal-Hypervisors planen Software-Layer-Hypervisors VM-Ressourcen abhängig vom Host-Betriebssystem und werden auf der Hardware ausgeführt. Sie sind in der Regel für PC-Benutzer geeignet, die Zugriff auf mehrere Betriebssysteme benötigen. Im Folgenden finden Sie Beispiele für Software-Layer-Hypervisors:

- VMware Workstation
- Oracle VirtualBox

Red Hat unterstützt keine OpenShift-Cluster, die sowohl Bare-Metal- als auch Software-Layer-Hypervisors nutzen. Ebenso unterstützen wir nicht mehrere Cloud-Anbieter für dieselben Cluster. Ein OpenShift-Cluster definiert Knoten anderer Cloud-Anbieter als Eindringlinge und ignoriert sie. Wenn ein Cloud-Anbieter aktiviert ist, ist der Cluster an diesen bestimmten Cloud-Anbieter gebunden.

Gründe für die Installation von OpenShift auf Hypervisors mit Full-Stack-Automatisierung

Für unterstützte Hypervisors empfiehlt Red Hat aus den folgenden Gründen die Verwendung der Installationsmethode „Full-Stack-Automatisierung“:

- Cluster-Administratoren installieren OpenShift mit minimalem manuellen Eingriff mit einem "eigenwilligen" Installations-Assistenten.
- Das OpenShift-Installationsprogramm erstellt die virtuellen Rechner, installiert RHCOS und startet dann den OpenShift-Installationsprozess.
- Das OpenShift-Installationsprogramm verwendet die bereitgestellten Anmeldeinformationen, um sich mit dem vCenter zu verbinden, ein Template hochzuladen und dann dieses Template zu klonen, das für die Bootstrap- und Control-Plane-Knoten verwendet werden soll.
- Das OpenShift-Installationsprogramm integriert die OpenShift Machine API-Ressource vollständig in die Cloud-Anbieter-Services. Das Installationsprogramm erstellt die OpenShift-Ressource „MachineSets“, sodass die automatische Knotenbereitstellung und die automatische Cluster-Skalierung direkt nach Abschluss der Installation verwendet werden können.
- Die einzige DNS-Anforderung sind zwei Einträge: Ingress und API.

Gründe für die Installation von OpenShift auf Hypervisors unter Verwendung einer bereits vorhandenen Infrastruktur

Die Bereitstellung von OpenShift in einer bereits bestehenden Infrastruktur kann aus den folgenden Gründen vorteilhaft sein:

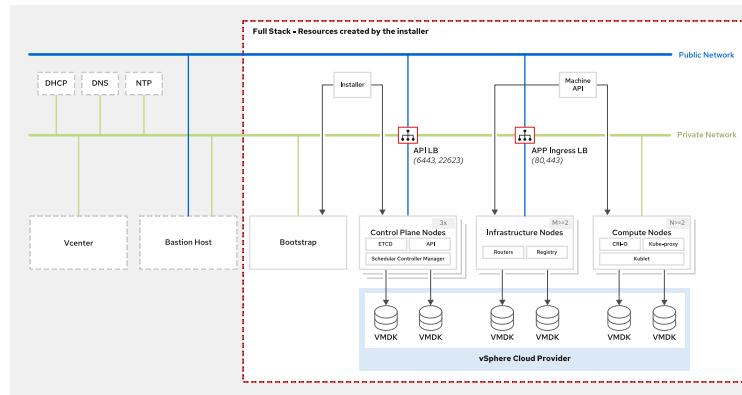
- Der Cluster-Administrator verwaltet die Infrastruktur und ermöglicht dadurch detaillierte Anpassungen.
- OpenShift-Cluster können in einer Infrastruktur bereitgestellt werden, die speziell auf ihre Anforderungen zugeschnitten ist.
- Cluster-Administratoren sind flexibler und können einfacher auf Ressourcenanforderungen reagieren.

Vergleichen der Installationsmethoden „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“

In der folgenden Tabelle werden die allgemeinen Unterschiede bei der Bereitstellung und den Anforderungen zwischen einer Full-Stack-Automation- und einer bereits vorhandenen Infrastruktur dargestellt..

Vergleich zwischen Full-Stack-Automatisierung und bereits vorhandener Infrastruktur

Aktion	Full-Stack-Automatisierung	Bereits vorhandene Infrastruktur
Netzwerk aufbauen	Installationsprogramm	Benutzer
Load Balancer einrichten	Installationsprogramm	Benutzer
DNS konfigurieren	Installationsprogramm	Benutzer
Hardware- oder VM-Bereitstellung	Installationsprogramm	Benutzer
Betriebssysteminstallation	Installationsprogramm	Benutzer
Ignition-Konfigurationen generieren	Installationsprogramm	Installationsprogramm
Betriebssystemunterstützung	Installationsprogramm: RHCOS	Benutzer: RHCOS
Persistenten Storage für die interne Registry konfigurieren	Installationsprogramm	Benutzer
Dynamischen Storage-Anbieter konfigurieren	Installationsprogramm	Benutzer
Knotenbereitstellung und automatische Skalierung konfigurieren	Ja	Nur für Anbieter mit OpenShift Machine API-Unterstützung

Beschreiben der allgemeinen Architektur eines Full-Stack-Clusters in einem Hypervisor.

Eine Full Stack-Cluster-Architektur besteht aus Folgendem:

Bastion-Host (optional)

Der Bastion-Host benötigt eine Netzwerkverbindung zur Cloud-Anbieter-API, um OpenShift zu installieren, und zur OpenShift-API, um OpenShift nach der Installation zu verwalten.

Control Plane-Knoten

Verwaltet Workloads für die Server-Knoten und führt die zum Steuern des Clusters erforderlichen Services aus.

Server-Knoten

Speicherort, an dem die von Kubernetes-Nutzern tatsächlich angeforderten Workloads ausgeführt und verwaltet werden.

Infrastrukturknoten

Ausführung zentraler Infrastrukturkomponenten wie Service Broker und Protokollierung

Temporärer Bootstrap-Knoten

Ein temporärer Knoten, auf dem eine minimale Kubernetes-Bereitstellung ausgeführt wird, die zum Bereitstellen der OpenShift-Control Plane verwendet wird. Er wird am Ende der Installation gelöscht.

Ingress-Load Balancer

Die virtuelle IP (VIP) wird von Keepalive verwaltet und nur auf Knoten gehostet, die über eine Routerinstanz verfügen. Datenverkehr, der für die Ingress-VIP „*.apps“ bestimmt ist, wird direkt an die Router-Instanz weitergegeben.

API-Load Balancer

Wenn ein Client eine neue Anforderung für die API erstellt, sorgt HAProxy auf dem Knoten, auf dem die API-IP ausgeführt wird, mittels Round Robin für die Lastverteilung zwischen den Knoten.

Beschreiben von OpenShift-Netzwerk-Plug-ins für vSphere

Für die Layer-2-VM-Konnektivität steht OpenShift Software Defined Networking (SDN) zur Verfügung. Es gibt jedoch vSphere-Plug-ins, die die von OpenShift SDN bereitgestellten Funktionen integrieren oder ersetzen können.

Networking (NSX-T)

Networking (NSX-T) ist das in vSphere integrierte Software Defined Network (SDN). NSX-T erstellt Overlay-Netzwerke für die VM-Konnektivität mit zusätzlichen Features, z. B. Mikrosegmentierung, Load Balancer und granulare Sicherheitsrichtlinien. Mit NSX-T kann beliebiger Datenverkehr zur oder von der VM auf dem Netzwerk-Layer durch eine Firewall geleitet werden.

NSX Container-Plug-in (NCP)

Das NSX Container-Plug-in integriert NSX-T und OpenShift. Er ersetzt OpenShift SDN und erstellt Load Balancer-Objekte im Cluster. Mit dem NSX-Manager wird sichtbar, welche Pods mit welchen Netzwerken verbunden sind. NCP wird während des Installationsprozesses über zuvor erstellte Manifeste bereitgestellt.

Beschreiben von Cluster-Storage

VMware hat den vSphere Cloud Provider entwickelt, um die persistenten Storage-Anforderungen von Containern zu unterstützen. Der vSphere Cloud Provider ist ein Storage-Anbieter, der Volume-Plug-ins bietet, auf die die OpenShift-Plattform zugreifen kann, die von VMware vSAN oder einem unterstützten vSphere Datastore unterstützt werden. VMFS (Virtual Machine File System), NFS (Network File System) oder vSAN-Datenspeicher (Virtual Storage Area Network) sind verfügbare Storage-Angebote.

Persistente Volumes (PV)

Persistente Volumes (PVs) ermöglichen einem Cluster-Administrator, persistenten Storage für einen Cluster bereitzustellen. PVs sind Ressourcen, die keinen Geltungsbereich für ein

bestimmtes Projekt und einen Lebenszyklus haben, der unabhängig von den einzelnen Pods ist, die das PV verwenden. Nachdem ein PV an eine PVC gebunden ist, kann dieses PV nicht mehr mit zusätzlichen PVCs verbunden werden. In der folgenden Tabelle werden die für ein Ziel-Volume-Plug-in verfügbaren Storage-Zugriffsmodi gezeigt.

Storage-Zugriffsmodi

Volume-Plug-in	ReadWriteOnce	ReadOnlyMany	ReadWriteMany
VMware vSphere	X		
Cinder	X		
NFS	X	X	X

Die Zugänglichkeit von Storage-Komponenten und -Funktionen kann je nach Virtualisierungslösung variieren.

Virtual Storage Area Network (vSAN)

Ein virtuelles Storage-Netzwerk oder „VSAN“ ist eine hyperkonvergente, softwaredefinierte Storage-Lösung. Lokale Ressourcen auf den physischen Servern werden in **Pools** zusammengefasst oder kombiniert. Hybrid- und All-Flash-Versionen enthalten eine Flash-basierte Cache-Ebene zum Speichern und Abrufen des initialen Datensatzes. Jeder Knoten enthält eine Gruppe von Disks, Festplatten oder Solid-State-Drives (SSDs), die die Kapazitätsebene definieren.

Ein Flash-basiertes Cache-Device wie SATA oder SSD wird auf den Knoten **gepoolt** und bildet den vSAN-Datenspeicher. Ein Versuch, in den VSAN-Datenspeicher zu schreiben, durchläuft zuerst den Cache und wird dann in die Kapazitätsebene entleert. Die Daten werden während des Einlesens in die Cache-Ebene zwischengespeichert. Wenn beispielsweise wiederholt auf denselben Satz an Blöcken für die VM zugegriffen wird, können Sie daraus schließen, dass sie sich im Cache befinden.

VSAN Version 7 und höher unterstützt Datei- und Block-Storage. Dadurch kann der Cluster-Administrator RWO- und RWX-PVCs aus dem VSAN erstellen.

vSphere ESXi-Plug-ins

VSphere-Storage for Kubernetes, auch als „vSphere Cloud Provider“ bezeichnet, ist ein Cloud-Provisioner. Eine OpenShift-Bereitstellung auf vSphere ermöglicht standardmäßig **In-tree Storage**.

Die In-Tree-Storage-Integration wird von Red Hat in vSphere 6.5 und höher unterstützt. OpenShift Container Platform ermöglicht die Verwendung von VMware vSphere Virtual Machine Disk-Volumes (VMDK). VSAN und VMFS werden als Volume-Typen unterstützt, und VMware vSphere-Volumes können dynamisch bereitgestellt werden.

Die vSphere-Disk wird von OpenShift erstellt und an das richtige Image angehängt. Der Bereitstellungs-Storage ist nur ein Block-Storage und unterstützt **ReadWriteOnce**. Der Cluster-Administrator kann jeden Datenspeicher angeben, ob Block, Datei oder VSAN, und der Provisioner erstellt die VMDK (Virtual Machine Disk) für diesen Datenspeicher.

- Storage-In-Tree-Treiber
 - Früher von VMware verwaltet, ist dies der Standard-Storage, der bei der Verwendung von Full-Stack- und bereits vorhandenen Infrastrukturinstallationsmethoden bereitgestellt wird. Das Plug-in erstellt mithilfe der in OpenShift Container Platform enthaltenen In-Tree-Storage-Treiber für vSphere den vSphere-Storage und wird verwendet, wenn

vSphere-CSI-Treiber nicht verfügbar sind. Diese Treiber werden von Red Hat in vSphere Version 6.5 und höher unterstützt.

- vSphere-CSI-Treiber
 - Der Container Storage Integration-Provisioner (CSI) wird nicht von Red Hat bereitgestellt. Es handelt sich um einen vom Kunden bereitgestellten „Day 2“-Vorgang. Das Plug-in erstellt vSphere-Storage über die standardmäßige Container-Storage-Schnittstelle. Der vSphere-CSI-Treiber wird von VMware bereitgestellt und unterstützt.



Anmerkung

RHEL NFS unterstützt nicht die gesamte POSIX-Locking- und Caching-Semantik. Beim Testen wurden Probleme mit bestimmten Apps erkannt, die Storage gemeinsam verwenden, wie die interne Registry und HA-Datenbanken. Daher wird die Verwendung von RHEL NFS zum Sichern von PVs, die von zentralen Services verwendet werden, nicht empfohlen. Andere NFS-Implementierungen auf dem Markt haben diese Probleme jedoch möglicherweise nicht.

Wenden Sie sich an den NFS-Implementierungsanbieter, um mehr über abgeschlossene Tests für die Kernkomponenten von OpenShift Container Platform zu erfahren.



Literaturhinweise

- Weitere Informationen finden Sie im Kapitel *Installation and update* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/architecture

► Quiz

Einführung in OpenShift- Installationsmethoden auf Hypervisors

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

► 1. Welcher der folgenden Punkte beschreibt einen Hypervisor am besten?

- a. Ein einsatzbereites Container-Paket mit allem, was zur Ausführung einer Anwendung benötigt wird
- b. Eine Open-Source-Anwendungsschicht (Application Layer), auf der Images gespeichert werden
- c. Eine spezifische Software-Schicht (Software Layer), die virtuelle Rechner erstellt und ausführt

► 2. Welche Aussage beschreibt einen Grund für die Bereitstellung von OpenShift mit der Full-Stack-Installation auf einem Hypervisor?

- a. Der Cluster-Administrator verwaltet die Infrastruktur.
- b. Der Cluster-Administrator kann die Umgebung an seine Anforderungen anpassen.
- c. Der Cluster-Administrator erstellt den initialen Load Balancer.
- d. Der Cluster-Administrator installiert OpenShift mit minimalem manuellen Eingriff mit einer „eigenwilligen“ Methode.

► 3. Welche zwei der folgenden Hypervisors sind Software-Layer-Hypervisors? (Wählen Sie zwei Antworten aus.)

- a. VMware Workstation
- b. RHV
- c. Oracle VirtualBox
- d. vSphere ESXi

► 4. Welche zwei der folgenden DNS-Einträge sind bei der Durchführung einer Full-Stack-Installation von OpenShift auf Hypervisors erforderlich? (Wählen Sie zwei Antworten aus.)

- a. API-VIP
- b. ETCD VIP
- c. Ingress VIP
- d. SSL VIP

► Lösung

Einführung in OpenShift-Installationsmethoden auf Hypervisors

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

► 1. Welcher der folgenden Punkte beschreibt einen Hypervisor am besten?

- a. Ein einsatzbereites Container-Paket mit allem, was zur Ausführung einer Anwendung benötigt wird
- b. Eine Open-Source-Anwendungsschicht (Application Layer), auf der Images gespeichert werden
- c. Eine spezifische Software-Schicht (Software Layer), die virtuelle Rechner erstellt und ausführt

► 2. Welche Aussage beschreibt einen Grund für die Bereitstellung von OpenShift mit der Full-Stack-Installation auf einem Hypervisor?

- a. Der Cluster-Administrator verwaltet die Infrastruktur.
- b. Der Cluster-Administrator kann die Umgebung an seine Anforderungen anpassen.
- c. Der Cluster-Administrator erstellt den initialen Load Balancer.
- d. Der Cluster-Administrator installiert OpenShift mit minimalem manuellen Eingriff mit einer „eigenwilligen“ Methode.

► 3. Welche zwei der folgenden Hypervisors sind Software-Layer-Hypervisors? (Wählen Sie zwei Antworten aus.)

- a. VMware Workstation
- b. RHV
- c. Oracle VirtualBox
- d. vSphere ESXi

► 4. Welche zwei der folgenden DNS-Einträge sind bei der Durchführung einer Full-Stack-Installation von OpenShift auf Hypervisors erforderlich? (Wählen Sie zwei Antworten aus.)

- a. API-VIP
- b. ETCD VIP
- c. Ingress VIP
- d. SSL VIP

Beschreiben der Installation von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierungsmethode

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die Installation von OpenShift auf vSphere unter Verwendung von Full-Stack-Automatisierung mit allgemeinen Anpassungen beschreiben können.

Installieren von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierung mit allgemeinen Anpassungen.

Der Cluster-Administrator muss vor der Bereitstellung des Clusters sicherstellen, dass die spezifischen Voraussetzungen für vSphere erfüllt sind.

Das OpenShift-Installationsprogramm fordert den Cluster-Administrator zur Eingabe von Werten auf und verwendet dann die Eingabe dieser Werte, um die Datei „install-config.yaml“ zu konfigurieren. Die standardmäßig automatisierte „install-config.yaml“ ist möglicherweise nicht vollständig für Ihre Umgebung geeignet. Der Cluster-Administrator kann jedoch die Datei „install-config.yaml“ anpassen. Vor der Bereitstellung des Clusters sind Parameter für die Anpassung erforderlich. Schließlich erstellt das OpenShift-Installationsprogramm die erforderlichen vSphere-Ressourcen und installiert einen OpenShift-Cluster.

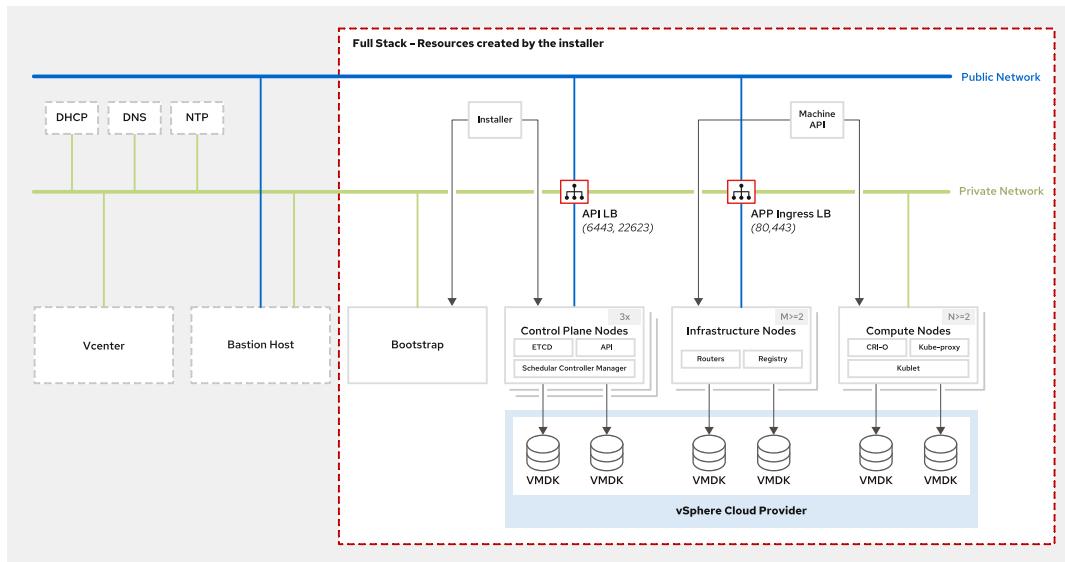
Überprüfen der Voraussetzungen für die Full-Stack-Installation von OpenShift auf vSphere

Vor der Bereitstellung eines OpenShift-Full-Stack-Clusters auf vSphere müssen Cluster-Administratoren die folgenden Voraussetzungen überprüfen:

- Persistenter Storage mit dem ReadWriteMany-Zugriffsmodus für die interne OpenShift-Registry als Day 2-Aufgabe wird verwendet.
- Verifizieren, dass der vSphere-Server nur ein Rechenzentrum und einen Cluster hat. Wenn mehr als ein Ressourcenpool vorhanden ist, werden Worker-Knoten während der Installation nicht bereitgestellt.
- Überprüfen der in der Lektion Beschreiben der OpenShift-Installationsvoraussetzungen beschriebenen allgemeinen Voraussetzungen und Voraussetzungen für die Full-Stack-Automatisierung.
- VMware vSphere Version 6 oder 7 wird zur Bereitstellung eines OpenShift Container Platform-Clusters verwendet.
- Konfigurieren von vCenter für die Infrastruktur vor der Bereitstellung eines OpenShift-Clusters.

Beschreiben der OpenShift-Full-Stack-Architektur auf vSphere

Das folgende Diagramm zeigt die Architektur einer Full-Stack-Bereitstellung in vSphere:



Diese Workflow-Beschreibung setzt voraus, dass Sie den vSphere ESXi-Host konfiguriert haben.

- Verifizieren Sie, dass die beiden DNS-Einträge, API und .apps, gültig sind.
- Verwenden Sie den binären interaktiven Modus des OpenShift-Installationsprogramms, um die Zielwerte abzurufen, die das Installationsprogramm für die Konfiguration der Datei „install-config.yaml“ benötigt.
 - Stellen Sie nach dem Aufruf des Befehls `openshift-install create cluster` den SSH-Schlüssel bereit, wenn das interaktive Installationsprogramm Sie zur Eingabe des Schlüssels auffordert. Der SSH-Schlüssel ist für die Authentifizierung von Knoten während der Installation unabdingbar.
 - Stellen Sie die von der vSphere-Zielplattform benötigten Informationen bereit, einschließlich vCenter-URL, Benutzername und Passwort.
 - Stellen Sie Werte für die Basis-Domain, den Cluster-Namen und das Pull Secret bereit, wenn Sie vom Installationsprogramm dazu aufgefordert werden. Wenn Sie keine Antworten auf die interaktiven Abfragen angeben, verwendet das Installationsprogramm die Standardauswahl. Wenn beispielsweise nur ein Rechenzentrum oder ein Cluster vorhanden ist, verwendet das Installationsprogramm automatisch diese Werte. Sie müssen den entsprechenden Datenspeicher und das entsprechende Netzwerk für die Bereitstellung auswählen und außerdem sicherstellen, dass die virtuelle IP-Adresse für den API- und *apps-Platzhalter den Einträgen in DNS entspricht.

Die virtuellen Bootstrap- und Control Plane-Rechner werden durch Aufrufen der Werte konfiguriert, die in den interaktiven Eingabeaufforderungen des OpenShift-Installationsprogramms eingegeben wurden.

- Das Red Hat Enterprise Linux CoreOS-Image wird heruntergeladen und zum Erstellen der virtuellen Rechner verwendet. Nachdem die virtuellen Bootstrap- und Control Plane-Rechner geklont wurden, werden sie automatisch eingeschaltet und starten die Bereitstellung.
- Wenn das Bootstrapping abgeschlossen ist, wird der virtuelle Rechner automatisch entfernt. Anschließend erstellen die Control Plane-Knoten die Worker-Knoten mit demselben Template, das zuvor heruntergeladen wurde.

Kapitel 3 | Installieren von OpenShift in einer virtualisierten Umgebung

- Verwenden Sie nach dem Erstellen der Worker-Knoten und dem Abschluss der Bereitstellung das Passwort „kubeadm“, um sich beim Cluster anzumelden.

Für die Ingress- und API-Adressen sind statische IP-Adressen erforderlich.

- Für die Kommunikation mit der Cluster-API ist die virtuelle Ingress-IP erforderlich.
- Die API erfordert eine virtuelle IP für die Verarbeitung des Ingress-Datenverkehrs.

Das Konfigurieren von DNS-Einträgen für die statischen Ingress- und API-IPs ist für die vCenter-Instanz, die Ihren OpenShift Container Platform-Cluster hostet, obligatorisch.

- Die DNS-Einträge haben die folgende Form: <component>.<cluster_name>.<base_domain>.

Ein DHCP-Server ist für das Netzwerk erforderlich und stellt den Cluster-Rechnern persistente IP-Adressen zur Verfügung.

Erforderliche vCenter-Kontoberechtigungen

vCenter erfordert einen spezifischen Zugriff auf ein Konto, das mit Berechtigungen zum Lesen und Erstellen von Ressourcen, wie z. B. VMs, konfiguriert ist.

- Sie können einem Konto Administratorberechtigungen erteilen, anstatt bestimmte Berechtigungen zu gewähren. Es kann jedoch unbeabsichtigte Folgen für die Sicherheit haben, wenn mehr Zugriff gewährt wird, als erforderlich ist.

Überprüfen Sie vor der Bereitstellung die entsprechenden Dokumentation.

Checkliste für obligatorische Anforderungen zur Full-Stack-Installation von OpenShift auf vSphere

Im Folgenden sind die obligatorischen Anforderungen aufgeführt, die der Cluster-Administrator berücksichtigen muss, bevor er einen OpenShift-Cluster mit der Full-Stack-Automatisierungsmethode auf vSphere bereitstellt.

vSphere-Version

Die folgenden vSphere-Versionen werden unterstützt:

- VMware vSphere 6.5
- VMware vSphere 6.7
- VMware vSphere 7.0

vSphere Data Center

- vCenter hat nur ein Data Center.
- Das vSphere Data Center hat nur einen Cluster.

vSphere-Datenspeicher

Erstellen Sie einen Datenspeicher für persistente Volumes, auf den alle Rechnerknoten im Rechenzentrum für die Day 1-Installation zugreifen können.

vSphere-VM-Ordner

Erstellen Sie einen neuen Ordner, der alle Cluster-VMs enthält.

Ressourcenpool

Die Binärdatei des Installationsprogramms erstellt den Rechner im Standardressourcenpool.

Kapitel 3 | Installieren von OpenShift in einer virtualisierten Umgebung

- Die Konfiguration eines anderen Ressourcenpools ist **NICHT** möglich und muss bei der Planung einer OpenShift-Cluster-Bereitstellung berücksichtigt werden.

Physische Hosts

Verwenden Sie separate physische Hosts für die Cluster-Rechner, um die Hochverfügbarkeit des Clusters zu gewährleisten.

VM/VM-Anti-Affinität

Konfigurieren von Anti-Affinitätsregeln für Control Plane- und Infra-VMs, wenn sie sich nicht auf demselben physischen Host befinden.

DRS deaktiviert

VMware vMotion ist dazu vorgesehen, eine Live-Migration virtueller Rechner zwischen Hosts bereitzustellen und gleichzeitig Ausfallzeiten zu vermeiden. Der VMware Distributed Resource Scheduler (DRS) ist für Control Plane- und Infra-Worker-Knoten deaktiviert.

- Vor der Migration von Server-Worker-Knoten wird ein Verfahren zum Leeren von Knoten implementiert.
- In VMware DRS werden virtuelle Rechner migriert, wenn sie durch das Auslastungsniveau auf bestimmten Hosts ausgelöst werden.
- Eine VMware DRS-Migration kann wiederum eine andere VMware DRS-Migration auslösen, wenn der zu migrierende OpenShift-Knoten in den Zustand **NotReady** wechselt und die Workloads dieses Knotens auf einem anderen Knoten geplant werden.



Anmerkung

vMotion ist bisher nicht umfassend mit OpenShift getestet worden. Es wird jedoch eine Anfrage zur Funktionserweiterung in Betracht gezogen, um das formale Testen von vMotion mit OpenShift zu adressieren. Red Hat empfiehlt die Verwendung von vMotion derzeit nicht.

Netzwerk

Konfigurieren Sie DHCP, DHCP IP-Reservierungen oder statische IP-Adressen für die Cluster-VMs.

Image-Registry

Datei-Storage wird als Storage-Technologie für die hochverfügbare clusterinterne Registry installiert.

- NFS auf RHEL wird nicht unterstützt.
- Andere NFS-Implementierungen werden unterstützt (NetApp, HPE, DELL usw.).
- OCS mit CephFS wird unterstützt.

Überwachung und Protokollierung

Installieren Sie Block-Storage für Überwachung und Protokollierung.

Storage-Bereitstellungsstrategie

Wählen Sie eine statische oder eine dynamische Storage-Bereitstellungsstrategie.

vSphere-Konto

Erstellen Sie ein Servicekonto in vSphere mit den Rollen und Berechtigungen, die auf der Webseite „vSphere Storage for Kubernetes Permission“ angegeben sind: <https://vmware.github.io/vsphere-storage-for-kubernetes/documentation/vcp-roles.html>

Neustartrichtlinie

Konfigurieren Sie die Neustartrichtlinie in der folgenden Reihenfolge:

1. Server-Knoten anhalten
2. Infra-Knoten anhalten
3. Masterknoten anhalten
4. Masterknoten starten
5. Infra-Knoten starten
6. Server-Knoten starten

Checkliste mit Empfehlungen für die Installation von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierung

In diesem Abschnitt werden die Empfehlungen für eine Full-Stack-Installation von OpenShift auf vSphere beschrieben.

- Es werden drei zusätzliche Worker-Rechner (CPU: 4 Kerne, RAM: 24 GB, Storage: 120 GB) für Infra (Registry, HAProxy usw.) bereitgestellt.
- Die anderen Worker-Rechner sind für Computing (Computing-Rechner) vorgesehen.

Installieren von OpenShift mit Full-Stack-Automatisierung auf vSphere

- Laden Sie die Installationsprogramm-Binärdatei „oc tools“ herunter, und rufen Sie das Secret von der Red Hat OpenShift Cluster Manager-Site ab.
- Laden Sie die Root-CA-Zertifikate von vCenter herunter, und fügen Sie diese Ihrer Bastion-VM hinzu.
- Führen Sie die Binärdatei „openshift-install“ aus, um die Datei „install-config.yaml“ zu erstellen.
- Geben Sie Werte ein, wenn sie von der Installationsprogramm-Binärdatei dazu aufgefordert werden.
- Führen Sie die Installationsprogramm-Binärdatei aus, um den Cluster zu erstellen.



Anmerkung

Zu dem Zeitpunkt, an dem dieser Kurs erstellt wurde, liegt ein Fehler auf „try.openshift.com/cloud.redhat.com“ vor, bei dem als Verfügbarkeit für vSphere nur UPI angezeigt wird. Das "pull-secret" und die Tools auf der bereits vorhandenen Infrastrukturwebseite sind jedoch auch für Full-Stack-Bereitstellungen geeignet.

- Installieren Sie die Binärdatei „openshift-install“ und oc-Tools auf dem Bastion-Host.

```
[user@bastion ~]$ sudo -i
[root@bastion ~]# OCP_VERSION=4.6.4
[root@bastion ~]# MIRROR=mirror.openshift.com/pub/openshift-v4/clients
[root@bastion ~]# wget \
```

Kapitel 3 | Installieren von OpenShift in einer virtualisierten Umgebung

```
> https://[MIRROR]/ocp/${OCP_VERSION}/openshift-install-linux-${OCP_VERSION}.tar.gz
[root@bastion ~]# tar zxvf openshift-install-linux-${OCP_VERSION}.tar.gz \
> -C /usr/bin
[root@bastion ~]# rm -f openshift-install-linux-${OCP_VERSION}.tar.gz
[root@bastion ~]# chmod +x /usr/bin/openshift-install
[root@bastion ~]# openshift-install version
openshift-install 4.6.4
built from commit 6e02d049701437fa81521fe981405745a62c86c5
release image quay.io/openshift-release-dev/ocp-release@sha256:668...6fc
```

```
[root@bastion ~]# wget \
> https://[MIRROR]/ocp/${OCP_VERSION}/openshift-client-linux-${OCP_VERSION}.tar.gz
[root@bastion ~]# tar zxvf openshift-client-linux-${OCP_VERSION}.tar.gz \
> -C /usr/bin
[root@bastion ~]# rm -f openshift-client-linux-${OCP_VERSION}.tar.gz
[root@bastion ~]# chmod +x /usr/bin/oc
[root@bastion ~]# oc completion bash >/etc/bash_completion.d/openshift
[root@bastion ~]# oc version
Client Version: 4.6.4
```

- Generieren Sie einen SSH-Schlüssel auf dem Bastion-Host.

```
[root@bastion ~]# su - user
[user@bastion ~]$ ssh-keygen -f ${HOME}/.ssh/ocp46-key -N ''
```

- Laden Sie die Root-CA-Zertifikate von vCenter herunter und extrahieren Sie sie.

```
[user@bastion ~]$ sudo wget \
> vcenter.sddc.vmwaremc.com/downloads/certs/download.zip
...output omitted...
Saving to: `download.zip'
100% [=====] 5,708 --.-K/s in 0s
2020-10-09 22:08:15 (781 MB/s) - `download.zip` saved [6708/6708]
```

```
[user@bastion ~]$ sudo unzip download.zip
Archive: download.zip
  inflating: download/certs//lin/000cec1a.0
  inflating: download/certs//mac/000cec1a.0
  inflating: download/certs//win/000cec1a.0.crt
  inflating: download/certs//lin/000cec1a.r0
  inflating: download/certs//mac/000cec1a.r0
  inflating: download/certs//win/000cec1a.r0.cr1
```

- Aktualisieren Sie die Linux-Zertifikate, um die vCenter-Authentifizierung zu aktivieren. vCenter erfordert eine bestehende Vertrauensbeziehung zum Konfigurationshost. Ohne diese Vertrauensbeziehung stellt das Installationsprogramm keine Verbindung zu vCenter her.

```
[user@bastion ~]$ sudo cp certs/lin/* /etc/pki/ca-trust/source/anchors
[user@bastion ~]$ sudo update-ca-trust extract
```

- Erstellen Sie ein Installationsverzeichnis mit dem Namen ocp46.

```
[user@bastion ~]# mkdir ocp46
```

- Führen Sie die Binärdatei „openshift-install“ aus, um die Datei „install-config.yaml“ zu erstellen.

```
[user@demo ~]$ ./openshift-install create install-config --dir=ocp46 \
> --log-level=info
? SSH Public Key /user/.ssh/ocp4ipi.pub
? Platform vsphere
? vCenter vcenter.vmwarecloud.com
? Username cloudadmin@vmclocal.com
Info connecting to vcenter vcenter.vmwarecloud.com
Info Defaulting to only available datacenter: SDDC
Info Defaulting to only available cluster: cluster1
? Default Datastore: WorkloadData
? Network network-segment1
? Virtual IP Address for API 192.168.1.100
? Virtual IP Address for Ingress 192.168.1.110
? Base Domain example.com
? Cluster Name ocp4
? Pull Secret ...output omitted...
```

- Führen Sie die Binärdatei „openshift-install“ aus, um den Cluster bereitzustellen.

```
[user@bastion ~]$ ./openshift-install create cluster --dir=ocp46 \
> --log-level=debug
```

Video



Literaturhinweise

- Weitere Informationen finden Sie im Kapitel *Installing on vSphere* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_vsphere/index

► Quiz

Beschreiben der Installation von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierungsmethode

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Bei der Full-Stack-Automatisierung ist der Cluster-Administrator für welche der folgenden Aufgaben nicht zuständig?
 - a. Überprüfung der ordnungsgemäßen Konfiguration von DNS
 - b. Sicherstellen, dass DHCP ordnungsgemäß konfiguriert ist
 - c. Erstellen der virtuellen Cluster-Rechner

- ▶ 2. Wie viele Rechenzentren können bei der Full-Stack-Automatisierung mit vCenter verwendet werden?
 - a. Nur zwei
 - b. Zwei oder mehr
 - c. Nur eins

- ▶ 3. Welche Zertifikate sind bei der Full-Stack-Automatisierung auf dem virtuellen Bastion-Rechner erforderlich, bevor die Installationskonfiguration beginnt?
 - a. Terraform-Zertifikate
 - b. Datenspeicherzertifikate
 - c. vCenter-Root-CA-Zertifikate

- ▶ 4. Was ist die empfohlene Neustartrichtlinie für die Full-Stack-Automatisierung in vSphere?
 - a. Infra-Knoten anhalten → Masterknoten anhalten → Server-Knoten anhalten → Masterknoten starten → Infra-Knoten starten → Server-Knoten starten
 - b. Masterknoten anhalten → Infra-Knoten anhalten → Server-Knoten anhalten → Server-Knoten starten → Infra-Knoten starten → Master-Knoten starten
 - c. Server-Knoten anhalten → Infra-Knoten anhalten → Masterknoten anhalten → Masterknoten starten → Infra-Knoten starten → Server-Knoten starten

► Lösung

Beschreiben der Installation von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierungsmethode

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Bei der Full-Stack-Automatisierung ist der Cluster-Administrator für welche der folgenden Aufgaben nicht zuständig?
 - a. Überprüfung der ordnungsgemäßen Konfiguration von DNS
 - b. Sicherstellen, dass DHCP ordnungsgemäß konfiguriert ist
 - c. Erstellen der virtuellen Cluster-Rechner

- ▶ 2. Wie viele Rechenzentren können bei der Full-Stack-Automatisierung mit vCenter verwendet werden?
 - a. Nur zwei
 - b. Zwei oder mehr
 - c. Nur eins

- ▶ 3. Welche Zertifikate sind bei der Full-Stack-Automatisierung auf dem virtuellen Bastion-Rechner erforderlich, bevor die Installationskonfiguration beginnt?
 - a. Terraform-Zertifikate
 - b. Datenspeicherzertifikate
 - c. vCenter-Root-CA-Zertifikate

- ▶ 4. Was ist die empfohlene Neustartrichtlinie für die Full-Stack-Automatisierung in vSphere?
 - a. Infra-Knoten anhalten → Masterknoten anhalten → Server-Knoten anhalten → Masterknoten starten → Infra-Knoten starten → Server-Knoten starten
 - b. Masterknoten anhalten → Infra-Knoten anhalten → Server-Knoten anhalten → Server-Knoten starten → Infra-Knoten starten → Master-Knoten starten
 - c. Server-Knoten anhalten → Infra-Knoten anhalten → Masterknoten anhalten → Masterknoten starten → Infra-Knoten starten → Server-Knoten starten

Beschreiben der Installation von OpenShift auf vSphere bei bereits vorhandener Infrastruktur

Ziele

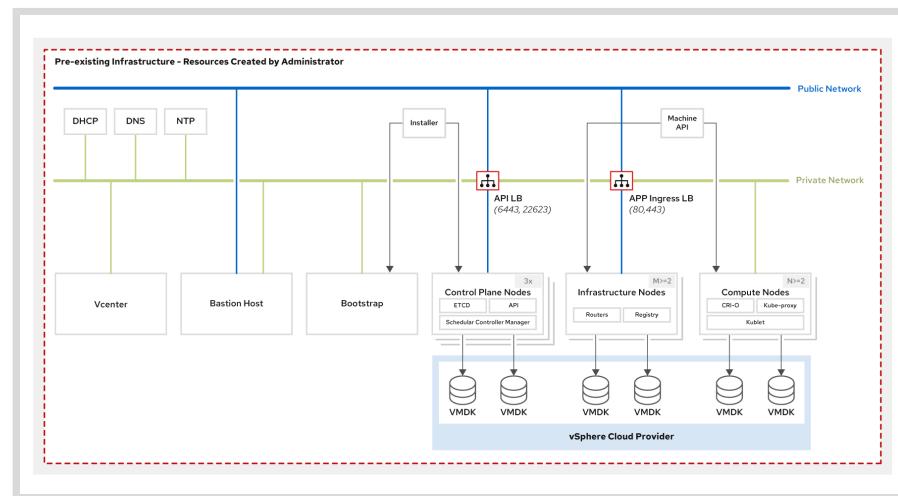
Nach Abschluss dieses Abschnitts sollten Sie die Installation von OpenShift auf vSphere unter Verwendung einer bereits vorhandenen Infrastruktur mit allgemeinen Anpassungen beschreiben können.

Installieren von OpenShift auf vSphere bei bereits vorhandener Infrastruktur

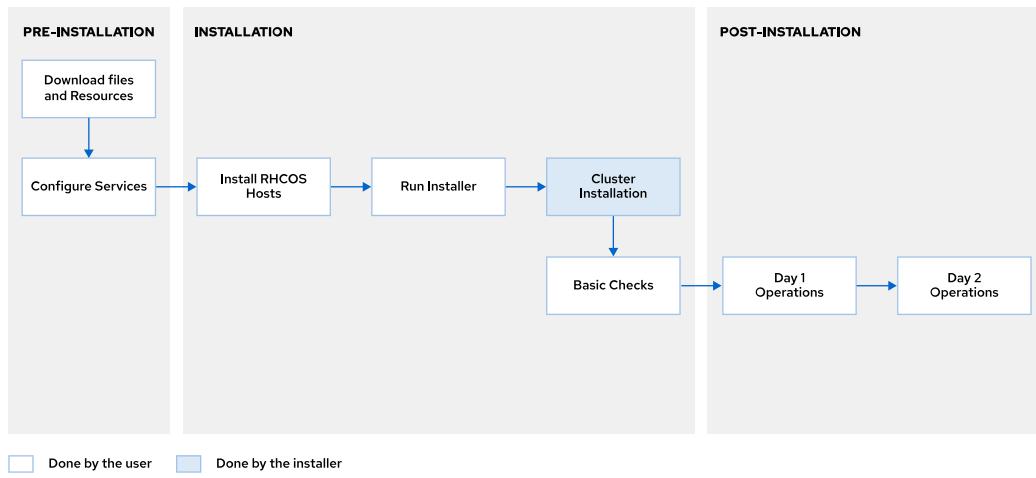
Der Administrator muss vor der Bereitstellung des Clusters sicherstellen, dass die spezifischen Voraussetzungen für vSphere-Infrastruktur erfüllt sind.

Beschreiben der OpenShift-Architektur auf vSphere bei bereits vorhandener Infrastruktur

Der Administrator ist für die Architekturerressourcen verantwortlich, die im folgenden Diagramm einer OpenShift-Bereitstellung mit einer bereits vorhandenen Infrastruktur in vSphere dargestellt werden:

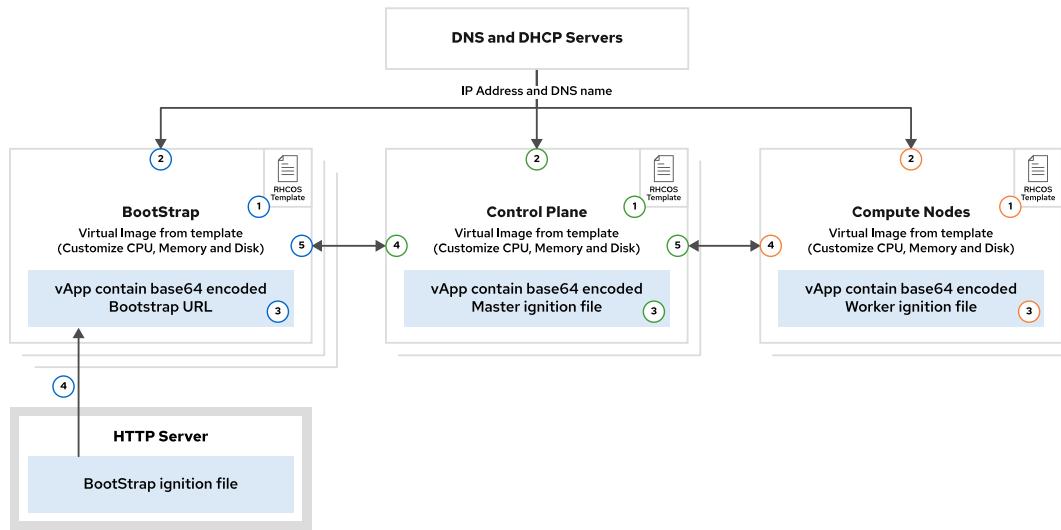


Das folgende Diagramm zeigt die verschiedenen Phasen des Installationsprozesses unter Verwendung einer vorhandenen Infrastruktur.



Beschreiben des Bereitstellungs-Workflows mit einer bereits vorhandenen Infrastruktur auf vSphere

Das folgende Diagramm zeigt den Workflow für die Bereitstellung einer OpenShift-Architektur bei einer bereits vorhandenen Infrastruktur auf vSphere.



Dieser Abschnitt bietet einen Überblick über die Prozessschritte zur Installation von OpenShift unter Verwendung einer bereits vorhandenen Infrastruktur auf vSphere.

Beachten Sie, dass die Zahlen im vorherigen Workflow-Diagramm zu den folgenden nummerierten Schritten passen.

1. Schalten Sie den Bootstrap-Knoten ein, um das RHCOS-Template-Image zu initialisieren.
2. Der Bootstrap-Knoten empfängt eine IP-Adresse vom DHCP-Server. Der DNS-Server enthält einen Eintrag, der den Bootstrap-Hostnamen der entsprechenden IP-Adresse zuordnet.
3. Die vApp gibt die base64-codierte URL für die Bootstrap-Ignition-Konfigurationsdatei an, die sich auf dem HTTPD-Server befindet.

Kapitel 3 | Installieren von OpenShift in einer virtualisierten Umgebung

4. Der Bootstrap-Knoten lädt die Ignition-Datei herunter, die sich unter der in der vApp angegebenen URL befindet.
5. Mit den in der Ignition-Konfigurationsdatei angegebenen Informationen beginnen die Control Plane-Knoten mit der Installation von RHCOS und empfangen die Ignition-Dateien vom Kubernetes-API-MCS, der auf dem Bootstrap-Knoten ausgeführt wird.

Für jedes VM-Image der Control Plane:

1. Das RHCOS-Template-Image wird abgeschlossen und der Knoten der Control Plane bootet.
2. Der Control Plane-Knoten empfängt eine IP-Adresse vom DHCP-Server.
3. Die vApp gibt die base64-codierten Ignition-Konfigurationsinformationen an, die zur Installation der Control Plane-Knoten verwendet werden.
4. Die Control Plane-Knoten rufen ihre Ignition-Konfigurationsdateien vom Bootstrap-Knoten ab.
5. Die Server-Knoten rufen ihre Ignition-Konfigurationsdateien vom Kubernetes-API-MCS ab, der auf den Control Plane-Knoten ausgeführt wird, nachdem die Control Plane vom Bootstrap-Knoten auf die Control Plane übertragen wurde.

Checkliste für obligatorische Anforderungen zur Installation von OpenShift auf vSphere bei bereits vorhandener Infrastruktur

In diesem Abschnitt werden die Anforderungen für die Installation von OpenShift auf vSphere bei bereits vorhandener Infrastruktur beschrieben.

Rechner

Die folgenden Rechner werden bereitgestellt:

- Ein Bootstrap-Rechner (CPU: 4 Kerne, RAM: 16 GB, Storage: 120 GB) – Dieser Rechner wird nach der Installation entfernt.
- Drei Control Plane-Rechner (CPU: 4 Kerne, RAM: 16 GB, Storage: 120 GB)
- Zwei Computing-Rechner (CPU: 2 Kerne, RAM: 8 GB, Storage: 120 GB)

vSphere-Version

Die folgenden vSphere-Versionen werden unterstützt:

- VMware vSphere 6.5
- VMware vSphere 6.7
- VMware vSphere 7.0

Netzwerk

Für jede VM müssen der DHCP-Zugriff, DHCP IP-Reservierungen oder statische IP-Adressen konfiguriert sein.

Netzwerkports

Für eingehenden Datenverkehr werden die folgenden Ports geöffnet:

- Alle Rechner:
 - ICMP

- TCP:22
- Alle Rechner mit Ausnahme des Bastion-Rechners:
 - TCP: 9000-9999, 10249-10259, 30000-32767
 - UDP: 4789, 6081, 9000-9999, 30000-32767
- Control Plane und Bootstrap:
 - TCP: 2379-2380, 6443, 22623
- Worker:
 - TCP: 443. Wenn die optionale Konfiguration des Infra-Worker-Rechners gewählt wird, öffnen Sie Port 443 nur für den Infra-Worker.
Für ausgehenden Datenverkehr werden alle Ports geöffnet.

API-Load Balancer

- Ein hochverfügbarer Logical Layer-4-Load Balancer wird mit 2 virtuellen IPs (VIPs) konfiguriert:
 - VIP1 (Externe API): Datenverkehr wird an alle IPs der Control Plane-Knoten und die Bootstrap-Knoten-IP mit der folgenden Portzuordnung weitergeleitet: 6443 → 6443
 - VIP2 (Interne API): Datenverkehr wird an alle IPs der Control Plane-Knoten und die Bootstrap-Knoten-IP mit der folgenden Portzuordnung weitergeleitet: 6443 → 6443, 22623 → 22623
- Die Umleitung zur Bootstrap-Knoten-IP wird während der Installation entfernt.
- Layer-7-konfigurierte Load Balancer werden nicht unterstützt.
- Beide VIP-Health Checks werden wie folgt konfiguriert:
 - Der Endpunkt für den API-Server ist bereit.
 - Das Timeout wird auf 30 Sekunden festgelegt, bevor die API-Serverinstanz entfernt wird.
 - Die Abfrage erfolgt alle 10 Sekunden, wobei zwei erfolgreiche Anfragen erforderlich sind, um den Zustand "healthy" zu erreichen, und drei erfolglose Anfragen, um den Zustand "unhealthy" zu erreichen.
- Der Load-Balancing-Algorithmus ist zustandslos.

Apps-Load Balancer

- Ein hochverfügbarer Logical Layer-4-Load Balancer wird mit einer virtuellen IP (VIP) konfiguriert.
 - Die Anwendungs-VIP verteilt den Datenverkehr an alle Server-Knoten-IPs mit der folgenden Portzuordnung: 443 → 443
- Die Umleitung zu Nicht-Infra-Knoten wird nach der Konfiguration des Infra-Knotens entfernt.
- Layer-7-konfigurierte Load Balancer werden nicht unterstützt.

Kapitel 3 | Installieren von OpenShift in einer virtualisierten Umgebung

- Der grundlegende VIP-Health Check wird konfiguriert.
- Der Load-Balancing-Algorithmus ist zustandslos.

Externe DNS-Einträge

Folgende externe DNS-Einträge sind erforderlich:

- API:
 - api.<cluster_name>.<base_domain>: VIP1
 - *.apps.<cluster_name>.<base_domain>: VIP2

Die DNS-Einträge müssen von beiden Clients außerhalb des Clusters und von allen Knoten im Cluster auflösbar sein.

Interne DNS-Einträge

Folgende interne DNS-Einträge sind erforderlich:

- API:
 - api-int.<cluster_name>.<base_domain>: VIP3

Die DNS-Einträge müssen von allen Knoten im Cluster auflösbar sein.

HTTP-Server

Der HTTP-Server muss über den Bastion-Rechner, auf dem die Installation ausgeführt wird, und über alle installierten Cluster-Rechner erreichbar sein.

- Wenn der HTTP-Server nicht verfügbar ist, verwenden Sie den Bastion-Rechner als HTTP-Server.
- Stellen Sie sicher, dass Port 80 auf dem Bastion-Rechner für eingehenden Datenverkehr geöffnet ist.

Red Hat Enterprise Linux CoreOS

Stellen Sie die Red Hat Enterprise Linux CoreOS-VMs aus der Red Hat Enterprise Linux CoreOS OVA-Datei oder ISO mit iPXE-Dateien bereit, die von Red Hat bereitgestellt werden.

Latenzempfindlichkeit

Legen Sie die Latenzempfindlichkeit für alle VMs als **Hoch** fest, mit Ausnahme des Bastion-Rechners.

Image-Registry-Storage

Wenn Dateispeicher als Storage-Technologie für die interne Registry auf einem hochverfügbaren Cluster installiert ist, dann gilt Folgendes:

- NFS auf RHEL wird nicht unterstützt.
- Andere NFS-Implementierungen werden unterstützt, z. B. NetApp, HPE und DELL.
- OCS mit CephFS wird unterstützt.

Storage-Bereitstellungsstrategie

Wählen Sie eine statische oder eine dynamische Storage-Bereitstellungsstrategie.

vSphere Datastore

Erstellen Sie einen Datenspeicher für persistente Volumes, auf den alle Rechnerknoten im Rechenzentrum für die Day 1-Installation zugreifen können.

vSphere-VM-Ordner

Erstellen Sie einen Ordner, der alle VMs enthält, die für die Bereitstellung erstellt werden.

Physische Hosts

Verwenden Sie separate physische Hosts für die Cluster-Rechner, um die Hochverfügbarkeit des Clusters zu gewährleisten.

VM/VM-Anti-Affinität

Konfigurieren von Anti-Affinitätsregeln für Control Plane- und Infra-VMs, wenn sie sich nicht auf demselben physischen Host befinden.

DRS deaktiviert

VMware vMotion ist dazu vorgesehen, eine Live-Migration virtueller Rechner zwischen Hosts bereitzustellen und gleichzeitig Ausfallzeiten zu vermeiden. Der VMware Distributed Resource Scheduler (DRS) ist für Control Plane- und Infra-Worker-Knoten deaktiviert.

- Vor der Migration von Server-Worker-Knoten wird ein Verfahren zum Leeren von Knoten implementiert.
- In VMware DRS werden virtuelle Rechner migriert, wenn sie durch das Auslastungsniveau auf bestimmten Hosts ausgelöst werden.
- Eine VMware DRS-Migration kann wiederum eine andere VMware DRS-Migration auslösen, wenn der zu migrierende OpenShift-Knoten in den Zustand `NotReady` wechselt und die Workloads dieses Knotens auf einem anderen Knoten geplant werden.

**Anmerkung**

vMotion ist bisher nicht umfassend mit OpenShift getestet worden. Es wird jedoch eine Anfrage zur Funktionserweiterung in Betracht gezogen, um das formale Testen von vMotion mit OpenShift zu adressieren. Red Hat empfiehlt die Verwendung von vMotion derzeit nicht.

vSphere-Konto

Erstellen Sie ein Servicekonto in vSphere mit den Rollen und Berechtigungen, die auf der Webseite „vSphere Storage for Kubernetes Permission“ angegeben sind: <https://vmware.github.io/vsphere-storage-for-kubernetes/documentation/vcp-roles.html>

Neustartrichtlinie

Konfigurieren Sie die Neustartrichtlinie in der folgenden Reihenfolge:

1. Server-Knoten anhalten
2. Infra-Knoten anhalten
3. Control Plane-Knoten anhalten
4. Control Plane-Knoten starten
5. Infra-Knoten starten
6. Server-Knoten starten

Installieren eines OpenShift-Clusters auf vSphere bei bereits vorhandener Infrastruktur

- Laden Sie die Installationsprogramm-Binärdatei „oc tools“ herunter, und rufen Sie das Pull Secret von der Red Hat OpenShift Cluster Manager-Site ab.

Kapitel 3 | Installieren von OpenShift in einer virtualisierten Umgebung

- Laden Sie die Root-CA-Zertifikate für vCenter herunter, und fügen Sie diese Ihrer Bastion-VM hinzu.
- Laden Sie das Bare-Metal-Installations-Image herunter.
- Überprüfen Sie die Konfigurationen für den DNS-, HAProxy- und Apache-Webserver.
- Erstellen Sie die Datei `install-config.yaml` manuell.
- Erstellen Sie die Manifestdateien.
- Ändern Sie die Datei `manifest/cluster-schedular-02` so, dass sie `False` (mit Großbuchstabe) verwenden.
- Erstellen Sie die Ignition-Dateien.
- Stellen Sie die Ignition-Dateien über HTTPD zur Verfügung.
- Erstellen Sie die Datei `append-bootstrap.ign`, um auf die URL für `bootstrap.ign` zu verweisen.
- Konvertieren Sie die Ignition-Dateien mit base64-Codierung.
- Laden Sie die Datei `RHCOS.ova` herunter, und stellen Sie das Template bereit.
- Klonen Sie das Template, um Cluster-VMs zu erstellen.
- Fügen Sie den entsprechenden VMs die base64-Codierung der Ignition-Dateien hinzu.
- Schalten Sie die VM ein, und übergeben Sie Kernel-Line-Argumente, um die Cluster-VMs anzupassen.
- Verwenden Sie die Binärdatei „`openshift-install`“, um den Bootstrapping-Prozess zu überwachen.
- Entfernen Sie nach Abschluss des Bootstrappings den Bootstrap-Eintrag aus dem Load Balancer.
- Verwenden Sie die Binärdatei „`openshift-install`“, um den Bereitstellungsfortschritt bis zum Abschluss zu überwachen.

Video



Literaturhinweise

- Weitere Informationen finden Sie im Leitfaden *Installing vSphere* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_vsphere/index

► Quiz

Beschreiben der Installation von OpenShift auf vSphere bei bereits vorhandener Infrastruktur

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

► 1. Welche der folgenden Aufgaben liegen bei der Installation von OpenShift auf einer bereits vorhandenen Infrastruktur in der Verantwortung des Cluster-Administrators?

- a. Erstellen von virtuellen Cluster-Rechnern
- b. Konfigurieren eines Load Balancers
- c. Konfigurieren eines Apache-Servers für die Verteilung von Ignition-Dateien
- d. Erstellen der Ignition-Dateien
- e. Konfigurieren von NTP
- f. Alle oben genannten

► 2. Welche Knoten empfangen bei der Bereitstellung in einer bereits vorhandenen Infrastruktur Datenverkehr von der Load-Balancer-Anwendungs-VIP der Apps?

- a. Control Plane-Knoten
- b. Server-Knoten
- c. Operator-Knoten
- d. Firewall-Knoten

► 3. Welcher interne DNS-Eintrag ist bei Verwendung einer bereits vorhandenen Infrastruktur erforderlich?

- a. api.<cluster_name>.<base_domain>
- b. *apps.<cluster_name>.<base_domain>
- c. api-int.<cluster_name>.<base_domain>
- d. api.<base-domain>.<cluster_name>

► 4. An welche Knoten und Ports verteilt die interne Load Balancer-API in einer bereits vorhandenen Infrastruktur den Datenverkehr?

- a. Datenverkehr wird an alle IPs der Control Plane-Knoten und die Bootstrap-Knoten-IP mit der folgenden Portzuordnung weitergeleitet: 6443 → 6443
- b. Datenverkehr wird an alle IPs der Control Plane-Knoten und die Bootstrap-Knoten-IP mit der folgenden Portzuordnung weitergeleitet: 6443 → 6443, 22623 → 22623
- c. Datenverkehr wird an die IP des Bastion-Knotens und die Server-Knoten-IPs mit der folgenden Portzuordnung weitergeleitet: 6443 → 6443
- d. Datenverkehr wird an die IPs der Webserver-Knoten und die Bootstrap-Knoten-IP mit der folgenden Portzuordnung weitergeleitet: 80 → 8080

► Lösung

Beschreiben der Installation von OpenShift auf vSphere bei bereits vorhandener Infrastruktur

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche der folgenden Aufgaben liegen bei der Installation von OpenShift auf einer bereits vorhandenen Infrastruktur in der Verantwortung des Cluster-Administrators?
- a. Erstellen von virtuellen Cluster-Rechnern
 - b. Konfigurieren eines Load Balancers
 - c. Konfigurieren eines Apache-Servers für die Verteilung von Ignition-Dateien
 - d. Erstellen der Ignition-Dateien
 - e. Konfigurieren von NTP
 - f. Alle oben genannten
- 2. Welche Knoten empfangen bei der Bereitstellung in einer bereits vorhandenen Infrastruktur Datenverkehr von der Load-Balancer-Anwendungs-VIP der Apps?
- a. Control Plane-Knoten
 - b. Server-Knoten
 - c. Operator-Knoten
 - d. Firewall-Knoten
- 3. Welcher interne DNS-Eintrag ist bei Verwendung einer bereits vorhandenen Infrastruktur erforderlich?
- a. api.<cluster_name>.<base_domain>
 - b. *apps.<cluster_name>.<base_domain>
 - c. api-int.<cluster_name>.<base_domain>
 - d. api.<base-domain>.<cluster_name>
- 4. An welche Knoten und Ports verteilt die interne Load Balancer-API in einer bereits vorhandenen Infrastruktur den Datenverkehr?
- a. Datenverkehr wird an alle IPs der Control Plane-Knoten und die Bootstrap-Knoten-IP mit der folgenden Portzuordnung weitergeleitet: 6443 → 6443
 - b. Datenverkehr wird an alle IPs der Control Plane-Knoten und die Bootstrap-Knoten-IP mit der folgenden Portzuordnung weitergeleitet: 6443 → 6443, 22623 → 22623
 - c. Datenverkehr wird an die IP des Bastion-Knotens und die Server-Knoten-IPs mit der folgenden Portzuordnung weitergeleitet: 6443 → 6443
 - d. Datenverkehr wird an die IPs der Webserver-Knoten und die Bootstrap-Knoten-IP mit der folgenden Portzuordnung weitergeleitet: 80 → 8080

► Quiz

Kapitelwiederholung: Installieren von OpenShift in einer virtualisierten Umgebung

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche beiden Versionen von VMware vSphere können zur Bereitstellung eines OpenShift Container Platform-Clusters verwendet werden? (Wählen Sie zwei Antworten aus.)
- a. 6
 - b. 7
 - c. 8
 - d. 9
- 2. Welche der folgenden Aussagen in Bezug auf das Entfernen des Bootstrap-Knotens aus HAProxy ist richtig?
- a. Der Bootstrap-Knoten wird nur in einem Cluster mit drei Knoten entfernt.
 - b. Der Bootstrap-Knoten wird nach Abschluss der Installation entfernt.
 - c. Die Bootstrap-Knoten wird während der Installation entfernt.
 - d. Der Bootstrap-Knoten wird nach dem ersten Neustart entfernt.
- 3. Welche der folgenden Aussagen in Bezug auf die Full-Stack-Automatisierungsinstallation von OpenShift auf Hypervisors ist richtig?
- a. Cluster-Administratoren installieren OpenShift mit minimalem manuellen Eingriff mit einer „eigenwilligen“ Methode.
 - b. Cluster-Administratoren installieren OpenShift manuell.
 - c. Cluster-Administratoren sind nur für Load-Balancer-Konfigurationen verantwortlich.
 - d. Cluster-Administratoren sind für die manuelle Erstellung von Disks virtueller Rechner verantwortlich.
- 4. Welche zwei der folgenden Gründe für die Bereitstellung von OpenShift in einer bereits vorhandenen Infrastruktur auf vSphere sind richtig? (Wählen Sie zwei Antworten aus.)
- a. Cluster-Administratoren sind flexibler und können einfacher auf Ressourcenanforderungen reagieren.
 - b. Die Verwaltung der Infrastruktur durch Cluster-Administratoren ermöglicht detaillierte Anpassungen.
 - c. Das OpenShift-Installationsprogramm verwendet die bereitgestellten Anmeldeinformationen, um sich mit dem vCenter zu verbinden, ein Template hochzuladen und dann dieses Template zu klonen, das für die Bootstrap- und Control-Plane-Knoten verwendet werden soll.
 - d. Es gibt nur zwei DNS-Anforderungen: Ingress VIP und API VIP.

- **5. Wie viele Rechenzentren kann vCenter bei der Full-Stack-Bereitstellung von OpenShift auf vSphere haben?**
- a. Nur eins
 - b. Nur zwei
 - c. Nur fünf
 - d. Unbegrenzt
- **6. Welche der folgenden Dateien werden von vCenter zur Authentifizierung heruntergeladen?**
- a. OpenShift-Zertifikate
 - b. OpenStack-Root-CA-Zertifikate
 - c. vCenter-Root-CA-Zertifikate
 - d. OpenShift-Cluster-SSL-Zertifikate

► Lösung

Kapitelwiederholung: Installieren von OpenShift in einer virtualisierten Umgebung

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche beiden Versionen von VMware vSphere können zur Bereitstellung eines OpenShift Container Platform-Clusters verwendet werden? (Wählen Sie zwei Antworten aus.)
- a. 6
 - b. 7
 - c. 8
 - d. 9
- 2. Welche der folgenden Aussagen in Bezug auf das Entfernen des Bootstrap-Knotens aus HAProxy ist richtig?
- a. Der Bootstrap-Knoten wird nur in einem Cluster mit drei Knoten entfernt.
 - b. Der Bootstrap-Knoten wird nach Abschluss der Installation entfernt.
 - c. Die Bootstrap-Knoten wird während der Installation entfernt.
 - d. Der Bootstrap-Knoten wird nach dem ersten Neustart entfernt.
- 3. Welche der folgenden Aussagen in Bezug auf die Full-Stack-Automatisierungsinstallation von OpenShift auf Hypervisors ist richtig?
- a. Cluster-Administratoren installieren OpenShift mit minimalem manuellen Eingriff mit einer „eigenwilligen“ Methode.
 - b. Cluster-Administratoren installieren OpenShift manuell.
 - c. Cluster-Administratoren sind nur für Load-Balancer-Konfigurationen verantwortlich.
 - d. Cluster-Administratoren sind für die manuelle Erstellung von Disks virtueller Rechner verantwortlich.
- 4. Welche zwei der folgenden Gründe für die Bereitstellung von OpenShift in einer bereits vorhandenen Infrastruktur auf vSphere sind richtig? (Wählen Sie zwei Antworten aus.)
- a. Cluster-Administratoren sind flexibler und können einfacher auf Ressourcenanforderungen reagieren.
 - b. Die Verwaltung der Infrastruktur durch Cluster-Administratoren ermöglicht detaillierte Anpassungen.
 - c. Das OpenShift-Installationsprogramm verwendet die bereitgestellten Anmeldeinformationen, um sich mit dem vCenter zu verbinden, ein Template hochzuladen und dann dieses Template zu klonen, das für die Bootstrap- und Control-Plane-Knoten verwendet werden soll.
 - d. Es gibt nur zwei DNS-Anforderungen: Ingress VIP und API VIP.

- **5. Wie viele Rechenzentren kann vCenter bei der Full-Stack-Bereitstellung von OpenShift auf vSphere haben?**
- a. Nur eins
 - b. Nur zwei
 - c. Nur fünf
 - d. Unbegrenzt
- **6. Welche der folgenden Dateien werden von vCenter zur Authentifizierung heruntergeladen?**
- a. OpenShift-Zertifikate
 - b. OpenStack-Root-CA-Zertifikate
 - c. vCenter-Root-CA-Zertifikate
 - d. OpenShift-Cluster-SSL-Zertifikate

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Die Überlegungen und Optionen für die Installation von OpenShift in einem Hypervisor.
- Die Architektur einer OpenShift-Bereitstellung in vSphere mit der Full-Stack-Automatisierungsinstallationsmethode.
- Die Überlegungen, Anforderungen und allgemeinen Schritte für die Bereitstellung von OpenShift in vSphere mit der Full-Stack-Automatisierungsmethode.
- Die Architektur einer OpenShift-Bereitstellung in vSphere mit der Installationsmethode bei vorhandener Infrastruktur.
- Die Überlegungen, Anforderungen und allgemeinen Schritte für die Bereitstellung von OpenShift in einem Hypervisor mit der Installationsmethode bei vorhandener Infrastruktur.

Kapitel 4

Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

Ziel

Konfigurieren der Voraussetzungen für die Bereitstellung von OpenShift-Clustern ohne Integration in die zugrunde liegende Infrastruktur.

Ziele

- Beschreiben der Architektur und des Workflows für die Installation von OpenShift ohne Integration in die zugrunde liegende Infrastruktur.
- Bereitstellen der Voraussetzungen für die Installation von OpenShift ohne Integration in die zugrunde liegende Infrastruktur.

Abschnitte

- Einführung in die OpenShift-Installation ohne Infrastrukturanbieter (und Test)
- Konfigurieren von Netzwerkservices und Hosts für die Installation von OpenShift ohne einen Infrastrukturanbieter (und angeleitete Übung)
- Kapitelwiederholung: Planen der Installation von OpenShift ohne einen Infrastrukturanbieter (Test)

Einführung in die OpenShift-Installation ohne Infrastrukturanbieter

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die Architektur und den Workflow für die Installation von OpenShift ohne Integration in die zugrunde liegende Infrastruktur beschreiben können.

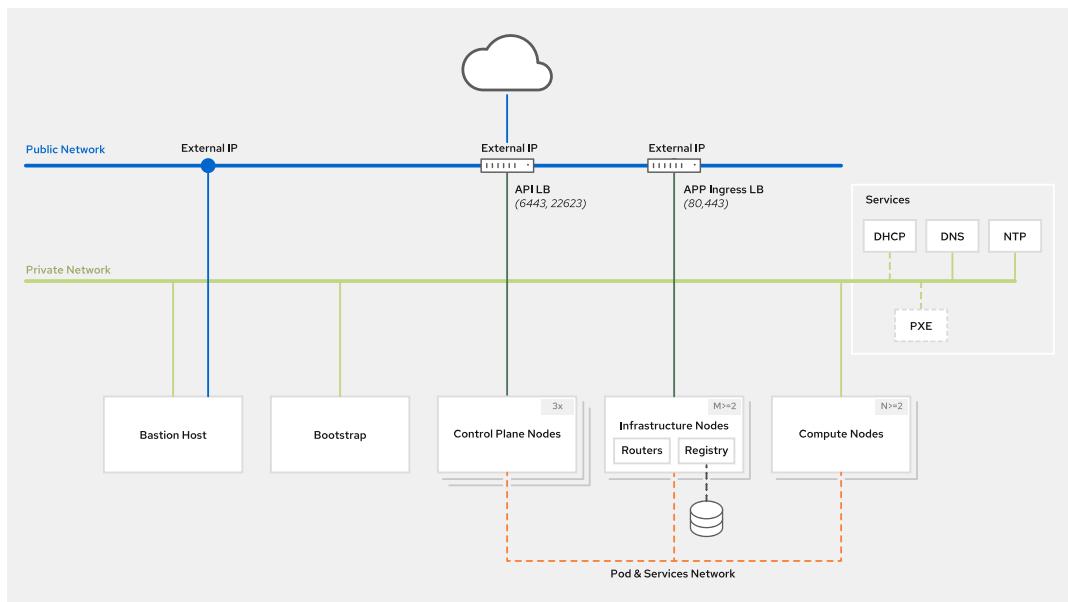
Gründe für die Installation von OpenShift ohne Integration mit einem Infrastrukturanbieter

- In einigen Fällen haben Administratoren begrenzten Zugriff auf die Infrastrukturservices und Komponenten, die für die Durchführung einer Full-Stack-Installation erforderlich sind.
- In anderen Fällen benötigen Administratoren mehr Flexibilität für die Installation von OpenShift in einer heterogenen Umgebung.

Aus diesen oder anderen Gründen können Administratoren OpenShift ohne Integration mit einem definierten Infrastrukturanbieter installieren.

Architektur einer Installation ohne Integration mit einem Infrastrukturanbieter

Das folgende Diagramm zeigt die Architektur einer Installation ohne Integration mit einem Infrastrukturanbieter:

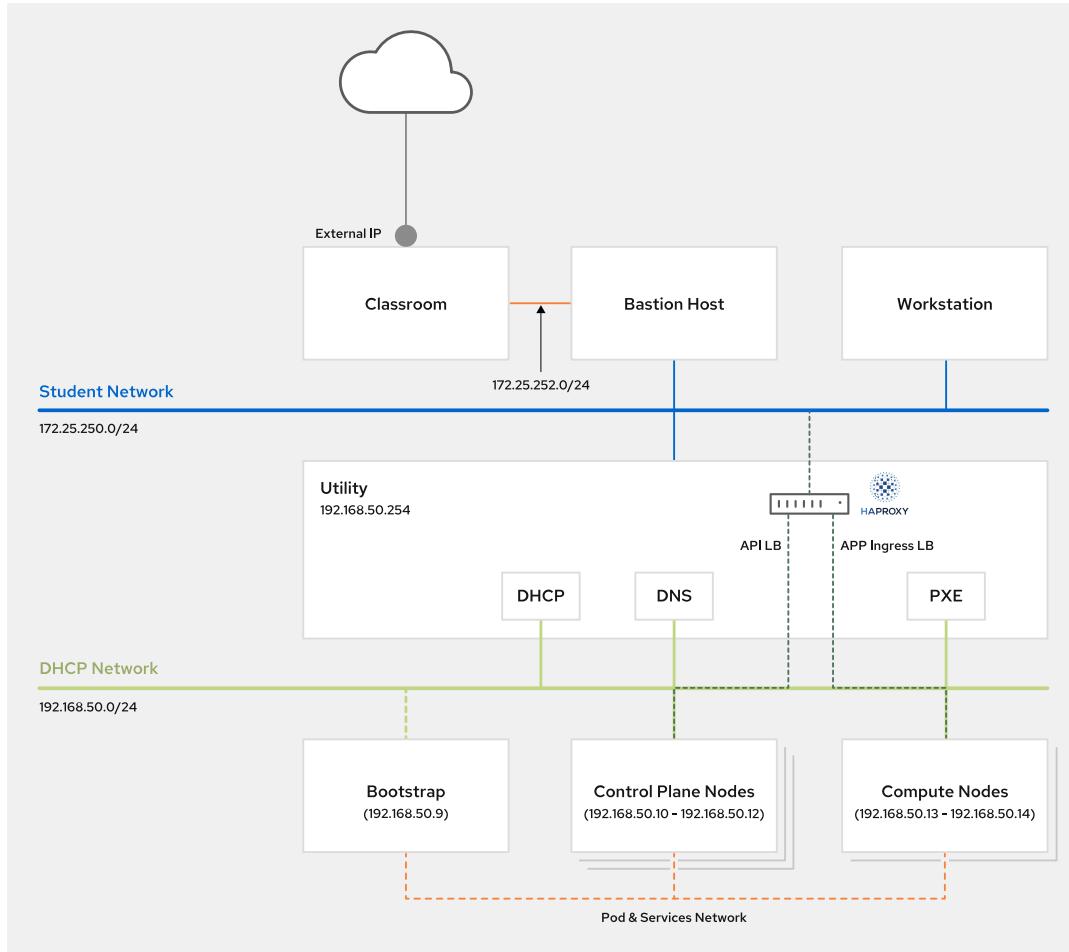


Wie Sie im Diagramm oben sehen können, sind die folgenden Ressourcen und Services für die Durchführung einer Standardinstallation ohne Integration mit einem Infrastrukturanbieter erforderlich:

- **NTP** (obligatorisch): Kann ein Server sein, der im Cluster ausgeführt wird, oder ein öffentlicher NTP-Server.
- **DHCP** (optional): Administratoren können einen DHCP-Server verwenden, um IPs dynamisch aus einem vordefinierten Bereich zu zuweisen. Alternativ können Administratoren den Servern die IPs manuell zuweisen.
- **PXE** (optional): Administratoren können einen PXE-Server für die Netzwerkbereitstellung verwenden. Alternativ können Sie die Hosts mit einem ISO-Image installieren.
- **DNS** (obligatorisch): Alle Server im Cluster müssen in der Lage sein, die Einträge der folgenden Komponenten aufzulösen: Tabelle [https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_bare_metal/index#installation-infrastructure-user-infra_installing-bare-metal].
- **Load Balancers** (obligatorisch): Bedienen die API-Ports und die Anwendungssports, wie im Folgenden dargestellt: Dokumentation [https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_bare_metal/index#installation-infrastructure-user-infra_installing-bare-metal].
- **Netzwerkkonnektivität** (obligatorisch): Die Rechner im Cluster müssen über die folgenden Ports und Protokolle miteinander kommunizieren: Tabelle [https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_bare_metal/index#installation-infrastructure-user-infra_installing-bare-metal].

Architektur der Kursumgebung

Das folgende Diagramm zeigt die Architektur der Kursumgebung für die Installation von OpenShift ohne Integration mit einem Infrastrukturanbieter:



In der Kursumgebung finden Sie die erforderlichen Services, die auf dem **utility**-Server ausgeführt werden, darunter:

- **dhcpd:** DHCP-Server
- **named:** Der DNS-Server
- **tftp:** Der Server für das PXE-Booten
- **haproxy:** Der Load Balancer für API- und Anwendungsservices
- **httpd:** Der als Dateiserver verwendete HTTP-Server



Literaturhinweise

- Weitere Informationen finden Sie im Kapitel *Installing on bare metal* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_bare_metal/index#installing-bare-metal

► Quiz

Einführung in die OpenShift-Installation ohne Infrastrukturanbieter

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche zwei der folgenden Situationen können dazu beitragen, dass ein Administrator OpenShift auf einer bereits vorhandenen Infrastruktur installiert? (Wählen Sie zwei Antworten aus.)
- Der Administrator verfügt über umfassende Flexibilität und Berechtigungen in seiner Unternehmensinfrastruktur, einschließlich einer Private Cloud-Umgebung.
 - Der Administrator hat keinen Zugriff auf eine Cloud-Umgebung und nur eingeschränkten Zugriff auf die Netzwerkconfiguration des Unternehmens.
 - Der Administrator möchte detaillierte Kontrolle über den OpenShift-Installationsprozess haben.
 - Der Administrator möchte eine voll automatisierte Methode für die Installation von OpenShift verwenden.
- 2. Welche der folgenden Punkte ist die Empfehlung von Red Hat zur Verwendung von NTP-Servern bei der Installation von OpenShift in Produktionsumgebungen?
- Ein NTP-Server ist nicht erforderlich.
 - Ein privater NTP-Server wird einem öffentlichen vorgezogen.
 - Ein öffentlicher NTP-Server wird einem privaten vorgezogen.
 - Ein NTP-Server ist erforderlich, unabhängig davon, ob er privat oder öffentlich ist.
- 3. Welche der folgenden Services sind für die Installation von OpenShift auf einer bereits vorhandenen Infrastruktur erforderlich?
- Ein PXE-Server
 - Ein DNS-Server
 - Ein DHCP-Server
- 4. Welche der folgenden Services werden vom OpenShift-Installationsprogramm bei der Installation auf einer bereits vorhandenen Infrastruktur erstellt?
- Ein HAProxy-Server
 - Ein DHCP-Server für das dynamische Zuweisen von IPs zu neuen Knoten
 - Ein keepalive-Server, um den Load Balancer des API-Servers hochverfügbar zu machen
 - Keiner der oben genannten Services

► Lösung

Einführung in die OpenShift-Installation ohne Infrastrukturanbieter

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ **1. Welche zwei der folgenden Situationen können dazu beitragen, dass ein Administrator OpenShift auf einer bereits vorhandenen Infrastruktur installiert? (Wählen Sie zwei Antworten aus.)**
 - a. Der Administrator verfügt über umfassende Flexibilität und Berechtigungen in seiner Unternehmensinfrastruktur, einschließlich einer Private Cloud-Umgebung.
 - b. Der Administrator hat keinen Zugriff auf eine Cloud-Umgebung und nur eingeschränkten Zugriff auf die Netzwerkkonfiguration des Unternehmens.
 - c. Der Administrator möchte detaillierte Kontrolle über den OpenShift-Installationsprozess haben.
 - d. Der Administrator möchte eine voll automatisierte Methode für die Installation von OpenShift verwenden.

- ▶ **2. Welche der folgenden Punkte ist die Empfehlung von Red Hat zur Verwendung von NTP-Servern bei der Installation von OpenShift in Produktionsumgebungen?**
 - a. Ein NTP-Server ist nicht erforderlich.
 - b. Ein privater NTP-Server wird einem öffentlichen vorgezogen.
 - c. Ein öffentlicher NTP-Server wird einem privaten vorgezogen.
 - d. Ein NTP-Server ist erforderlich, unabhängig davon, ob er privat oder öffentlich ist.

- ▶ **3. Welche der folgenden Services sind für die Installation von OpenShift auf einer bereits vorhandenen Infrastruktur erforderlich?**
 - a. Ein PXE-Server
 - b. Ein DNS-Server
 - c. Ein DHCP-Server

- ▶ **4. Welche der folgenden Services werden vom OpenShift-Installationsprogramm bei der Installation auf einer bereits vorhandenen Infrastruktur erstellt?**
 - a. Ein HAProxy-Server
 - b. Ein DHCP-Server für das dynamische Zuweisen von IPs zu neuen Knoten
 - c. Ein keepalive-Server, um den Load Balancer des API-Servers hochverfügbar zu machen
 - d. Keiner der oben genannten Services

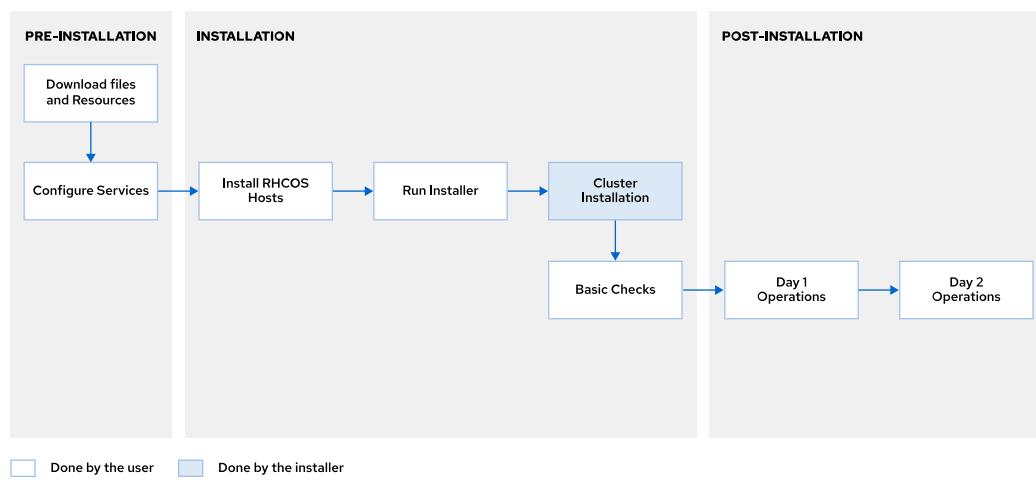
Konfigurieren von Netzwerkservices und Hosts für die Installation von OpenShift ohne einen Infrastrukturanbieter

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die Voraussetzungen für die Installation von OpenShift ohne Integration in die zugrunde liegende Infrastruktur bereitstellen können.

Übersicht über den Installationsprozess bei bereits vorhandener Infrastruktur

Das folgende Diagramm zeigt die verschiedenen Phasen des Installationsprozesses unter Verwendung einer vorhandenen Infrastruktur.



Herunterladen von Ressourcen für die Installation

- Laden Sie zunächst die binären Dateien `openshift-install` und `oc` vom OpenShift-Clients-Mirror [<https://mirror.openshift.com/pub/openshift-v4/clients/ocp/>] herunter.
- Laden Sie dann das Pull Secret von `cloud.redhat.com` [<https://cloud.redhat.com/openshift/install/metal/installer-provisioned/>] herunter. Sie müssen das Pull Secret später verwenden, um die Datei `install-config.yaml` zu vervollständigen.
- Laden Sie abschließend die RHCOS-Images für eine PXE-basierte Installation vom RHCOS-Mirror [<https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/>] herunter.

In der Kursumgebung sind eine private Registry mit einer Spiegelung der OpenShift-Release-Images und andere Ressourcen verfügbar. Die Kursumgebung hat Zugriff auf das Internet.



Anmerkung

Nach der Installation der Binärdateien `oc` und `openshift-install` ist es nützlich, die Vervollständigung der Bash-Befehle einzurichten (und zu sourcen):

```
[root@demo ~]# oc completion bash > /etc/bash_completion.d/openshift
[root@demo ~]# openshift-install completion bash \
> /etc/bash_completion.d/openshift-install
[root@demo ~]# source /etc/bash_completion.d/openshift
[root@demo ~]# source /etc/bash_completion.d/openshift-install
```

Analysieren der erforderlichen Services für die Installation

Wie an anderer Stelle erläutert, führt der `utility`-Server die erforderlichen Services für die Installation in der Kursumgebung aus. In der Kursumgebung werden alle Services bereits ausgeführt. Sie müssen jedoch einige von ihnen konfigurieren, um die Installation erfolgreich abzuschließen. Sie können die Umgebung und ihre aktuellen Konfigurationen mithilfe der entsprechenden `systemd service units` untersuchen.

- `dhcpd` – DHCP-Server

Die folgende Beispieldatei zeigt eine mögliche DHCP-Konfiguration.

```
[root@utility ~]# cat /etc/dhcp/dhcpd.conf
ddns-update-style interim;
ignore client-updates;
authoritative;
allow booting;
allow bootp;
allow unknown-clients;
# Set default and max IP lease time to infinite with -1 value
default-lease-time -1;
max-lease-time -1;

subnet 192.168.50.0 netmask 255.255.255.0 {
    option routers 192.168.50.254;
    option domain-name-servers 192.168.50.254;
    option ntp-servers 103.16.182.23,103.16.182.214;
    option domain-search "ocp4.example.com","example.com";
    filename "pxelinux.0";
    next-server 192.168.50.254;
        host master01.ocp4.example.com { hardware ethernet
52:54:00:00:32:0A; fixed-address 192.168.50.10; option host-name "master01"; }
        host master02.ocp4.example.com { hardware ethernet
52:54:00:00:32:0B; fixed-address 192.168.50.11; option host-name "master02"; }
        host master03.ocp4.example.com { hardware ethernet
52:54:00:00:32:0C; fixed-address 192.168.50.12; option host-name "master03"; }
        host bootstrap.ocp4.example.com { hardware ethernet
52:54:00:00:32:09; fixed-address 192.168.50.9; option host-name "bootstrap"; }
```

```
host worker01.ocp4.example.com { hardware ethernet  
52:54:00:00:32:0D; fixed-address 192.168.50.13; option host-name "worker01"; }  
host worker02.ocp4.example.com { hardware ethernet  
52:54:00:00:32:0E; fixed-address 192.168.50.14; option host-name "worker02"; }  
}
```



Anmerkung

Weitere Informationen finden Sie im Kapitel *Providing DHCP services* im *Red Hat Enterprise Linux 8 Configuring and Managing Networking Guide* unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_networking/index#providing-dhcp-services_configuring-and-managing-networking.

- named – DNS-Server

Die folgende Beispieldatei zeigt eine mögliche DNS-Konfiguration.

```
[root@utility ~]# cat /etc/named.conf  
  
options {  
#listen-on port 53 { 127.0.0.1; };  
#listen-on-v6 port 53 { ::1; };  
directory "/var/named";  
dump-file "/var/named/data/cache_dump.db";  
statistics-file "/var/named/data/named_stats.txt";  
memstatistics-file "/var/named/data/named_mem_stats.txt";  
secroots-file "/var/named/data/named.secroots";  
recursing-file "/var/named/data/named.recurising";  
allow-query { localhost; 192.168.50.0/24; 172.25.250.254; };  
recursion yes;  
dnssec-enable yes;  
dnssec-validation yes;  
managed-keys-directory "/var/named/dynamic";  
pid-file "/run/named/named.pid";  
session-keyfile "/run/named/session.key";  
  
forwarders {  
    172.25.250.254;  
};  
/* https://fedoraproject.org/wiki/Changes/CryptoPolicy */  
include "/etc/crypto-policies/back-ends/bind.config";  
};  
  
logging {  
    channel default_debug {  
        file "data/named.run";  
        severity dynamic;  
    };  
};  
  
zone "." IN {  
    type hint;
```

Kapitel 4 | Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

```
file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";

zone "example.com" {
    type master;
    file "example.com.db";
    allow-update { none; };
};

zone "50.168.192.in-addr.arpa" IN {
    type master;
    file "example.com.reverse.db";
    allow-update { none; };
};
```

Diese Konfigurationsdatei verweist auf die Dateien `example.com.db` (Forward-Auflösung) und `example.com.reverse.db` (Reverse-Auflösung), die im Verzeichnis `/var/named` gespeichert sind:

```
[root@utility ~]# cat /var/named/example.com.db
$TTL 1D
@ IN SOA dns.ocp4.example.com. root.example.com. (
    2019022400 ; serial
    3h          ; refresh
    15          ; retry
    1w          ; expire
    3h          ; minimum
)
IN NS dns.ocp4.example.com.
dns.ocp4      IN A 192.168.50.254
api.ocp4      IN A 192.168.50.254
api-int.ocp4  IN A 192.168.50.254
*.apps.ocp4   IN A 192.168.50.254
bootstrap.ocp4 IN A 192.168.50.9
master01.ocp4 IN A 192.168.50.10
etcd-0.ocp4   IN A 192.168.50.10
_etcfd-server-ssl._tcp.ocp4 IN SRV 0 10 2380 etcd-0.ocp4
master02.ocp4 IN A 192.168.50.11
etcd-1.ocp4   IN A 192.168.50.11
_etcfd-server-ssl._tcp.ocp4 IN SRV 0 10 2380 etcd-1.ocp4
master03.ocp4 IN A 192.168.50.12
etcd-2.ocp4   IN A 192.168.50.12
_etcfd-server-ssl._tcp.ocp4 IN SRV 0 10 2380 etcd-2.ocp4
worker01.ocp4 IN A 192.168.50.13
worker02.ocp4 IN A 192.168.50.14
```

```
[root@utility ~]# cat /var/named/example.com.reverse.db
$TTL 1D
@ IN SOA dns.ocp4.example.com. root.example.com. (
    2019022400 ; serial
```

```

        3h      ; refresh
        15      ; retry
        1w      ; expire
        3h      ; minimum
    )
IN NS dns.ocp4.example.com.
254 IN PTR api.ocp4.example.com.
254 IN PTR api-int.ocp4.example.com.
9 IN PTR bootstrap.ocp4.example.com.
10 IN PTR master01.ocp4.example.com.
11 IN PTR master02.ocp4.example.com.
12 IN PTR master03.ocp4.example.com.
13 IN PTR worker01.ocp4.example.com.
14 IN PTR worker02.ocp4.example.com.

```



Anmerkung

Weitere Informationen finden Sie im Abschnitt *BIND* im *Red Hat Enterprise Linux 7 Networking Guide* unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/networking_guide/index#sec-BIND.

- tftp – Server für das PXE-Booten

Die PXE-Boot-Dateien müssen im Verzeichnis `/var/lib/tftpboot/pxelinux.cfg/` gespeichert werden. Um für jede einzelne MAC-Adresse unterschiedliche PXE-Boot-Dateien zu verwenden, benennen Sie die Dateien `01-aa-bb-cc-dd-ee-ff`, wobei aa-bb-cc-dd-ee-ff die MAC-Adresse in Kleinbuchstaben ist und wobei Bindestriche anstelle von Doppelpunkten verwendet werden.

Ein Beispiel für eine PXE-Boot-Datei für die MAC-Adresse `52:54:00:00:32:09` ist:

```

[root@utility ~]# cat /var/lib/tftpboot/pxelinux.cfg/01-52-54-00-00-32-09
default menu.c32
prompt 0
timeout 0
menu title **** OpenShift 4 BOOTSTRAP PXE Boot Menu ****

label Install RHCOS 4.6.1 Bootstrap Node
kernel http://192.168.50.254:8080/openshift4/images/rhcos-4.6.1-x86_64-live-
kernel-x86_64
append ip=dhcp rd.neednet=1 coreos.inst.install_dev=vda console=tty0
console=ttyS0 coreos.inst=yes coreos.live.rootfs_url=http://192.168.50.254:8080/
openshift4/images/rhcos-4.6.1-x86_64-live-rootfs.x86_64.img
coreos.inst.ignition_url=http://192.168.50.254:8080/openshift4/4.6.4/ignitions/
bootstrap.ign initrd=http://192.168.50.254:8080/openshift4/images/rhcos-4.6.1-
x86_64-live-initramfs.x86_64.img

```



Anmerkung

Weitere Informationen finden Sie im Kapitel *Preparing to install from the network using PXE* im Red Hat Enterprise Linux 8 Performing an Advanced RHEL Installation Guide unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/installation_guide/index#chap-installation-server-setup.

- haproxy – Load Balancer für API- und Anwendungsservices

Die folgende Beispieldatei zeigt eine mögliche HAProxy-Konfiguration:

```
[root@utility ~]# cat /etc/haproxy/haproxy.cfg

#-----
# Global settings
#-----

global
    log          127.0.0.1 local2
    chroot       /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     4000
    user         haproxy
    group        haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats
    # utilize system-wide crypto-policies
    ssl-default-bind-ciphers PROFILE=SYSTEM
    ssl-default-server-ciphers PROFILE=SYSTEM

#-----
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#-----

defaults
    mode          http
    log           global
    option         httplog
    option         dontlognull
    option http-server-close
    option forwardfor   except 127.0.0.0/8
    option         redispach
    retries       3
    timeout http-request 10s
    timeout queue   1m
    timeout connect 10s
    timeout client   1m
    timeout server   1m
    timeout http-keep-alive 10s
    timeout check    10s
    maxconn      3000
```

```
#-----  
# round robin balancing for RHOCOP Kubernetes API Server  
#-----  
frontend k8s_api  
    bind *:6443  
    mode tcp  
    default_backend k8s_api_backend  
backend k8s_api_backend  
    balance roundrobin  
    mode tcp  
    server bootstrap 192.168.50.9:6443 check  
    server master01 192.168.50.10:6443 check  
    server master02 192.168.50.11:6443 check  
    server master03 192.168.50.12:6443 check  
  
# -----  
# round robin balancing for RHOCOP Machine Config Server  
#-----  
frontend machine_config  
    bind *:22623  
    mode tcp  
    default_backend machine_config_backend  
backend machine_config_backend  
    balance roundrobin  
    mode tcp  
    server bootstrap 192.168.50.9:22623 check  
    server master01 192.168.50.10:22623 check  
    server master02 192.168.50.11:22623 check  
    server master03 192.168.50.12:22623 check  
  
# -----  
# round robin balancing for RHOCOP Ingress Insecure Port  
#-----  
frontend ingress_insecure  
    bind *:80  
    mode tcp  
    default_backend ingress_insecure_backend  
backend ingress_insecure_backend  
    balance roundrobin  
    mode tcp  
    server worker01 192.168.50.13:80 check  
    server worker02 192.168.50.14:80 check  
  
# -----  
# round robin balancing for RHOCOP Ingress Secure Port  
#-----  
frontend ingress_secure  
    bind *:443  
    mode tcp  
    default_backend ingress_secure_backend  
backend ingress_secure_backend  
    balance roundrobin  
    mode tcp  
    server worker01 192.168.50.13:443 check  
    server worker02 192.168.50.14:443 check
```

```
# -----  
# Exposing HAProxy Statistic Page  
# -----  
listen stats  
    bind :32700  
    stats enable  
    stats uri /  
    stats hide-version  
    stats auth admin:RedH@t322
```



Anmerkung

Weitere Informationen finden Sie im *HAProxy 2.4 Configuration Manual* unter <https://www.haproxy.org/download/2.4/doc/configuration.txt>.

- **httpd** zum Bereitstellen von Dateien

Der Apache HTTPD-Server ist standardmäßig für die Veröffentlichung des Ordners `/var/www/html/` konfiguriert. Sie können diese Konfiguration in der Datei `/etc/httpd/conf/httpd.conf` ändern.

In der Kursumgebung wurde der HTTP-Standardport für den Apache HTTPD-Server auf Port 8080 festgelegt. Dies liegt daran, dass der HAProxy-Server Port 80 für unsichere HTTP-Verbindungen zu den Ingress-Router-Pods verwendet.



Anmerkung

Weitere Informationen finden Sie im Kapitel *Setting up the Apache HTTP Webserver* des *Red Hat Enterprise Linux 8 Deploying Different Types of Servers Guide* unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/deploying_different_types_of_servers/index#setting-apache-http-server_Deploying-different-types-of-servers.



Anmerkung

Der **chrony**-Server ist standardmäßig für jede RHCOS-Installation aktiviert. Auf diese Weise sind die Cluster-Knoten so konfiguriert, dass sie dieselben Upstream-NTP-Server verwenden. Sie können die NTP-Server mithilfe einer Rechnerkonfigurationsdatei konfigurieren.

Weitere Informationen finden Sie im Leitfaden *Installing and configuring OpenShift Container Platform clusters* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing/index#installation-special-config-chrony_installing-customizing.

Erläutern der Installationskonfigurationsdatei und Erstellen der Ignition-Dateien

Die Datei `install-config.yaml` ist die Konfigurationsdatei für die Installation. Der Befehl `openshift-install` hat eine Option zur Erläuterung der Felder in der Datei `install-config.yaml`:

```
[user@demo ~]$ openshift-install explain installconfig

KIND:     InstallConfig
VERSION:  v1

RESOURCE: <object>
InstallConfig is the configuration for an OpenShift install.

FIELDS:
  additionalTrustBundle <string>
    AdditionalTrustBundle is a PEM-encoded X.509 certificate bundle that is
    added to the nodes' trusted certificate store.

  apiVersion <string>
    APIVersion defines the versioned schema of this representation of an object.
    Servers should convert recognized schemas to the latest internal value, and may
    reject unrecognized values. More info: https://git.k8s.io/community/contributors/
    devel/sig-architecture/api-conventions.md#resources

  baseDomain <string> -required-
    BaseDomain is the base domain to which the cluster should belong.

...output omitted...
```

Weitere Details zum Anhängen des Felds werden angezeigt:

```
[user@demo ~]$ openshift-install explain installconfig.baseDomain
KIND:     InstallConfig
VERSION:  v1

RESOURCE: <string>
BaseDomain is the base domain to which the cluster should belong.
```

Ein Beispiel der Datei `install-config.yaml` finden Sie im Kapitel *Installing on bare metal* der Red Hat OpenShift Container Platform 4.6. Dokumentation [https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_bare_metal/index#installation-initializing-manual_installing-bare-metal].

Nach dem Erstellen der Datei `install-config.yaml` müssen Sie die Kubernetes-Manifeste erstellen, sie bei Bedarf anpassen und schließlich die Ignition-Dateien erstellen.



Warnung

Die Datei `install-config.yaml` wird nach dem Verarbeiten entfernt, um die Kubernetes-Manifeste zu generieren. Auch wenn sie neu erstellt werden kann, empfiehlt Red Hat, eine Kopie der Datei `install-config.yaml` zu erstellen, bevor sie mit dem Befehl `openshift-install` verarbeitet wird. Wenn die Installation fehlschlägt, kann die Datei `install-config.yaml` überprüft werden, um mögliche Probleme zu debuggen.

Schließlich müssen Sie die Ignition-Dateien für die Knoten während der Installation zur Verfügung stellen und die `tftpboot`-Dateien so konfigurieren, dass sie während der RHCOS-Installation abgerufen werden.

Generieren eines SSH-Schlüssels für die Fehlerbehebung beim Installationsprogramm

Um Fehler beim Installationsprozess auf dem `bootstrap`-Server zu beheben, können Sie einen SSH-Schlüssel erstellen. Fügen Sie nach dem Generieren des SSH-Schlüsselpaares den öffentlichen SSH-Schlüssel zur Datei `install-config.yaml` hinzu. Das Installationsprogramm enthält den öffentlichen Schlüssel im `bootstrap`-Knoten, um über SSH eine Verbindung herzustellen und Fehler in den ersten Phasen der Installation zu beheben.

Dieser Ansatz ist nützlich für die Behebung von Problemen beim Abrufen der Images, der Installation von RHCOS im `bootstrap` oder `master`-Knoten oder bei Verbindungsproblemen mit den `master`-Knoten oder dem API-Server.



Literaturhinweise

- Weitere Informationen finden Sie im Kapitel *Installing on bare metal* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing_on_bare_metal/index#installing-bare-metal

► Angeleitete Übung

Konfigurieren von Netzwerkservices und Hosts für die Installation von OpenShift ohne einen Infrastrukturanbieter

In dieser Übung prüfen und konfigurieren Sie Services und bereiten die Umgebung für die Installation von OpenShift in einer bereits vorhandenen Infrastruktur vor.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Laden Sie die RHCOS-Dateien herunter.
- Erstellen Sie die Ignition-Dateien aus der Datei `install-config.yaml`.
- Überprüfen und konfigurieren Sie die erforderlichen Services.

Bevor Sie Beginnen

Stellen Sie zum Durchführen dieser Übung sicher, dass Sie das Kursskript *Planen der Installation von OpenShift ohne einen Infrastrukturanbieter* gelesen haben.

Stellen Sie außerdem sicher, dass Sie die angeleitete Übung *Erfüllen der OpenShift-Installationsvoraussetzungen* abgeschlossen haben.

Wenn Sie Änderungen an der Kursumgebung vorgenommen haben, müssen Sie den Kursraum löschen und neu erstellen, um diese Aktivität erfolgreich abschließen zu können.

Anweisungen

- 1. Überprüfen und konfigurieren Sie auf dem Server `utility` als Benutzer `root` den DNS-Service.
- 1.1. Prüfen und konfigurieren Sie (falls erforderlich) den DNS-Service, der vom named-Daemon bereitgestellt wird. Die Konfigurationsdatei ist `/etc/named.conf`, wie in der Ausgabe des systemd-Einheitenstatus gezeigt:

```
[student@workstation ~]$ ssh lab@utility
...output omitted...
[lab@utility ~]$ sudo -i
[root@utility ~]#
```

```
[root@utility ~]# systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; vendor preset:
dis>
   Active: active (running) since Thu 2020-11-19 06:13:07 EST; 3h 32min ago
     Main PID: 944 (named)
        Tasks: 5 (limit: 11345)
```

Kapitel 4 | Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

```
Memory: 60.7M
CGroup: /system.slice/named.service
└─944 /usr/sbin/named -u named -c /etc/named.conf
```

- 1.2. Öffnen Sie die Datei `/etc/named.conf`, um deren Inhalt anzuzeigen. Für diese Datei sind keine Änderungen erforderlich.

```
[root@utility ~]# cat /etc/named.conf
...output omitted...
```

- 1.3. Überprüfen Sie die Datenbankdateien für die Domain `example.com` in `/var/named`:

```
[root@utility ~]# cat /var/named/example.com.db
$TTL 1D
@ IN SOA dns.ocp4.example.com. root.example.com. (
    2019022400 ; serial
    3h          ; refresh
    15          ; retry
    1w          ; expire
    3h          ; minimum
)
IN NS dns.ocp4.example.com.
dns.ocp4      IN A 192.168.50.254
api.ocp4      IN A 192.168.50.254
api-int.ocp4   IN A 192.168.50.254
*.apps.ocp4   IN A 192.168.50.254
bootstrap.ocp4 IN A 192.168.50.9
master01.ocp4 IN A 192.168.50.10
etcd-0.ocp4   IN A 192.168.50.10
_etcd-server-ssl._tcp.ocp4 IN SRV 0 10 2380 etcd-0.ocp4
master02.ocp4 IN A 192.168.50.11
etcd-1.ocp4   IN A 192.168.50.11
_etcd-server-ssl._tcp.ocp4 IN SRV 0 10 2380 etcd-1.ocp4
master03.ocp4 IN A 192.168.50.12
etcd-2.ocp4   IN A 192.168.50.12
_etcd-server-ssl._tcp.ocp4 IN SRV 0 10 2380 etcd-2.ocp4
worker01.ocp4 IN A 192.168.50.13
worker02.ocp4 IN A 192.168.50.14
```

```
[root@utility ~]# cat /var/named/example.com.reverse.db
$TTL 1D
@ IN SOA dns.ocp4.example.com. root.example.com. (
    2019022400 ; serial
    3h          ; refresh
    15          ; retry
    1w          ; expire
    3h          ; minimum
)
IN NS dns.ocp4.example.com.
254 IN PTR api.ocp4.example.com.
254 IN PTR api-int.ocp4.example.com.
9 IN PTR bootstrap.ocp4.example.com.
10 IN PTR master01.ocp4.example.com.
```

```
11 IN PTR master02.ocp4.example.com.  
12 IN PTR master03.ocp4.example.com.  
13 IN PTR worker01.ocp4.example.com.  
14 IN PTR worker02.ocp4.example.com.
```

1.4. Führen Sie den Befehl `dig` aus, um die DNS-Auflösung zu testen:

```
[root@utility ~]# dig api.ocp4.example.com  
...output omitted...  
;api.ocp4.example.com. IN A  
  
;; ANSWER SECTION:  
api.ocp4.example.com. 86400 IN A 192.168.50.254  
  
;; AUTHORITY SECTION:  
example.com. 86400 IN NS dns.ocp4.example.com.  
  
;; ADDITIONAL SECTION:  
dns.ocp4.example.com. 86400 IN A 192.168.50.254  
  
;; Query time: 2 msec  
;; SERVER: 172.25.250.254#53(172.25.250.254)  
;; WHEN: Wed Jan 27 12:07:49 EST 2021  
;; MSG SIZE rcvd: 127  
  
[root@utility ~]# dig test.apps.ocp4.example.com  
...output omitted...  
;test.apps.ocp4.example.com. IN A  
  
;; ANSWER SECTION:  
test.apps.ocp4.example.com. 86400 IN A 192.168.50.254  
  
;; AUTHORITY SECTION:  
example.com. 86400 IN NS dns.ocp4.example.com.  
  
;; ADDITIONAL SECTION:  
dns.ocp4.example.com. 86400 IN A 192.168.50.254  
  
;; Query time: 2 msec  
;; SERVER: 172.25.250.254#53(172.25.250.254)  
;; WHEN: Wed Jan 27 12:08:41 EST 2021  
;; MSG SIZE rcvd: 133  
  
[root@utility ~]# dig master01.ocp4.example.com  
...output omitted...  
;master01.ocp4.example.com. IN A  
  
;; ANSWER SECTION:  
master01.ocp4.example.com. 86400 IN A 192.168.50.10  
  
;; AUTHORITY SECTION:  
example.com. 86400 IN NS dns.ocp4.example.com.  
  
;; ADDITIONAL SECTION:
```

```
dns.ocp4.example.com. 86400 IN A 192.168.50.254

;; Query time: 1 msec
;; SERVER: 172.25.250.254#53(172.25.250.254)
;; WHEN: Wed Jan 27 12:09:19 EST 2021
;; MSG SIZE rcvd: 132
```

► 2. Prüfen und konfigurieren Sie den DHCP-Service.

- 2.1. Prüfen und konfigurieren Sie (falls erforderlich) den DHCP-Service, der vom dhcpcd-Daemon bereitgestellt wird. Die Konfigurationsdatei ist /etc/dhcp/dhcpcd.conf, wie in der Ausgabe des systemd-Unit-Status gezeigt:

```
[root@utility ~]# systemctl status dhcpcd
● dhcpcd.service - DHCPv4 Server Daemon
  Loaded: loaded (/usr/lib/systemd/system/dhcpcd.service; enabled; vendor preset: disabled)
  Active: active (running) since Thu 2020-11-19 06:13:08 EST; 3h 39min ago
    Docs: man:dhcpcd(8)
          man:dhcpcd.conf(5)
  Main PID: 1261 (dhcpcd)
    Status: "Dispatching packets..."
      Tasks: 1 (limit: 11345)
     Memory: 11.9M
    CGroup: /system.slice/dhcpcd.service
            └─1261 /usr/sbin/dhcpcd -f -cf /etc/dhcp/dhcpcd.conf -user dhcpcd -group
dhcpcd --no-pid
```

- 2.2. Öffnen Sie die Datei /etc/dhcp/dhcpcd.conf, um deren Inhalt anzuzeigen. Diese Datei enthält die MAC-Adressen der Server in der Kursumgebung. Später verwenden Sie diese MAC-Adressen, um die PXE-Boot-Konfigurationsdateien zu erstellen. Nehmen Sie keine Änderungen an dieser Datei vor.

```
[root@utility ~]# cat /etc/dhcp/dhcpcd.conf
...output omitted...
host master01.ocp4.example.com { hardware ethernet 52:54:00:00:32:0A; fixed-
address 192.168.50.10; option host-name "master01"; }
host master02.ocp4.example.com { hardware ethernet 52:54:00:00:32:0B; fixed-
address 192.168.50.11; option host-name "master02"; }
host master03.ocp4.example.com { hardware ethernet 52:54:00:00:32:0C; fixed-
address 192.168.50.12; option host-name "master03"; }
host bootstrap.ocp4.example.com { hardware ethernet 52:54:00:00:32:09; fixed-
address 192.168.50.9; option host-name "bootstrap"; }
host worker01.ocp4.example.com { hardware ethernet 52:54:00:00:32:0D; fixed-
address 192.168.50.13; option host-name "worker01"; }
host worker02.ocp4.example.com { hardware ethernet 52:54:00:00:32:0E; fixed-
address 192.168.50.14; option host-name "worker02"; }

...output omitted...
```

► 3. Prüfen und konfigurieren Sie den HTTPD-Service.

- 3.1. Prüfen und konfigurieren Sie (falls erforderlich) den HTTP-Service, der vom `httpd`-Daemon bereitgestellt wird.

```
[root@utility ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
  Active: active (running) since Thu 2020-11-19 06:13:07 EST; 3h 44min ago
    Docs: man:httpd.service(8)
  Main PID: 957 (httpd)
    Status: "Running, listening on: port 8080"
       Tasks: 213 (limit: 11345)
      Memory: 39.4M
     CGroup: /system.slice/httpd.service
             ├─957 /usr/sbin/httpd -DFOREGROUND
             ├─966 /usr/sbin/httpd -DFOREGROUND
             ├─970 /usr/sbin/httpd -DFOREGROUND
             ├─971 /usr/sbin/httpd -DFOREGROUND
             └─974 /usr/sbin/httpd -DFOREGROUND
```

- 3.2. Die Standardkonfigurationsdatei für den `httpd`-Server ist `/etc/httpd/conf/httpd.conf`. Da Sie ihn als Dateiserver verwenden, müssen Sie das Verzeichnis suchen, in dem die Dateien gespeichert werden sollen. Führen Sie den folgenden Befehl aus, um nach diesem Verzeichnis zu suchen.

```
[root@utility ~]# grep DocumentRoot /etc/httpd/conf/httpd.conf
# DocumentRoot: The directory out of which you will serve your
DocumentRoot "/var/www/html"
...output omitted...
```

- 3.3. Sehen Sie sich die Struktur des Ordners `/var/www/html` an:

```
[root@utility ~]# tree /var/www/html/
/var/www/html/
└── openshift4
    ├── 4.6.4
    │   └── ignitions
    └── images
```

Der `httpd`-Server ist bereit, die Ignition-Dateien in `/var/www/html/openshift4/4.6.4/ignitions/` und die RHCOS-Images in `/var/www/html/openshift4/images/` zu platzieren.

- 3.4. Laden Sie die RHCOS-Images in den Ordner `/var/www/html/openshift4/images/` herunter:

```
[root@utility ~]# ocp_maj=4.6;rhcov_ver=4.6.1
[root@utility ~]# mirror=https://mirror.openshift.com/pub/openshift-v4
[root@utility ~]# baseurl=${mirror}/dependencies/rhcos/${ocp_maj}/${rhcov_ver}
[root@utility ~]# cd /var/www/html/openshift4/images/
[root@utility images]# wget \
> ${baseurl}/rhcos-${rhcov_ver}-x86_64-live-rootfs.x86_64.img
```

Kapitel 4 | Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

```
...output omitted...
```

```
2020-11-23 10:14:02 (57,0 MB/s) - "rhcos-4.6.1-x86_64-live-rootfs.x86_64.img"
 saved [819969024/819969024]
```

```
[root@utility images]# wget \
> ${baseurl}/rhcos-${rhcos_ver}-x86_64-live-kernel-x86_64
...output omitted...
```

```
2020-11-23 10:15:50 (6,26 MB/s) - "rhcos-4.6.1-x86_64-live-kernel-x86_64" saved
[8924528/8924528]
```

```
[root@utility images]# wget \
> ${baseurl}/rhcos-${rhcos_ver}-x86_64-live-initramfs.x86_64.img
...output omitted...
```

```
2020-11-23 10:16:33 (29,2 MB/s) - "rhcos-4.6.1-x86_64-live-initramfs.x86_64.img"
 saved [80239428/80239428]
```

3.5. Kehren Sie zum root-Home-Verzeichnis zurück:

```
[root@utility images]# cd ~
[root@utility ~]#
```

► 4. Prüfen und konfigurieren Sie den Load Balancer.

- 4.1. Prüfen und konfigurieren Sie (falls erforderlich) den HAProxy-Load-Balancer-Service, der vom haproxy-Daemon bereitgestellt wird. Die Konfigurationsdatei ist /etc/haproxy/haproxy.cfg, wie in der Ausgabe des systemd-Unit-Status gezeigt:

```
[root@utility ~]$ systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; vendor
  preset: disabled)
    Active: active (running) since Fri 2020-11-20 10:20:07 EST; 26min ago
      Process: 917 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -c -q (code=exited,
      Status: 0/SUCCESS)
    Main PID: 932 (haproxy)
       Tasks: 2 (limit: 11345)
     Memory: 4.7M
        CPU: 0.000 CPU(s) (idle)
       CGroup: /system.slice/haproxy.service
                 ├─932 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/
                 └─943 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/
                     haproxy.pid
```

- 4.2. Bearbeiten Sie die Datei /etc/haproxy/haproxy.cfg, um die Back-End-Server für den API-Server, Machine Config Server und Anwendungen (sicher und unsicher) hinzuzufügen.

```
[root@utility ~]# vi /etc/haproxy/haproxy.cfg

...output omitted...

# -----
# round robin balancing for RHOCUP Kubernetes API Server
# -
frontend k8s_api
    bind *:6443
    mode tcp
    default_backend k8s_api_backend
backend k8s_api_backend
    balance roundrobin
    mode tcp
    server bootstrap 192.168.50.9:6443 check
    server master01 192.168.50.10:6443 check
    server master02 192.168.50.11:6443 check
    server master03 192.168.50.12:6443 check

# -----
# round robin balancing for RHOCUP Machine Config Server
# -
frontend machine_config
    bind *:22623
    mode tcp
    default_backend machine_config_backend
backend machine_config_backend
    balance roundrobin
    mode tcp
    server bootstrap 192.168.50.9:22623 check
    server master01 192.168.50.10:22623 check
    server master02 192.168.50.11:22623 check
    server master03 192.168.50.12:22623 check

# -----
# round robin balancing for RHOCUP Ingress Insecure Port
# -
frontend ingress_insecure
    bind *:80
    mode tcp
    default_backend ingress_insecure_backend
backend ingress_insecure_backend
    balance roundrobin
    mode tcp
    server worker01 192.168.50.13:80 check
    server worker02 192.168.50.14:80 check

# -----
# round robin balancing for RHOCUP Ingress Secure Port
# -
frontend ingress_secure
    bind *:443
    mode tcp
    default_backend ingress_secure_backend
```

Kapitel 4 | Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

```
backend ingress_secure_backend
    balance roundrobin
    mode tcp
    server worker01 192.168.50.13:443 check
    server worker02 192.168.50.14:443 check

# -----
# Exposing HAProxy Statistic Page
# -----
listen stats
    bind :32700
    stats enable
    stats uri /
    stats hide-version
    stats auth admin:RedH@t322
```

Speichern Sie die Änderungen an der Datei /etc/haproxy/haproxy.cfg.

**Anmerkung**

Sie können den Inhalt Ihrer Datei /etc/haproxy/haproxy.cfg mit der Lösungsdatei in <http://classroom.example.com/materials/solutions/haproxy/haproxy.cfg> vergleichen.

Verwenden Sie den Befehl wget, um diese Datei in Ihr Arbeitsverzeichnis herunterzuladen:

```
[root@utility ~]# wget
  http://classroom.example.com/materials/solutions/haproxy/haproxy.cfg
...output omitted...
```

- 4.3. Stellen Sie sicher, dass in der Datei haproxy.cfg keine Fehler enthalten sind. Laden Sie den HAProxy-Service neu, um die Änderungen anzuwenden. Überprüfen Sie die Service-Status.

```
[root@utility ~]# haproxy -c -f /etc/haproxy/haproxy.cfg
[WARNING] 026/114452 (1646) : parsing [/etc/haproxy/haproxy.cfg:49] : 'option
httplog' not usable with frontend 'k8s_api' (needs 'mode http'). Falling back to
'option tcplog'.
[WARNING] 026/114452 (1646) : config : 'option forwardfor' ignored for frontend
'k8s_api' as it requires HTTP mode.
[WARNING] 026/114452 (1646) : config : 'option forwardfor' ignored for backend
'k8s_api_backend' as it requires HTTP mode.
[WARNING] 026/114452 (1646) : parsing [/etc/haproxy/haproxy.cfg:49] : 'option
httplog' not usable with frontend 'machine_config' (needs 'mode http'). Falling
back to 'option tcplog'.
[WARNING] 026/114452 (1646) : config : 'option forwardfor' ignored for frontend
'machine_config' as it requires HTTP mode.
[WARNING] 026/114452 (1646) : config : 'option forwardfor' ignored for backend
'machine_config_backend' as it requires HTTP mode.
[WARNING] 026/114452 (1646) : parsing [/etc/haproxy/haproxy.cfg:49] : 'option
httplog' not usable with frontend 'ingress_insecure' (needs 'mode http'). Falling
back to 'option tcplog'.
```

```
[WARNING] 026/114452 (1646) : config : 'option forwardfor' ignored for frontend  
'ingress_insecure' as it requires HTTP mode.  
[WARNING] 026/114452 (1646) : config : 'option forwardfor' ignored for backend  
'ingress_insecure_backend' as it requires HTTP mode.  
[WARNING] 026/114452 (1646) : parsing [/etc/haproxy/haproxy.cfg:49] : 'option  
httplog' not usable with frontend 'ingress_secure' (needs 'mode http'). Falling  
back to 'option tcplog'.  
[WARNING] 026/114452 (1646) : config : 'option forwardfor' ignored for frontend  
'ingress_secure' as it requires HTTP mode.  
[WARNING] 026/114452 (1646) : config : 'option forwardfor' ignored for backend  
'ingress_secure_backend' as it requires HTTP mode.  
Configuration file is valid
```



Anmerkung

Der oben aufgeführte Befehl gibt Warnmeldungen zurück, da einige HTTP-Standardoptionen nicht auf TCP-Verbindungen angewendet werden können. Sie können diese Warnungen ignorieren.

```
[root@utility ~]# systemctl reload haproxy
```

```
[root@utility ~]$ systemctl status haproxy  
● haproxy.service - HAProxy Load Balancer  
    Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; vendor  
    preset: disabled)  
    Active: active (running) since Fri 2020-11-20 10:20:07 EST; 26min ago  
      ...output omitted...
```

- 5. Prüfen und konfigurieren Sie den TFTP-Service. Nach diesem Schritt sollte die Ordnerstruktur /var/lib/tftpboot/pxelinux.cfg/ wie folgt aussehen:

```
[root@utility ~]# tree /var/lib/tftpboot/pxelinux.cfg/  
/var/lib/tftpboot/pxelinux.cfg/  
└── 01-52-54-00-00-32-09  
    ├── 01-52-54-00-00-32-0a  
    └── 01-52-54-00-00-32-0b
```

```
└── 01-52-54-00-00-32-0c
    ├── 01-52-54-00-00-32-0d
    ├── 01-52-54-00-00-32-0e
    └── 01-example
```



Anmerkung

Sie können `wget` verwenden, um die pxeboot-Dateien vom Server `classroom` herunterzuladen:

```
[root@utility ~]# sol_url=http://classroom.example.com/materials/solutions
[root@utility ~]# wget $sol_url/pxelinux.cfg/01-52-54-00-00-32-09 \
> -P /var/lib/tftpboot/pxelinux.cfg/
...output omitted...
```

- 5.1. Prüfen und konfigurieren Sie (falls erforderlich) den TFTP-Service, der vom `tftpd`-Daemon bereitgestellt wird.

```
[root@utility ~]# systemctl status tftp
● tftp.service - Tftp Server
  Loaded: loaded (/usr/lib/systemd/system/tftp.service; indirect; vendor preset: disabled)
  Active: active (running) since Mon 2020-11-23 09:37:46 EST; 21min ago
    Docs: man:in.tftpd(8)
 Main PID: 1331 (in.tftpd)
   Tasks: 1 (limit: 11345)
  Memory: 1.2M
    CGroup: /system.slice/tftp.service
           └─1331 /usr/sbin/in.tftpd -s /var/lib/tftpboot
```

- 5.2. Erstellen Sie mit den MAC-Adressen aus der Datei `/etc/dhcp/dhcpd.conf`, auf die in einem vorherigen Schritt verwiesen wurde, wie folgt sechs Dateien in `/var/lib/tftpboot/pxelinux.cfg/`.

Stellen Sie der MAC-Adresse in Kleinbuchstaben `01`-voran, und trennen Sie jedes Paar von Hexadezimalziffern mit einem Bindestrich.

- `master01 - /var/lib/tftpboot/pxelinux.cfg/01-52-54-00-00-32-0a`
- `master02 - /var/lib/tftpboot/pxelinux.cfg/01-52-54-00-00-32-0b`
- `master03 - /var/lib/tftpboot/pxelinux.cfg/01-52-54-00-00-32-0c`

Inhalt der drei Dateien:

```

default menu.c32
prompt 0
timeout 0
menu title **** OpenShift 4 MASTER PXE Boot Menu ****

label Install RHCOS 4.6.1 Master Node
kernel http://192.168.50.254:8080/openshift4/images/rhcos-4.6.1-x86_64-live-
kernel-x86_64
append ip=dhcp rd.neednet=1 coreos.inst.install_dev=vda console=tty0
console=ttyS0 coreos.inst=yes coreos.live.rootfs_url=http://192.168.50.254:8080/
openshift4/images/rhcos-4.6.1-x86_64-live-rootfs.x86_64.img
coreos.inst.ignition_url=http://192.168.50.254:8080/openshift4/4.6.4/
ignitions/master.ign initrd=http://192.168.50.254:8080/openshift4/images/
rhcos-4.6.1-x86_64-live-initramfs.x86_64.img

```



Anmerkung

Stellen Sie sicher, dass nach `label Install RHCOS 4.6.1 Master Node` nur zwei weitere Zeilen in der Datei enthalten sind. Die vorletzte Zeile beginnt mit `kernel`, und die letzte Zeile der Datei beginnt mit `append`. Zwischen `http://192.168.50.254:8080/openshift4/images/rhcos-4.6.1-x86_64-live-kernel-x86_64` und `append` darf sich nur ein Zeilenumbruch befinden.

- `bootstrap - /var/lib/tftpboot/pxelinux.cfg/01-52-54-00-00-32-09`

Inhalt der Datei:

```

default menu.c32
prompt 0
timeout 0
menu title **** OpenShift 4 BOOTSTRAP PXE Boot Menu ****

label Install RHCOS 4.6.1 Bootstrap Node
kernel http://192.168.50.254:8080/openshift4/images/rhcos-4.6.1-x86_64-live-
kernel-x86_64
append ip=dhcp rd.neednet=1 coreos.inst.install_dev=vda console=tty0
console=ttyS0 coreos.inst=yes coreos.live.rootfs_url=http://192.168.50.254:8080/
openshift4/images/rhcos-4.6.1-x86_64-live-rootfs.x86_64.img
coreos.inst.ignition_url=http://192.168.50.254:8080/openshift4/4.6.4/
ignitions/bootstrap.ign initrd=http://192.168.50.254:8080/openshift4/images/
rhcos-4.6.1-x86_64-live-initramfs.x86_64.img

```



Anmerkung

Stellen Sie sicher, dass nach `label Install RHCOS 4.6.1 Bootstrap Node` nur zwei weitere Zeilen in der Datei enthalten sind. Die vorletzte Zeile beginnt mit `kernel`, und die letzte Zeile der Datei beginnt mit `append`. Zwischen `http://192.168.50.254:8080/openshift4/images/rhcos-4.6.1-x86_64-live-kernel-x86_64` und `append` darf sich nur ein Zeilenumbruch befinden.

Kapitel 4 | Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

- worker01 - /var/lib/tftpboot/pixelinux.cfg/01-52-54-00-00-32-0d
- worker02 - /var/lib/tftpboot/pixelinux.cfg/01-52-54-00-00-32-0e

Inhalt der beiden Dateien:

```
default menu.c32
prompt 0
timeout 0
menu title **** OpenShift 4 WORKER PXE Boot Menu ****

label Install RHCOS 4.6.1 Worker Node
kernel http://192.168.50.254:8080/openshift4/images/rhcos-4.6.1-x86_64-live-
kernel-x86_64
append ip=dhcp rd.neednet=1 coreos.inst.install_dev=vda console=tty0
console=ttyS0 coreos.inst=yes coreos.live.rootfs_url=http://192.168.50.254:8080/
openshift4/images/rhcos-4.6.1-x86_64-live-rootfs.x86_64.img
coreos.inst.ignition_url=http://192.168.50.254:8080/openshift4/4.6.4/
ignitions/worker.ign initrd=http://192.168.50.254:8080/openshift4/images/
rhcos-4.6.1-x86_64-live-initramfs.x86_64.img
```

**Anmerkung**

Stellen Sie sicher, dass nach `label Install RHCOS 4.6.1 Worker Node` nur zwei weitere Zeilen in der Datei enthalten sind. Die vorletzte Zeile beginnt mit `kernel`, und die letzte Zeile der Datei beginnt mit `append`. Zwischen `http://192.168.50.254:8080/openshift4/images/rhcos-4.6.1-x86_64-live-kernel-x86_64` und `append` darf sich nur ein Zeilenumbruch befinden.

- 5.3. Überprüfen Sie die Ordnerstruktur `/var/lib/tftpboot/pixelinux.cfg/`:

Jetzt kann der TFTP-Server die PXE-Boot-Dateien pro MAC-Adresse bereitstellen.
Die Server rufen diese Dateien beim Neustart ab.

**Anmerkung**

Sie können den Inhalt Ihrer Dateien in `/var/lib/tftpboot/pixelinux.cfg/` mit den Lösungsdateien in `http://classroom.example.com/materials/solutions/pixelinux.cfg/` vergleichen.

**Warnung**

Versuchen Sie noch nicht, RHCOS auf den Servern zu installieren. Sie müssen zuerst die Ignition-Dateien erstellen, da die Installation ansonsten fehlschlägt.

- 6. Erstellen Sie als Benutzer `lab` die Ignition-Dateien, und legen Sie sie auf dem Dateiserver ab.



Anmerkung

Führen Sie die übrigen Schritte in dieser angeleiteten Übung als Benutzer `lab` aus.

- 6.1. Erstellen Sie einen Ordner mit dem Namen `ocp4upi`, und wechseln Sie zu diesem Ordner:

```
[root@utility ~]# exit
logout
[lab@utility ~]$ mkdir ocp4upi && cd ocp4upi
[lab@utility ocp4upi]$
```

- 6.2. Erstellen Sie die Datei `install-config.yaml`.

Verwenden Sie zum Beschleunigen der Installation die lokale Registry der Region, in der Ihr Cluster bereitgestellt ist, und die von `quay.io` gespiegelten OpenShift-Release-Images enthält.



Anmerkung

Standardmäßig wird die lokale Registry der Region `northamerica (nexus-registry-int.apps.tools-na.prod.nextcle.com)` in allen Lektionen und angeleiteten Übungen dieses Kurses verwendet.

Stellen Sie sicher, dass Sie in allen angeleiteten Übungen den FQDN der lokalen Registry der Region Ihrer Kursumgebung verwenden. Die Verwendung einer lokalen Registry aus anderen Regionen führt dazu, dass die OpenShift-Installation fehlschlägt.

Navigieren Sie auf dem Rechner **workstation** mit Firefox zur Registerkarte **Lab Environment** auf der Webseite des "Red Hat OpenShift Installation Lab"-Kurses, um zu ermitteln, in welcher Region Ihr Cluster bereitgestellt wird.

Klicken Sie dort auf **Information**, um Informationen zur Kursumgebung anzuzeigen. Die Region der Kursumgebung wird im Feld **Published region** angezeigt.

Table of Contents Course Lab Environment ★ ?

▶ Lab Controls

Click **CREATE** to build all of the virtual machines needed for the classroom lab environment. This may take several minutes to complete. Once created the environment can then be stopped and restarted to pause your experience.

If you **DELETE** your lab, you will remove all of the **virtual machines in your classroom and lose all of your progress**.

Lab Information

- Project id: 2ebdcefb42a144ada7e8c1c3fce8b56
- Project name: ole-3030f6f7-9253-47ed-9ea1-73c0aa7e62c9
- Project state: stopped
- Lab definition id: do322ea-4.6
- Published region: northamerica**
- openstack region: us-east-1

Actions

DELETE	START	?
bastion	stopped	
bootstrap	stopped	
classroom	stopped	

ACTION - OPEN CONSOLE

Zuvor haben Sie in diesem Kurs die Datei `pull-secret-oneLine.json` mithilfe der lokalen Registry in Ihrer Region generiert und auf den Server `utility` kopiert.

Kapitel 4 | Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

```
[lab@utility ocp4upi]$ cat ~/pull-secret-oneline.json
>{"auths":{"cloud.openshift.com":
>{"auth":"UxUUj...UUjYw==","email":"student@redhat.com"}, "nexus-
registry-int.apps.tools-na.prod.nextcle.com":
>{"auth":"cmVnd...STTM=","email":"nobody@example.com"}, "registry.connect.redhat.com":
>{"auth":"CvmsW...qKLJN=","email":"student@redhat.com"}, "registry.redhat.io":
>{"auth":"CvmsW...qKLJN=","email":"student@redhat.com"}}}
```

Sie müssen das Pull Secret in den Abschnitt `pullSecret` der Datei `install-config.yaml` einschließen.

Schließen Sie außerdem den öffentlichen SSH-Schlüssel in `/home/lab/.ssh/ocp4upi.pub` ein.

Schließen Sie zum Schluss auch die Konfiguration `imageContentSources` ein. Diese Konfiguration ermöglicht die Verwendung der lokalen Mirror-Registry während der Installation und hängt von der Region ab, in der Ihr Cluster bereitgestellt wird.

In der folgenden Tabelle wird der FQDN der lokalen Registry für die einzelnen Regionen aufgeführt.

FQDN der lokalen Registry nach Kursumgebungsregion

Region	FQDN der lokalen Registry
Nordamerika	nexus-registry-int.apps.tools-na.prod.nextcle.com
EMEA	nexus-registry-int.apps.tools-eu.prod.nextcle.com
APAC	nexus-registry-int.apps.tools-ap.prod.nextcle.com

```
[lab@utility ocp4upi]$ vi install-config.yaml
```

```
apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 2
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: ocp4
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
```

```

platform:
  none: {}
fips: false
pullSecret: |
  <CHANGE_ME_KEEPING_THE_INDENTATION_LEVEL>
sshKey: |
  <CHANGE_ME_KEEPING_THE_INDENTATION_LEVEL>
imageContentSources:
- mirrors:
  - <LOCAL_REGISTRY_FQDN>/openshift/ocp4
    source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <LOCAL_REGISTRY_FQDN>/openshift/ocp4
    source: quay.io/openshift-release-dev/ocp-v4.0-art-dev

```

Speichern Sie die Datei, und schließen Sie den Editor.



Anmerkung

Sie können die Basisdatei `install-config.yaml` von <http://classroom.example.com/materials/solutions/ocp4upi/install-config.yaml> herunterladen.

Sie müssen diese Datei trotzdem mit den richtigen Informationen für `pullSecret`, `sshKey` und `imageContentSources` vervollständigen.

Verwenden Sie den Befehl `wget`, um diese Datei in Ihr Arbeitsverzeichnis herunterzuladen:

```
[root@utility ~]# wget \
> http://classroom.example.com/materials/solutions/ocp4upi/install-config.yaml
...output omitted...
```

- 6.3. Überschreiben Sie das Image der OpenShift-Installationsversion, um das in der lokalen Registry gespeicherte Image zu verwenden. Sie müssen den FQDN der lokalen Registry der Region verwenden, in der Ihr Cluster bereitgestellt wird. Wenn beispielsweise ein Cluster in der Region `northamerica` bereitgestellt wird, verwenden Sie den FQDN `nexus-registry-int.apps.tools-na.prod.nextcle.com`.

```
[lab@utility ocp4upi]$ reg="nexus-registry-int.apps.tools-na.prod.nextcle.com"
[lab@utility ocp4upi]$ releaseimg="/openshift/ocp4:4.6.4-x86_64"
[lab@utility ocp4upi]$ export \
> OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE=$reg$releaseimg
```

Verifizieren Sie den vollständigen URL-Pfad zum Release-Image wie folgt:

```
[lab@utility ~]$ echo $OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE
nexus-registry-int.apps.tools-na.prod.nextcle.com/openshift/ocp4:4.6.4-x86_64
```

- 6.4. Speichern Sie eine Kopie der Datei `install-config.yaml` im Ordner `home` des Benutzers `lab`.

Kapitel 4 | Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

```
[lab@utility ocp4upi]$ cp install-config.yaml /home/lab/
```

- 6.5. Generieren Sie die Installationsmanifeste mit dem Tool `openshift-install`. Die Datei `install-config.yaml` wird in den Manifestdateien verarbeitet.

```
[lab@utility ocp4upi]$ openshift-install create manifests --dir=.
INFO Consuming Install Config from target directory
INFO Manifests created in: manifests and openshift
```

- 6.6. Überprüfen Sie die Struktur der Ordner `manifests` und `openshift`. Diese Ordner enthalten `yaml`-Konfigurationsdateien, mit denen Sie die RHCOS-Ignition-Dateien erstellen können.

```
[lab@utility ocp4upi]$ tree .
.
└── manifests
    ├── 04-openshift-machine-config-operator.yaml
    ├── cluster-config.yaml
    ├── cluster-dns-02-config.yml
    ├── cluster-infrastructure-02-config.yml
    ├── cluster-ingress-02-config.yml
    ├── cluster-network-01-crd.yml
    ├── cluster-network-02-config.yml
    ├── cluster-proxy-01-config.yaml
    ├── cluster-scheduler-02-config.yml
    ├── cvo-overrides.yaml
    ├── etcd-ca-bundle-configmap.yaml
    ├── etcd-client-secret.yaml
    ├── etcd-metric-client-secret.yaml
    ├── etcd-metric-serving-ca-configmap.yaml
    ├── etcd-metric-signer-secret.yaml
    ├── etcd-namespace.yaml
    ├── etcd-service.yaml
    ├── etcd-serving-ca-configmap.yaml
    ├── etcd-signer-secret.yaml
    └── kube-cloud-config.yaml
        └── kube-system-configmap-root-ca.yaml
    └── machine-config-server-tls-secret.yaml
    └── openshift-config-secret-pull-secret.yaml
.
└── openshift
    ├── 99_kubeadmin-password-secret.yaml
    ├── 99_openshift-cluster-api_master-user-data-secret.yaml
    ├── 99_openshift-cluster-api_worker-user-data-secret.yaml
    ├── 99_openshift-machineconfig_99-master-ssh.yaml
    ├── 99_openshift-machineconfig_99-worker-ssh.yaml
    └── openshift-install-manifests.yaml
```

- 6.7. Erstellen Sie die Ignition-Dateien.

```
[lab@utility ocp4upi]$ openshift-install create ignition-configs --dir=.
WARNING Found override for release image. Please be warned, this is not advised
INFO Consuming Worker Machines from target directory
INFO Consuming Openshift Manifests from target directory
INFO Consuming Master Machines from target directory
INFO Consuming OpenShift Install (Manifests) from target directory
INFO Consuming Common Manifests from target directory
INFO Ignition-Configs created in: . and auth
```

6.8. Überprüfen Sie die neue Ordnerstruktur.

```
[lab@utility ocp4upi]$ tree .
.
└── auth
    ├── kubeadmin-password
    └── kubeconfig
├── bootstrap.ign
├── master.ign
├── metadata.json
└── worker.ign
```

6.9. Kopieren Sie die Ignition-Dateien auf den `httpd`-Server DocumentRoot, und machen Sie sie für jeden Benutzer lesbar.

```
[lab@utility ocp4upi]$ sudo cp *.ign /var/www/html/openshift4/4.6.4/ignitions/
[lab@utility ocp4upi]$ sudo chmod +r \
> /var/www/html/openshift4/4.6.4/ignitions/*.ign
```

► 7. Kehren Sie zum `/home/lab/-`Ordner zurück.

```
[lab@utility ocp4upi]$ cd ~
[lab@utility ~]$
```

Beenden

Nehmen Sie erst in der nächsten angeleiteten Übung weitere Änderungen an der Übungsumgebung vor. Sie verwenden die Dateien in späteren angeleiteten Übungen.

Hiermit ist die angeleitete Übung beendet.

► Quiz

Kapitelwiederholung: Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

► 1. Welche der folgenden Services sind für die Installation von OpenShift auf einer bereits vorhandenen Infrastruktur NICHT erforderlich? (Wählen Sie zwei Antworten aus.)

- a. Ein PXE-Server
- b. Ein DNS-Server
- c. Ein DHCP-Server
- d. Ein Load Balancer für den API-Server

► 2. Welcher der folgenden Aussagen über den bootstrap-Knoten ist richtig?

- a. Der Administrator kann nach Abschluss der Installation über SSH eine Verbindung mit dem bootstrap-Knoten herstellen, indem er den Benutzer `core` verwendet.
- b. Der Administrator kann der Datei `install-config.yaml` einen öffentlichen SSH-Schlüssel hinzufügen, um eine Verbindung zum bootstrap-Knoten herzustellen und den Installationsprozess zu debuggen.
- c. Der Administrator kann Benutzer im bootstrap-Knoten mit der Datei `install-config.yaml` erstellen.
- d. Der Administrator muss der Datei `.ssh/authorized_keys` auf dem bootstrap-Server den öffentlichen SSH-Schlüssel hinzufügen, nachdem die Installation begonnen hat.

► 3. Welche der folgenden Aussagen über die Ignition-Dateien ist richtig?

- a. Die Ignition-Dateien werden mithilfe der Datei `install-config.yaml` in die verschiedenen Hosts injiziert.
- b. Red Hat empfiehlt, die Ignition-Dateien manuell aus den Beispielen in der offiziellen Dokumentation zu erstellen, um Fehler zu vermeiden.
- c. Die Ignition-Dateien werden mit dem Befehl `openshift-install` aus der Datei `install-config.yaml` erstellt.
- d. Die Ignition-Dateien werden mit dem Befehl `oc` aus der Datei `install-config.yaml` erstellt.

► 4. Welche zwei der folgenden Ressourcen sind erforderlich, um die Installation von RHCOS abzuschließen? (Wählen Sie zwei Antworten aus.)

- a. Die Ignition-Dateien
- b. Die Kickstart-Dateien
- c. Das `oc`-CLI-Tool
- d. Die RHCOS-Images

► Lösung

Kapitelwiederholung: Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

► 1. Welche der folgenden Services sind für die Installation von OpenShift auf einer bereits vorhandenen Infrastruktur NICHT erforderlich? (Wählen Sie zwei Antworten aus.)

- a. Ein PXE-Server
- b. Ein DNS-Server
- c. Ein DHCP-Server
- d. Ein Load Balancer für den API-Server

► 2. Welcher der folgenden Aussagen über den bootstrap-Knoten ist richtig?

- a. Der Administrator kann nach Abschluss der Installation über SSH eine Verbindung mit dem bootstrap-Knoten herstellen, indem er den Benutzer core verwendet.
- b. Der Administrator kann der Datei `install-config.yaml` einen öffentlichen SSH-Schlüssel hinzufügen, um eine Verbindung zum bootstrap-Knoten herzustellen und den Installationsprozess zu debuggen.
- c. Der Administrator kann Benutzer im bootstrap-Knoten mit der Datei `install-config.yaml` erstellen.
- d. Der Administrator muss der Datei `.ssh/authorized_keys` auf dem bootstrap-Server den öffentlichen SSH-Schlüssel hinzufügen, nachdem die Installation begonnen hat.

► 3. Welche der folgenden Aussagen über die Ignition-Dateien ist richtig?

- a. Die Ignition-Dateien werden mithilfe der Datei `install-config.yaml` in die verschiedenen Hosts injiziert.
- b. Red Hat empfiehlt, die Ignition-Dateien manuell aus den Beispielen in der offiziellen Dokumentation zu erstellen, um Fehler zu vermeiden.
- c. Die Ignition-Dateien werden mit dem Befehl `openshift-install` aus der Datei `install-config.yaml` erstellt.
- d. Die Ignition-Dateien werden mit dem Befehl `oc` aus der Datei `install-config.yaml` erstellt.

► 4. Welche zwei der folgenden Ressourcen sind erforderlich, um die Installation von RHCOS abzuschließen? (Wählen Sie zwei Antworten aus.)

- a. Die Ignition-Dateien
- b. Die Kickstart-Dateien
- c. Das `oc`-CLI-Tool
- d. Die RHCOS-Images

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Die Netzwerkservices, die für die Installation von OpenShift ohne Integration mit einem bestimmten Anbieter erforderlich sind.
- Die generische Architektur, die für die Installation von OpenShift ohne einen bestimmten Anbieter erforderlich ist.
- Überprüfen und Konfigurieren der Netzwerkservices für die Installation und Generieren des SSH-Schlüssels zur Fehlerbehebung.

Kapitel 5

Installieren von OpenShift ohne Infrastrukturanbieter

Ziel

Bereitstellen von OpenShift-Clustern ohne Integration in die zugrunde liegende Infrastruktur.

Ziele

Installieren Sie OpenShift, indem Sie den Bootstrap-Rechner und die Cluster-Knoten starten und den Fortschritt des Installationsvorgangs überwachen.

Abschnitte

- Durchführen der Installation von OpenShift ohne Infrastrukturanbieter (und angeleitete Übung)
- Kapitelwiederholung: Installieren von OpenShift ohne einen Infrastrukturanbieter (Test)

Durchführen der Installation von OpenShift ohne Infrastrukturanbieter

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, OpenShift zu installieren, indem Sie den Bootstrap-Rechner und die Cluster-Knoten starten und den Fortschritt des Installationsvorgangs überwachen.

Vorbereiten der Cluster-Rechner für die Installation

Für jeden Infrastrukturknoten, einschließlich Bootstrap, Control Plane und Computing-Rechner, muss vor der Bereitstellung des OpenShift-Clusters ein Betriebssystem installiert werden. Ohne einen Infrastrukturanbieter erfordern die Knoten in Abhängigkeit der jeweiligen Umgebung und des Ansatzes für diese Aufgabe möglicherweise eine manuelle Installation. Konfigurieren Sie vorzugsweise einen PXE-Server, um die Installation für die einzelnen Infrastrukturknoten durchzuführen. Für diesen Installationsansatz müssen Images verwendet werden, die über einen Dateiserver verfügbar sind und auf die die Infrastrukturknoten über das Netzwerk zugreifen können. Generieren Sie während dieser Installationsvorgänge einen allgemeinen SSH-Schlüssel für den Remote-Zugriff auf diese Infrastrukturknoten, und stellen Sie ihn bereit.

Um die PXE-Installationen zu initiieren, booten Sie im Netzwerk die einzelnen Infrastrukturknoten, und befolgen Sie die Eingabeaufforderungen im PXE-Menü. In den meisten Szenarien erfordert diese Installation während des Prozesses keinen Eingriff und wird schnell mit einem Begrenzungsfaktor für die Netzwerklatenz zum und von den Dateiserver- und PXE-Systemen ausgeführt.

Überprüfen Sie nach Abschluss der Installation die ordnungsgemäße Installation, indem Sie den Remote-Zugriff mithilfe des SSH-Schlüssels testen, um sicherzustellen, dass jeder Knoten ordnungsgemäß vorbereitet ist.

```
[user@demo ~]$ ssh -i ~/.ssh/sample_key.pub core@<HOSTNAME>
```

Nach Abschluss der Installation und der Validierung des Betriebssystems sind die Rechner für eine OpenShift-Cluster-Bereitstellung bereit.

Initiiieren der Installation und Überwachen des Fortschritts

Wenn alle Services und Knoten vorbereitet sind, können Sie den OpenShift-Cluster bereitstellen. Verwenden Sie nach dem Starten der OpenShift-Installation den Befehl `openshift-install`, um weitere Einblicke in den Prozess zu erhalten:

```
[user@demo ~]$ openshift-install wait-for bootstrap-complete
```

Dieser Befehl liefert die Ausgabe des Installationsprozesses, nachdem die Bootstrap-Phase und der Übergang zu den Knoten der Cluster-Control Plane abgeschlossen sind.

Die Überwachung der Bereitstellung dient zu Informationszwecken und unterstützt Sie dabei, mögliche Probleme zu identifizieren oder die erfolgreichen Übergänge in den einzelnen Phasen

des Vorgangs zu sehen. Das Herstellen einer Verbindung zum Bootstrap-Knoten zur Überwachung des Bootstrap-Servicevorgangs unterstützt Sie auch dabei, Probleme zu erkennen, die während der Installation aufgetreten sind. Überprüfen Sie die Services anhand einer SSH-Verbindung und des folgenden Befehls:

```
sh-4.2# journalctl -b -f -u release-image.service -u bootkube.service
```

Darüber hinaus erhalten Sie, wenn Sie die Protokolle erfassen und überprüfen, im Falle einer fehlgeschlagenen Installation weitere Informationen zu möglichen Maßnahmen zur Problembehebung. Nachdem Sie die entsprechenden Korrekturmaßnahmen für die dem Fehler zugrunde liegende Ursache ergriffen haben, können Sie die Installation wiederholen.

Authentifizieren mit dem neuen Cluster

Nach Abschluss der Installation enthalten die Meldungen in der endgültigen Ausgabe Informationen, die den Dateipfad der Anmeldedaten und den Link zur Web Console für den Zugriff auf den Cluster detailliert beschreiben, wie nachfolgend dargestellt:

```
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
  KUBECONFIG=/home/ocp_install_directory/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
  console.apps.ocp4.example.com
INFO Login to the console with user: "kubeadm", and password: "XXXXXX-XXXXX-
  XXXXX-XXXX"
DEBUG Time elapsed per stage:
DEBUG Cluster Operators: 27m11s
INFO Time elapsed: 27m11s
```

Während dieser Phase stellen die Knoten der Infrastruktur-Control Plane den Rest der Clusterkomponenten bereit. Der Zugriff ist jedoch auf verschiedene Arten möglich:

- Die Datei `kubeadm-password` enthält Anmeldedaten für den Benutzer `kubeadm`.
- Die Datei `kubeconfig` enthält Authentifizierungsinformationen, auf die mit dem Befehl `oc` für CLI-Interaktionen verwiesen werden kann.

Durch das Testen dieser Interaktion können die Anmeldedaten und die Konfiguration überprüft werden, was eine Überwachung und Überprüfung des restlichen Installationsprozesses ermöglicht.

Verwenden Sie eine der folgenden Methoden, um sich beim Cluster zu authentifizieren:

- Geben Sie die `kubeadm`-Anmeldedaten auf der Anmeldeseite der Web Console an.
- Geben Sie einen Pfad zu einer Authentifizierungsdatei an, indem Sie die folgende Option des Befehls `oc login` verwenden: `--kubeconfig=/path/to/kubconfig`.
- Verwenden Sie den Befehl `oc`, und geben Sie die `username-`, `password-` und `server`-Argumente über die interaktiven Eingabeaufforderungen an.
- Konfigurieren Sie eine `KUBECONFIG`-Umgebungsvariable, die in den Pfad für die `kubeconfig`-Datei aufgelöst wird und direkte `oc`-Interaktionen ermöglicht.

```
[user@demo ~]$ export KUBECONFIG=~/ocp_install_dir/auth/kubeconfig
```

Der Zugriff auf den Cluster mit einem dieser Ansätze ermöglicht die Überprüfung sowie die Cluster-Verwaltung bei der restlichen Installation.

Interaktion mit dem Cluster während der Installation

Während dieses Teils der Installation ermöglicht die Verbindung zu den Knoten der Control Plane im Cluster mit `oc debug` die Überwachung beim Erstellen der Container für die verschiedenen Cluster-Services. Wenn Sie den Status von Clustern, Operatoren, Pods und Knoten über einfache `oc get`-Interaktionen beobachten, erhalten Sie Einblicke in den Installationsfortschritt. Mit dem folgenden „watch“-Befehl können Sie alle diese Informationen erfassen und eine Überwachung aller Komponenten während der Installation ermöglichen:

```
[user@demo ~]$ watch 'oc get clusterversion; oc get clusteroperators; oc get \
> pods --all-namespaces | grep -v -E "Running|Completed"; \
> oc get csr; oc get nodes'
```

Darüber hinaus können Sie mit dem folgenden Befehl Cluster-Ereignisse überwachen, um weitere Details zum Installationsfortschritt zu erhalten:

```
[user@demo ~]$ oc get events -A -w
```

Abschluss der OpenShift-Installation

Während die Installation den Rest der für den Abschluss der Bereitstellung erforderlichen Aufgaben durchführt, ist die Überwachung der Cluster-Installation weiterhin hilfreich, um die erfolgreiche Bereitstellung zu gewährleisten.

Die CSRs für die einzelnen Knoten im Cluster müssen genehmigt werden, bevor die Installation abgeschlossen werden kann. Zeigen Sie mit dem folgenden Befehls alle ausstehenden Anforderungen an:

```
[user@demo ~]$ oc get csr
```

Genehmigen Sie die ausstehenden Anforderungen, und lassen Sie zu, dass die Installation mit dem folgenden Befehl fortfahren kann:

```
[user@demo ~]$ oc adm certificate approve <CSR>
```

Nach der initialen Genehmigung der ausstehenden CSRs werden weitere CSRs generiert und erfordern zusätzliche Genehmigungen, bevor die Cluster-Installation ordnungsgemäß abgeschlossen werden kann. Nehmen Sie nach einigen Minuten eine Überprüfung auf CSRs vor, und genehmigen Sie die ausstehenden Anforderungen. Diese Anforderungen bilden während dieser Installationsphase eine Warteschlange, die ohne diese Genehmigungen nicht abgeschlossen werden kann. Ausstehende CSR-Anforderungen, die auf Genehmigung warten, können die Cluster-Leistung und -Funktionalität beeinträchtigen. Daher ist es unerlässlich, die von der Installation generierten Anforderungen weiterhin abzurufen und zu genehmigen.

Während sich die Installation dem Abschluss nähert, verwenden Sie den folgenden Befehl, um eine ausführliche Ausgabe zum Status zu erhalten:

```
[user@demo ~]$ openshift-install --dir=ocp_install_directory wait-for
install-complete --log-level=debug
```

Nach Abschluss der Installation werden die Anmelde Daten und ein Link zur Web Console angezeigt, und die OpenShift Container Platform ist verfügbar. Führen Sie in dieser Phase Validierungsprüfungen durch, und testen Sie Workload-Bereitstellungen, um zu gewährleisten, dass der Cluster ordnungsgemäß funktioniert.



Literaturhinweise

- Weitere Informationen zur Überprüfung der Protokolle bei einem Fehler während der Installation finden Sie im Kapitel *Troubleshooting installation issues* der Dokumentation zu Red Hat OpenShift Container Platform 4.6 unter <https://docs.openshift.com/container-platform/4.6/installing/installing-troubleshooting.html>

► Angeleitete Übung

Durchführen der Installation von OpenShift ohne Infrastruktur Anbieter

In dieser Übung führen Sie eine Installation von OpenShift ohne Integration mit einem Infrastruktur Anbieter durch.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Führen Sie das Tool `openshift-install` aus, um eine manuelle Installation von OpenShift zu initiieren.
- Überwachung verschiedener Aspekte der Bereitstellung während einer OpenShift-Installation ohne Infrastruktur Anbieter

Bevor Sie Beginnen

Um diese Übung durchzuführen, müssen Sie alle vorherigen Übungen in diesem Kurs abgeschlossen haben.

Die Kursumgebung ist abgesehen von den Kursübungen unverändert.

Wenn Sie zusätzliche Änderungen an der Umgebung vorgenommen haben, erstellen Sie eine neue Kursumgebung, bevor Sie beginnen. Sie müssen die Schritte aus den angeleiteten Übungen in den vorherigen Kapiteln wiederholen, um sich auf diese Übung vorzubereiten.

Anweisungen

Diese angeleitete Übung basiert auf der konfigurierten Umgebung aus dem vorherigen Kapitel. Die Images sind für die Installation der Infrastrukturen per PXE-Boot konfiguriert. Die Installation des Betriebssystems auf jedem Knoten ist der erste Schritt, bevor Sie mit der Bereitstellung des Clusters fortfahren.

- ▶ 1. Installieren Sie den Red Hat Enterprise Linux CoreOS (RHCOS) `bootstrap`-Rechner mit der konfigurierten PXE-Startumgebung.
 - 1.1. Starten Sie den `bootstrap`-Rechner über die Konsole des Red Hat Learning-Portals neu, indem Sie in der oberen rechten Ecke auf **Ctrl Alt Del** klicken.
 - 1.2. Warten Sie, bis der Neustart abgeschlossen ist und der Rechner mit dem PXE-Menü bootet.
 - 1.3. Drücken Sie im PXE-Menü die **Eingabetaste**, um mit der Installation des Red Hat Enterprise Linux CoreOS auf dem Rechner fortfuzufahren.
- ▶ 2. Führen Sie einen PXE-basierten Installationsprozess für jeden `master01`-, `master02`-, `master03`-, `worker01`- und `worker02` -Knoten durch, indem Sie dem im vorherigen Schritt beschriebenen Neustartprozess folgen.

- 2.1. Starten Sie die einzelnen aufgeführten Rechner über die Konsole des Red Hat Learning-Portals neu, indem Sie in der oberen rechten Ecke auf **Ctrl Alt Del** klicken.
- 2.2. Warten Sie, bis der Neustart abgeschlossen ist und der Rechner mit dem PXE-Menü bootet.
- 2.3. Drücken Sie im PXE-Menü die **Eingabetaste**, um mit der Installation des Red Hat Enterprise Linux CoreOS auf dem Rechner fortzufahren.



Anmerkung

Während dieses Prozesses können Fehler bei den **master**-Knoten auftreten, da sie versuchen, ihre Ignition-Dateien über die Kubernetes-API abzurufen. Der **bootstrap**-Knoten muss zunächst die Installation abschließen und diese Dateien für die anderen Knoten bereitstellen.

```
[ 61.852424] ignition[726]: GET error: Get "https://api-int.ocp4.example.com:2623/config/master": EOF"
```

Darüber hinaus können während dieses Prozesses auch ähnliche Fehler bei den **worker**-Knoten auftreten, wenn sie versuchen, ihre Ignition-Dateien von der Produktions-Control Plane abzurufen.

```
[ 61.852424] ignition[726]: GET error: Get "https://api-int.ocp4.example.com:2623/config/master": EOF"
```

- 2.4. Drücken Sie die **Eingabetaste**, um mit der Installation des Red Hat Enterprise Linux CoreOS auf dem Rechner fortzufahren.



Anmerkung

Wenn der Installationsprozess nicht im PXE-Menü gestartet werden kann und keine Fehlermeldung ausgibt, überprüfen Sie die **httpd**-Protokolle in `/var/log/httpd/access_log`. In der Regel treten HTTP 403-Fehler (nicht zulässig) auf, wenn die Dateiberechtigungen nicht richtig sind, sowie HTTP 404-Fehler (nicht gefunden), wenn die RHCOS-Dateinamen in den pxeboot-Dateien nicht korrekt sind.



Anmerkung

Wenn die RHCOS-Installation nach dem Start aus irgendeinem Grund fehlschlägt, klicken Sie auf der Registerkarte **Lab Environment** im Red Hat Learning- Portal im Menü **ACTION** auf **Reset**, um den gewünschten virtuellen Rechner vollständig zurückzusetzen.

Nach Abschluss dieser Installationen sind die Knoten ordnungsgemäß vorbereitet und für die Bereitstellung des OpenShift-Clusters bereit.

► 3. Erstellen Sie den OpenShift-Cluster

- 3.1. Melden Sie sich vom Rechner **workstation** als Benutzer **lab** beim Rechner **utility** an.

```
[student@workstation ~]$ ssh lab@utility  
...output omitted...
```

- 3.2. Initiiieren Sie die Installation, und fragen Sie mit dem folgenden Befehl vom Bootstrap-Rechner Statusmeldungen bezüglich des Bereitstellungsfortschritts ab:

```
[lab@utility ~]$ openshift-install --dir=./ocp4upi wait-for bootstrap-complete

INFO Waiting up to 20m0s for the Kubernetes API at https://
api.ocp4.example.com:6443...
INFO API v1.19.0+9f84db3 up
INFO Waiting up to 30m0s for bootstrapping to complete...
E1203 02:31:28.643262    2817 reflector.go:307] k8s.io/client-go/
tools/watch/informerwatcher.go:146: Failed to watch *v1.ConfigMap:
Get "https://api.ocp4.example.com:6443/api/v1/namespaces/kube-
system/configmaps?allowWatchBookmarks=true&fieldSelector=metadata.name
%3Dbootstrap&resourceVersion=1&timeoutSeconds=401&watch=true": net/http: TLS
handshake timeout
E1203 02:38:49.170288    2817 reflector.go:307] k8s.io/client-go/
tools/watch/informerwatcher.go:146: Failed to watch *v1.ConfigMap:
Get "https://api.ocp4.example.com:6443/api/v1/namespaces/kube-
system/configmaps?allowWatchBookmarks=true&fieldSelector=metadata.name
%3Dbootstrap&resourceVersion=3331&timeoutSeconds=307&watch=true": net/http: TLS
handshake timeout
I1203 02:39:00.172528    2817 trace.go:116] Trace[2001942447]: "Reflector
ListAndWatch" name:k8s.io/client-go/tools/watch/informerwatcher.go:146
(started: 2020-12-03 02:38:50.170439408 -0500 EST m+=705.503447896) (total time:
10.002043765s):
Trace[2001942447]: [10.002043765s] [10.002043765s] END
E1203 02:39:00.172549    2817 reflector.go:153] k8s.io/client-go/tools/
watch/informerwatcher.go:146: Failed to list *v1.ConfigMap: Get "https://
api.ocp4.example.com:6443/api/v1/namespaces/kube-system/configmaps?
fieldSelector=metadata.name%3Dbootstrap&limit=500&resourceVersion=0": net/http:
TLS handshake timeout
I1203 02:39:11.175147    2817 trace.go:116] Trace[1730834876]: "Reflector
ListAndWatch" name:k8s.io/client-go/tools/watch/informerwatcher.go:146
(started: 2020-12-03 02:39:01.172662955 -0500 EST m+=716.505671444) (total time:
10.002460755s):
Trace[1730834876]: [10.002460755s] [10.002460755s] END
E1203 02:39:11.175176    2817 reflector.go:153] k8s.io/client-go/tools/
watch/informerwatcher.go:146: Failed to list *v1.ConfigMap: Get "https://
api.ocp4.example.com:6443/api/v1/namespaces/kube-system/configmaps?
fieldSelector=metadata.name%3Dbootstrap&limit=500&resourceVersion=0": net/http:
TLS handshake timeout
E1203 02:39:16.233355    2817 reflector.go:153] k8s.io/client-go/tools/
watch/informerwatcher.go:146: Failed to list *v1.ConfigMap: Get "https://
api.ocp4.example.com:6443/api/v1/namespaces/kube-system/configmaps?
fieldSelector=metadata.name%3Dbootstrap&limit=500&resourceVersion=0": EOF
INFO It is now safe to remove the bootstrap resources
INFO Time elapsed: 18m6s
```



Anmerkung

Die in der vorherigen Ausgabe gezeigten Fehler sind auf die Nichtverfügbarkeit der Kubernetes-API im bootstrap-Rechner zurückzuführen. Sobald die Kubernetes-API ausgeführt wird, werden diese Fehler nicht mehr angezeigt.

Sobald die Kubernetes-API verfügbar ist, wird eine Meldung angezeigt, dass die Bootstrap-Ressourcen jetzt sicher entfernt werden können.

► 4. Entfernen Sie den bootstrap-Rechner vom Load Balancer.

- 4.1. Bearbeiten Sie die Datei `/etc/haproxy/haproxy.cfg`, und entfernen Sie die Einträge für den bootstrap-Rechner. Suchen und entfernen Sie die folgenden Einträge:

```
#-----  
# round robin balancing for RHOC P Kubernetes API Server  
#-----  
  
server bootstrap 192.168.50.9:6443 check
```

```
#-----  
# round robin balancing for RHOC P Machine Config Server  
#-----  
  
server bootstrap 192.168.50.9:22623 check
```



Anmerkung

Sie können die Syntax der Datei `haproxy.cfg` mit dem folgenden Befehl validieren:

```
haproxy -c -f /etc/haproxy/haproxy.cfg
```

4.2. Laden Sie die HAProxy-Konfiguration neu.

```
[lab@utility ~]$ sudo systemctl reload haproxy
```

► 5. Überprüfen Sie die Protokolle des Bootstrap-Rechners, indem Sie eine Verbindung über SSH herstellen, um den Fortschritt der Installation zu sehen.

```
[lab@utility ~]$ ssh -i .ssh/ocp4upi core@bootstrap.ocp4.example.com  
  
Red Hat Enterprise Linux CoreOS 46.82.202010091720-0  
Part of OpenShift 4.6, RHCOS is a Kubernetes native operating system  
managed by the Machine Config Operator (`clusteroperator/machine-config`).  
  
WARNING: Direct SSH access to machines is not recommended; instead,  
make configuration changes via `machineconfig` objects:
```

Kapitel 5 | Installieren von OpenShift ohne Infrastrukturanbieter

```
https://docs.openshift.com/container-platform/4.6/architecture/architecture-rhcos.html

---
This is the bootstrap node; it will be destroyed when the master is fully up.

The primary services are release-image.service followed by bootkube.service. To
watch their status, run e.g.

journalctl -b -f -u release-image.service -u bootkube.service
Last login: Thu Dec  3 08:10:24 2020 from 192.168.50.254
[core@bootstrap ~]$ journalctl -b -f -u release-image.service -u bootkube.service
-- Logs begin at Thu 2020-12-03 07:27:53 UTC. --
Dec 03 07:45:01 bootstrap bootkube.sh[2314]: Skipped "secret-initial-kube-
controller-manager-service-account-private-key.yaml" secrets.v1./initial-service-
account-private-key -n openshift-config as it already exists
Dec 03 07:45:03 bootstrap bootkube.sh[2314]: Sending bootstrap-finished
  event.Tearing down temporary bootstrap control plane...
Dec 03 07:45:03 bootstrap bootkube.sh[2314]: Waiting for CEO to finish...
Dec 03 07:45:03 bootstrap bootkube.sh[2314]: I1203 07:45:03.848000      1
  waitforceo.go:64] Cluster etcd operator bootstrapped successfully
Dec 03 07:45:03 bootstrap bootkube.sh[2314]: I1203 07:45:03.849938      1
  waitforceo.go:58] cluster-etcd-operator bootstrap etcd
Dec 03 07:45:03 bootstrap bootkube.sh[2314]: bootkube.service complete
```

Schließen Sie die Verbindung zum Bootstrap-Rechner.

```
[core@bootstrap ~]$ exit
[lab@utility ~]$
```

- 6. Konfigurieren Sie den Zugriff auf den Cluster mit der während der Installation generierten kubeconfig-Datei.

- 6.1. Konfigurieren Sie mithilfe der während der Installation generierten kubeconfig-Authentifizierungsdatei eine KUBECONFIG-Umgebungsvariable.

```
[lab@utility ~]$ export KUBECONFIG=~/ocp4upi/auth/kubeconfig
```

- 6.2. Testen Sie die konfigurierten Anmeldedaten, und überprüfen Sie den aktuellen Status des Clusters.
- Anmeldedaten

```
[lab@utility ~]$ oc whoami
system:admin
```

- Knoten

```
[lab@utility ~]$ oc get nodes
master01  Ready    master    76m    v1.19.0+9f84db3
master02  Ready    master    70m    v1.19.0+9f84db3
master03  Ready    master    69m    v1.19.0+9f84db3
```

- Cluster-Version

```
[lab@utility ~]$ oc get clusterversion
NAME      VERSION  AVAILABLE  PROGRESSING  SINCE    STATUS
version          False       True        84m      Unable to apply 4.6.4: some
cluster operators have not yet rolled out
```

- Cluster-Operatoren

```
[lab@utility ~]$ oc get clusteroperator
NAME                           VERSION  AVAILABLE  PROGRESSING
DEGRADED   SINCE
authentication                  False     False
True      78m
cloud-credential               4.6.4    True      False
False     84m
config-operator                 4.6.4    True      False
False     79m
console                        4.6.4    Unknown   True
False     68m
csi-snapshot-controller        4.6.4    True      False
False     78m
...output omitted...
```

- Statusüberwachung

```
[lab@utility ~]$ watch 'oc get clusterversion; oc get clusteroperators; oc get \
> pods --all-namespaces | grep -v -E "Running|Completed"; oc get csr; oc get
nodes'
```

► 7. Schließen Sie die Installation ab.

- 7.1. Melden Sie sich in einem separaten Terminalfenster beim Server **utility** als Benutzer **lab** an, starten Sie den Befehl **openshift-install**, und überwachen Sie die Statusausgabe auf Meldungen bezüglich des Abschlusses der Installation.

```
[lab@utility ~]$ export KUBECONFIG=~/ocp4upi/auth/kubeconfig
[lab@utility ~]$ openshift-install --dir=ocp4upi wait-for install-complete \
> --log-level=debug
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
DEBUG Loading Install Config...
DEBUG Loading SSH Key...
DEBUG Loading Base Domain...
DEBUG Loading Platform...
DEBUG Loading Cluster Name...
DEBUG Loading Base Domain...
DEBUG Loading Platform...
DEBUG Loading Pull Secret...
```

```
DEBUG Loading Platform...
DEBUG Using Install Config loaded from state file
INFO Waiting up to 40m0s for the cluster at https://api.ocp4.example.com:6443 to
initialize...
```

**Anmerkung**

Die aktuelle Installation ist ausstehend, da die ausstehenden CSRs genehmigt werden müssen, bevor das Verfahren fortgesetzt wird. Fahren Sie mit dem nächsten Schritt fort, um diesen Aspekt der Installation zu bearbeiten.

► 8. Genehmigen Sie die ausstehenden CSRs.

- 8.1. Kehren Sie zum ursprünglichen Terminalfenster zurück, und zeigen Sie den aktuellen Status der Cluster-CSRs an.

```
[lab@utility ~]$ oc get csr
NAME      AGE      SIGNERNAME           REQUESTOR
         CONDITION
csr-2hm74  39m     kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Pending
csr-58szz  101m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Pending
csr-75c4j   55m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Pending
csr-7zx55   8m59s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Pending
...output omitted...
```

- 8.2. Genehmigen Sie mit dem folgenden Befehl alle ausstehenden CSRs.

```
[lab@utility ~]$ oc get csr -o go-template='{{range .items}}{{if not .status}}
{{.metadata.name}}\n{{end}}{{end}}' | xargs oc adm certificate approve
certificatesigningrequest.certificates.k8s.io/csr-2hm74 approved
certificatesigningrequest.certificates.k8s.io/csr-58szz approved
certificatesigningrequest.certificates.k8s.io/csr-75c4j approved
certificatesigningrequest.certificates.k8s.io/csr-7mrvd approved
certificatesigningrequest.certificates.k8s.io/csr-7zx55 approved
...output omitted...
```

**Anmerkung**

Der vorherige Befehl enthält keine Zeilenumbrüche.

Es ist häufig erforderlich, den vorherigen Genehmigungsbefehl mehrmals auszuführen, um sicherzustellen, dass alle generierten CSRs genehmigt wurden. Bei den zusätzlichen CSRs kann es einige Minuten dauern, bis sie ihre Anforderungen übermitteln. Überprüfen Sie den Status mit dem Befehl `oc get csr`, bis keine weiteren CSRs angezeigt werden. Dieser Befehl gibt einen Fehler zurück, wenn keine ausstehenden CSRs vorhanden sind. Sie können dann fortfahren:



Anmerkung

Die erstmalige Ausführung des CSR-Genehmigungsbefehls im vorangegangenen Schritt ist erforderlich, um den Kubelet-Dienst für die Knoten zu genehmigen. Zur Genehmigung der Rechnerobjekte des Knotens ist eine nachfolgende Ausführung dieses Befehls erforderlich.

- ▶ 9. Kehren Sie zum sekundären Terminal zurück, und überwachen Sie die Installationsausgabe, bis sie abgeschlossen ist. Die Installation dauert etwa 20 Minuten.

```
[lab@utility ~]$ openshift-install --dir=ocp4upi wait-for install-complete  
--log-level=debug  
  
...output omitted...  
  
DEBUG Still waiting for the cluster to initialize: Some cluster operators are  
still updating: authentication, console, ingress, kube-storage-version-migrator,  
monitoring  
DEBUG Still waiting for the cluster to initialize: Working towards 4.6.4: 99%  
complete  
DEBUG Still waiting for the cluster to initialize: Multiple errors are preventing  
progress:  
* Cluster operator authentication is reporting a failure:  
WellKnownReadyControllerDegraded: kube-apiserver oauth endpoint  
https://192.168.50.12:6443/.well-known/oauth-authorization-server is not yet  
served and authentication operator keeps waiting (check kube-apiserver operator,  
and check that instances roll out successfully, which can take several minutes  
per instance)  
* Cluster operator console is reporting a failure: RouteHealthDegraded: route not  
yet available, https://console-openshift-console.apps.ocp4.example.com/health  
returns '503 Service Unavailable'  
DEBUG Still waiting for the cluster to initialize: Working towards 4.6.4: 99%  
complete  
DEBUG Still waiting for the cluster to initialize: Working towards 4.6.4: 100%  
complete  
DEBUG Cluster is initialized  
INFO Waiting up to 10m0s for the openshift-console route to be created...  
DEBUG Route found in openshift-console namespace: console  
DEBUG Route found in openshift-console namespace: downloads  
DEBUG OpenShift console route is created  
INFO Install complete!  
INFO To access the cluster as the system:admin user when using 'oc', run 'export  
KUBECONFIG=/home/lab/ocp4upi/auth/kubeconfig'  
INFO Access the OpenShift web-console here: https://console-openshift-console.apps.ocp4.example.com
```

```
INFO Login to the console with user: "kubeadm", and password: "GQabC-Sva8j-nF7r7-M4Ui6"
DEBUG Time elapsed per stage:
DEBUG Cluster Operators: 27m11s
INFO Time elapsed: 27m11s
```



Anmerkung

Die Fehlermeldungen in der Ausgabe des vorherigen Schritts sind ergeben sich daraus, dass die kube-apiserver-Pods noch nicht verfügbar waren. Nachdem die kube-apiserver-Pods auf den Control Plane-Knoten bereitgestellt wurden, ist die Cluster-Installation abgeschlossen.

Nach Abschluss des Installationsskripts erhalten Sie Authentifizierungsinformationen und einen Link zur OpenShift-Web Console.

► **10.** Testen Sie den Zugriff auf die OpenShift-Web Console.

- 10.1. Klicken Sie in der Ausgabe der Installation mit der rechten Maustaste auf den Link zur Web Console, oder rufen Sie `https://console.openshift-console.apps.ocp4.example.com` auf, um den Zugriff zu testen.
- 10.2. Verwenden Sie das in der Ausgabe der Installation bereitgestellte oder in der Datei `/home/lab/ocp4upi/auth/kubeadm-password` enthaltene Passwort auf dem utility-Server, um sich als Benutzer `kubeadm` anzumelden.

► **11.** Führen Sie einen Vergleich der Cluster-Statusausgabe mit den während des Installationsprozesses erfassten Ergebnissen durch.

- Knoten

```
[lab@utility ~]$ oc get nodes
NAME      STATUS    ROLES      AGE     VERSION
master01  Ready     master     173m    v1.19.0+9f84db3
master02  Ready     master     167m    v1.19.0+9f84db3
master03  Ready     master     166m    v1.19.0+9f84db3
worker01  Ready     worker     14m     v1.19.0+9f84db3
worker02  Ready     worker     14m     v1.19.0+9f84db3
```

- Cluster-Version

```
[lab@utility ~]$ oc get clusterversion
NAME      VERSION      AVAILABLE      PROGRESSING      SINCE      STATUS
version   4.6.4        True         False           3m14s     Cluster version is 4.6.4
```

- Cluster-Operatoren

```
[lab@utility ~]$ oc get clusteroperator
NAME                           VERSION      AVAILABLE      PROGRESSING
DEGRADED      SINCE
authentication          4.6.4        True         False
False          3m26s
```

cloud-credential	4.6.4	True	False
False 178m			
cluster-autoscaler	4.6.4	True	False
False 171m			
config-operator	4.6.4	True	False
False 172m			
console	4.6.4	True	False
False 8m46s			
csi-snapshot-controller	4.6.4	True	False
False 171m			
<i>...output omitted...</i>			



Anmerkung

Nach Abschluss der Installation haben alle Cluster-Operatoren den Status AVAILABLE=true, PROGRESSING=False und DEGRADED=False an. Fahren Sie mit den Schritten zur Cluster-Überprüfung fort, um eine ordnungsgemäße Installation zu gewährleisten. Weitere Informationen zur Überprüfung des neuen Clusters finden Sie in den Inhalten im Abschnitt *Überprüfen der Installation von OpenShift auf AWS* des Kapitels *Installieren von OpenShift auf einem Cloud-Anbieter*.

Beenden

Nehmen Sie erst in der nächsten angeleiteten Übung weitere Änderungen an der Übungsumgebung vor. Sie verwenden die Dateien in späteren angeleiteten Übungen.

Hiermit ist die angeleitete Übung beendet.

► Quiz

Kapitelwiederholung: Installieren von OpenShift ohne einen Infrastrukturanbieter

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

► 1. Welche zwei Methoden der Cluster-Authentifizierung stehen nach Abschluss der OpenShift-Installation zur Verfügung? (Wählen Sie zwei Antworten aus.)

- a. htpasswd-Authentifizierung mit dem Benutzer admin.
- b. Exportieren der Umgebungsvariablen KUBECONFIG mithilfe der Datei /auth/kubeconfig im OpenShift-Installationsordner.
- c. Als Benutzer admin mit den Anmelddaten, die in der Datei /auth/kubeadmin im OpenShift-Installationsordner gespeichert sind.
- d. Als Benutzer kubeadmin mit den Anmelddaten, die in der Datei /auth/kubeadmin im OpenShift-Installationsordner gespeichert sind.

► 2. Welcher Befehl kann nach Abschluss der Installation für die Authentifizierung bei OpenShift vom dem Host verwendet werden, auf dem die Installation durchgeführt wurde?

- a. oc login -u system -p admin
- b. oc login -u kubeadmin -p kubeadmin
- c. oc export KUBECONFIG && oc login
- d. export KUBECONFIG=~/ocp4_installation_folder/auth/kubeconfig

► 3. Welche zwei der folgenden Aussagen über den Befehl openshift-install sind richtig? (Wählen Sie zwei Antworten aus.)

- a. Mit openshift-install werden die Kubernetes-Manifeste generiert.
- b. Mit openshift-install werden die RHCOS-Kickstart-Dateien generiert.
- c. Mit openshift-install werden die RHCOS-Ignition-Dateien generiert.
- d. Der Befehl openshift-install muss als root ausgeführt werden.
- e. Mit dem Befehl openshift-install wird der Benutzer kubeadmin nach Abschluss der Installation automatisch bei OpenShift authentifiziert.

► **4. Welche drei der folgenden Aussagen sind für die Certificate Signing Requests (CSRs) bei der OpenShift-Installation ohne Infrastrukturanbieter richtig? (Wählen Sie drei aus.)**

- a. Alle CSRs für die Control Plane-Knoten werden während der Installation automatisch ausgegeben und akzeptiert.
- b. Alle CSRs für alle Knoten werden während der Installation automatisch ausgegeben und akzeptiert.
- c. Die CSRs der Server-Knoten müssen manuell genehmigt werden.
- d. Administratoren müssen für jeden Server-Knoten zwei verschiedene CSRs genehmigen, um eine korrekte Verbindung mit dem Cluster herzustellen.
- e. Administratoren müssen für jeden Server-Knoten nur eine CSR genehmigen, um eine korrekte Verbindung mit dem Cluster herzustellen.

► Lösung

Kapitelwiederholung: Installieren von OpenShift ohne einen Infrastrukturanbieter

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche zwei Methoden der Cluster-Authentifizierung stehen nach Abschluss der OpenShift-Installation zur Verfügung? (Wählen Sie zwei Antworten aus.)
- a. htpasswd-Authentifizierung mit dem Benutzer admin.
 - b. Exportieren der Umgebungsvariablen KUBECONFIG mithilfe der Datei /auth/kubeconfig im OpenShift-Installationsordner.
 - c. Als Benutzer admin mit den Anmelddaten, die in der Datei /auth/kubeadmin im OpenShift-Installationsordner gespeichert sind.
 - d. Als Benutzer kubeadmin mit den Anmelddaten, die in der Datei /auth/kubeadmin im OpenShift-Installationsordner gespeichert sind.
- 2. Welcher Befehl kann nach Abschluss der Installation für die Authentifizierung bei OpenShift vom dem Host verwendet werden, auf dem die Installation durchgeführt wurde?
- a. oc login -u system -p admin
 - b. oc login -u kubeadmin -p kubeadmin
 - c. oc export KUBECONFIG && oc login
 - d. export KUBECONFIG=~/ocp4_installation_folder/auth/kubeconfig
- 3. Welche zwei der folgenden Aussagen über den Befehl openshift-install sind richtig? (Wählen Sie zwei Antworten aus.)
- a. Mit openshift-install werden die Kubernetes-Manifeste generiert.
 - b. Mit openshift-install werden die RHCOS-Kickstart-Dateien generiert.
 - c. Mit openshift-install werden die RHCOS-Ignition-Dateien generiert.
 - d. Der Befehl openshift-install muss als root ausgeführt werden.
 - e. Mit dem Befehl openshift-install wird der Benutzer kubeadmin nach Abschluss der Installation automatisch bei OpenShift authentifiziert.

► **4. Welche drei der folgenden Aussagen sind für die Certificate Signing Requests (CSRs) bei der OpenShift-Installation ohne Infrastrukturanbieter richtig? (Wählen Sie drei aus.)**

- a. Alle CSRs für die Control Plane-Knoten werden während der Installation automatisch ausgegeben und akzeptiert.
- b. Alle CSRs für alle Knoten werden während der Installation automatisch ausgegeben und akzeptiert.
- c. Die CSRs der Server-Knoten müssen manuell genehmigt werden.
- d. Administratoren müssen für jeden Server-Knoten zwei verschiedene CSRs genehmigen, um eine korrekte Verbindung mit dem Cluster herzustellen.
- e. Administratoren müssen für jeden Server-Knoten nur eine CSR genehmigen, um eine korrekte Verbindung mit dem Cluster herzustellen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Der Installationsprozess für die Bereitstellung von OpenShift ohne einen bestimmten Infrastrukturanbieter.
- Die Befehlszeileninteraktionen mit der „openshift-install“-Binärdatei, die die Installation durchführen.
- Die Methoden zur Überwachung und Überprüfung der OpenShift-Installation während des Verfahrens.
- Die verfügbaren Befehlszeileninteraktionen, die während der Installation die Validierung der Prozesse bereitstellen.
- Die Techniken, die während der Installation von OpenShift die ausführlichste Ausgabe bereitstellen.
- Die genauen Interaktionen, Ausgaben und Informationen, die während einer erfolgreichen OpenShift-Installation auftreten.

Kapitel 6

Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

Ziel

Durchführen wesentlicher Aufgaben, die vor dem Onboarding von Benutzern und Anwendungen in einem neu bereitgestellten OpenShift-Cluster erforderlich sind.

Ziele

- Durchführen der erforderlichen Anpassungen, bevor Benutzer und Anwendungen in einen neu installierten Cluster integriert werden.
- Sichern und Wiederherstellen eines Control Plane-Knotens.

Abschnitte

- Durchführen von Day 1- und Day 2-Vorgängen (und angeleitete Übung)
- Ersetzen eines Control Plane-Knotens (und angeleitete Übung)
- Kapitelwiederholung: Abschließen der Installation von OpenShift ohne Infrastrukturanbieter (Test)

Durchführen von Day 1- und Day 2-Vorgängen

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, die erforderlichen Anpassungen durchzuführen, bevor Benutzer und Anwendungen in einen neu installierten Cluster integriert werden.

Erläutern von Day 1- und Day 2-Vorgängen

In Produktionsumgebungen müssen Administratoren OpenShift-Cluster für Endbenutzer bereit machen. Die Aufgaben, die in der Regel direkt nach Abschluss der Installation ausgeführt werden, werden im Allgemeinen als „Day 1“- und „Day 2“-Vorgänge bezeichnet.

Bei „Day 1“-Vorgängen handelt es sich um Anpassungen, die während der Installation vorgenommen werden. Bei „Day 2“-Vorgängen handelt es sich um Anpassungen, die nach Abschluss der Installation und dem Bereitstellen des Clusters für die Endbenutzer vorgenommen werden.

Die Anzahl der „Day 1“- und „Day 2“-Anpassungen ist nicht festgelegt und hängt von den Anforderungen des Unternehmens, den Cluster-Administratoren und den Endbenutzern ab.

Beschreiben der üblichen „Day 1“- und „Day 2“-Vorgänge für OpenShift-Cluster

Bevor Administratoren einen OpenShift-Cluster für die Produktion freigeben, müssen sie eine oder mehrere der folgenden Aufgaben ausführen:

- Sie fügen persistenten Storage für die OpenShift-Image-Registry hinzu oder konfigurieren die Unternehmens-Registry.
- Sie fügen persistenten Storage für den Überwachungs-Stack hinzu, um Metriken für die zukünftige Analyse zu erfassen.
- Sie konfigurieren einen Autorisierungs- und Authentifizierungsanbieter.
- Sie konfigurieren einen dynamischen Storage-Anbieter für den persistenten Storage für die containerisierten Anwendungen.
- Sie führen Funktionstests nach dem Abschluss der gewünschten Konfigurationen durch.

Darüber hinaus kann der Cluster Anpassungen erfordern, wie in Red Hat DO280: *Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster* und DO380: *Red Hat OpenShift Administration III: Scaling Kubernetes Deployments in the Enterprise* erläutert. Zu Anpassungen, die Sie auf den Cluster anwenden können, zählen:

- Einrichten von Netzwerkrichtlinien (DO280).
- Steuern der Pod-Planung (DO280).
- Hinzufügen von Ressourcenlimits (DO280).
- Installieren und Konfigurieren der Cluster-Protokollierung (DO380).

Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

- Konfigurieren von OpenShift-Alarmen (DO380).
- Hinzufügen einer benutzerdefinierten CA (DO380).
- Bereitstellen verschiedener Operatoren für Endbenutzer (DO380).
- Erstellen von benutzerdefinierten Machine Config-Pools (DO380).
- Wiederherstellen fehlgeschlagener Worker-Knoten (DO380).

Weitere Details zu verschiedenen „Day 1“- und „Day 2“-Anpassungen finden Sie im *Post-installation configuration Guide* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/post-installation_configuration/index.

Sie erfahren an anderer Stelle in diesem Kurs, wie ein fehlgeschlagener Control Plane-Knoten gesichert und wiederhergestellt wird.

Schließen Sie die Installation ab.

Nach Abschluss der Cluster-Installation müssen Sie den OpenShift-Cluster in der Kursumgebung weiterhin konfigurieren, damit er voll funktionsfähig ist. Diese Konfigurationsschritte sind für jede OpenShift-Installation auf bereits vorhandener Infrastruktur erforderlich. Nach Abschluss des Befehls `openshift-install` müssen Administratoren möglicherweise eine oder mehrere der folgenden Aufgaben ausführen:

- Sie erstellen eine Kopie der `kubeconfig`-Datei und speichern sie.
- Sie konfigurieren den Image-Registry-Operator durch Hinzufügen vom temporären oder persistenten Storage.
- Sie fügen einen dynamischen Storage-Anbieter hinzu.
- Sie konfigurieren einen Autorisierungs- und Authentifizierungsanbieter.
- Sie führen Funktionstests des Clusters durch.



Warnung

Sie speichern die `kubeconfig`-Datei an einem sicheren Ort. Diese Datei gewährt `cluster-admin`-Berechtigungen im Cluster. Behandeln Sie sie entsprechend.



Literaturhinweise

Weitere Informationen finden Sie im *Post-installation configuration Guide* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/post-installation_configuration/index

► Angeleitete Übung

Durchführen von Day 1- und Day 2-Vorgängen

In dieser Übung führen Sie eine grundlegende Konfiguration durch, bevor Sie einen OpenShift-Cluster für die Produktion freigeben.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Speichern einer Kopie der kubeconfig-Datei
- Konfigurieren eines dynamischen Storage-Anbieters
- Konfigurieren des Registry-Operators, Hinzufügen von persistentem Storage
- Bereitstellen einer Anwendung zum Durchführen eines Funktionstests des Clusters

Bevor Sie Beginnen

Stellen Sie vor Beginn dieser Übung sicher, dass Sie die angeleitete Übung `install-practice` abgeschlossen haben.

Anweisungen

► 1. Speichern Sie die kubeconfig-Datei.

- 1.1. Kopieren Sie als Benutzer `student` die kubeconfig-Datei von `/home/lab/ocp4upi/auth/kubeconfig` auf den Rechner „workstation“.

```
[student@workstation ~]$ ssh lab@utility  
  
[lab@utility ~]$ scp /home/lab/ocp4upi/auth/kubeconfig student@workstation:  
The authenticity of host 'workstation (172.25.250.9)' can't be established.  
ECDSA key fingerprint is SHA256:DDmgpJ0DDjGdZHJRAhDuN6XyPrl43F4IXMYhj5orlnQ.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'workstation,172.25.250.9' (ECDSA) to the list of known hosts.  
student@workstation's password:  
kubeconfig  
100% 8942      5.2MB/s   00:00
```

► 2. Konfigurieren Sie die Image-Registry mit persistentem Storage.

- 2.1. Verwenden Sie die kubeconfig-Datei, um sich bei OpenShift zu authentifizieren.

```
[lab@utility ~]$ export KUBECONFIG=~/ocp4upi/auth/kubeconfig
```

- 2.2. Untersuchen Sie die NFS-Exporte auf dem Server `utility`.

```
[lab@utility ~]$ cat /etc(exports
/exports *(rw,sync,no_wdelay,no_root_squash,insecure,fsid=0)

[lab@utility ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda   252:0    0   10G  0 disk
└─vda1 252:1    0     1M  0 part
└─vda2 252:2    0  100M  0 part /boot/efi
└─vda3 252:3    0  9,9G  0 part /
vdb   252:16   0   40G  0 disk
└─vdb1 252:17   0   40G  0 part /exports

[lab@utility ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
...output omitted...
/dev/vdb1        40G  318M   40G  1% /exports
/dev/vda2       100M  6,8M  94M  7% /boot/efi
tmpfs          183M     0  183M  0% /run/user/1000
```

Für den NFS-Export `/exports` sind ca. 40 GB Speicherplatz verfügbar.

2.3. Erstellen Sie die Persistent Volume-Datei (PV) mit dem Namen `pv.yaml`.

```
[lab@utility ~]$ vi pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: registry-pv
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteMany
  nfs:
    path: /exports/registry
    server: 192.168.50.254
  persistentVolumeReclaimPolicy: Recycle
```

Speichern Sie die Datei, und schließen Sie anschließend den Editor.

2.4. Erstellen Sie den Ordner `/exports/registry` mit den entsprechenden Berechtigungen und dem persistenten Volume (PV) aus der Datei `pv.yaml`:

```
[lab@utility ~]$ mkdir /exports/registry
[lab@utility ~]$ sudo chmod 777 /exports/registry/
[lab@utility ~]$ oc create -f pv.yaml
persistentvolume/registry-pv created
```

2.5. Erstellen Sie eine Datei mit dem Namen `pvc.yaml` mit folgendem Inhalt:

```
[lab@utility ~]$ vi pvc.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: registry-claim
  namespace: openshift-image-registry
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 5Gi
```

Speichern Sie die Datei, und schließen Sie anschließend den Editor.

- 2.6. Erstellen Sie die Anforderung für ein persistentes Volume (PVC) aus der Datei `pvc.yaml`:

```
[lab@utility ~]$ oc create -f pvc.yaml
persistentvolumeclaim/registry-claim created
```

- 2.7. Legen Sie im Abschnitt `spec` für den Image-Registry-Operator den Status „Managed“ fest. Bearbeiten Sie die Konfiguration der Cluster-Image-Registry, um die PVC hinzuzufügen. Konfigurieren Sie außerdem die Image-Registry so, dass sie über zwei Pod-Replikate verfügt.

```
[lab@utility ~]$ oc edit configs.imageregistry/cluster
...output omitted...
spec:
...output omitted...
  managementState: Managed
...output omitted...
  proxy: {}
  replicas: 2
  requests:
...output omitted...
  rolloutStrategy: RollingUpdate
  storage:
    pvc:
      claim: registry-claim
...output omitted...
```



Anmerkung

Vergewissern Sie sich, dass Sie den Abschnitt `spec` und nicht den Abschnitt `status` bearbeiten. Die Bearbeitung des Abschnitts `status` hat keine Auswirkungen auf die Konfiguration.

- 2.8. Überprüfen Sie den PV-, PVC- und Image-Registry-Operator-Status..

```
[lab@utility ~]$ oc get pv -A
NAME          CAPACITY   ACCESS MODES  RECLAIM POLICY  STATUS    CLAIM
           STORAGECLASS
registry-pv   5Gi        RWX          Recycle      Bound     openshift-image-
registry/registry-claim                         111m

[lab@utility ~]$ oc get pvc -A
NAMESPACE          NAME          STATUS    VOLUME          CAPACITY
ACCESS MODES  STORAGECLASS  AGE
openshift-image-registry  registry-claim  Bound    registry-pv  5Gi        RWX
                                         105m

[lab@utility ~]$ oc get clusteroperator
NAME          VERSION  AVAILABLE  PROGRESSING
DEGRADED      SINCE
authentication 4.6.4     True       False
False         174m
cloud-credential 4.6.4     True       False
False         3d
cluster-autoscaler 4.6.4     True       False
False         3d
config-operator 4.6.4     True       False
False         3d
console        4.6.4     True       False
False         175m
csi-snapshot-controller 4.6.4     True       False
False         3d
dns            4.6.4     True       False
False         3d
etcd           4.6.4     True       False
False         3d
image-registry 4.6.4     True       False
False         99m
...output omitted...
```

2.9. Überprüfen Sie die Anzahl der Registry-Pods im Namespace `openshift-image-registry`.

```
[lab@utility ~]$ oc get pods -n openshift-image-registry -o wide
NAME          READY   STATUS    RESTARTS
AGE          IP          NODE
cluster-image-registry-operator-8455b85cc6-z7wcv  1/1     Running   1
78m          10.130.0.18  master03 ...
image-registry-76467959f4-kmdf8  1/1     Running   0
8m38s        10.131.0.21  worker01 ...
image-registry-76467959f4-tcczp  1/1     Running   0
9m45s        10.128.2.39  worker02 ...
node-ca-dczt7  1/1     Running   0
66m          192.168.50.14  worker02 ...
node-ca-gsbtk  1/1     Running   0
67m          192.168.50.13  worker01 ...
```

Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

node-ca-mpbkm	192.168.50.12	master03 ...	1/1	Running	0
node-ca-xjbp8	192.168.50.10	master01 ...	1/1	Running	0
node-ca-xwfsf	192.168.50.11	master02 ...	1/1	Running	0

► 3. Konfigurieren Sie einen dynamischen Storage-Provisioner.

- 3.1. Klonen Sie das GitHub-Repository [kubernetes-sigs/nfs-subdir-external-provisioner](https://github.com/kubernetes-sigs/nfs-subdir-external-provisioner):

```
[lab@utility ~]$ git clone \
> https://github.com/kubernetes-sigs/nfs-subdir-external-provisioner/
...output omitted...
```

**Warnung**

`nfs-subdir-external-provisioner` wird weder von Red Hat unterstützt noch für Produktionsumgebungen empfohlen.

- 3.2. Bearbeiten Sie die Dateien im Ordner `deploy` unter Verwendung der Informationen für den NFS-Server in `utility`. Die Ausgabe sollte wie folgt aussehen:

```
[lab@utility ~]$ vi nfs-subdir-external-provisioner/deploy/class.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: managed-nfs-storage
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: nfs-dynamic-provisioner
reclaimPolicy: Retain
...output omitted...
```

```
[lab@utility ~]$ vi nfs-subdir-external-provisioner/deploy/rbac.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nfs-client-provisioner
  namespace: nfs-dynamic-namespace
---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: nfs-client-provisioner-runner
rules:
  - apiGroups: []
    resources: ["persistentvolumes"]
    verbs: ["get", "list", "watch", "create", "delete"]
  - apiGroups: []
    resources: ["persistentvolumeclaims"]
```

```
    verbs: ["get", "list", "watch", "update"]
  - apiGroups: ["storage.k8s.io"]
    resources: ["storageclasses"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["create", "update", "patch"]
  ---

kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: run-nfs-client-provisioner
subjects:
  - kind: ServiceAccount
    name: nfs-client-provisioner
    namespace: nfs-dynamic-namespace
roleRef:
  kind: ClusterRole
  name: nfs-client-provisioner-runner
  apiGroup: rbac.authorization.k8s.io
  ---

kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: leader-locking-nfs-client-provisioner
  namespace: nfs-dynamic-namespace
rules:
  - apiGroups: [""]
    resources: ["endpoints"]
    verbs: ["get", "list", "watch", "create", "update", "patch"]
  ---

kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: leader-locking-nfs-client-provisioner
  namespace: nfs-dynamic-namespace
subjects:
  - kind: ServiceAccount
    name: nfs-client-provisioner
    namespace: nfs-dynamic-namespace
roleRef:
  kind: Role
  name: leader-locking-nfs-client-provisioner
  apiGroup: rbac.authorization.k8s.io
```

```
[lab@utility ~]$ vi nfs-subdir-external-provisioner/deploy/deployment.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nfs-client-provisioner
  labels:
    app: nfs-client-provisioner
    namespace: nfs-dynamic-namespace
```

```
spec:
  replicas: 1
  strategy:
    type: Recreate
  selector:
    matchLabels:
      app: nfs-client-provisioner
  template:
    metadata:
      labels:
        app: nfs-client-provisioner
    spec:
      serviceAccountName: nfs-client-provisioner
      containers:
        - name: nfs-client-provisioner
          image: quay.io/external_storage/nfs-client-provisioner:latest
          volumeMounts:
            - name: nfs-client-root
              mountPath: /persistentvolumes
        env:
          - name: PROVISIONER_NAME
            value: nfs-dynamic-provisioner
          - name: NFS_SERVER
            value: 192.168.50.254
          - name: NFS_PATH
            value: /exports
      volumes:
        - name: nfs-client-root
          nfs:
            server: 192.168.50.254
            path: /exports
```

3.3. Erstellen Sie die Objekte im Namespace nfs-dynamic-namespace.

```
[lab@utility ~]$ oc create namespace nfs-dynamic-namespace
namespace/nfs-dynamic-namespace created
```

```
[lab@utility ~]$ oc create -f nfs-subdir-external-provisioner/deploy/rbac.yaml
serviceaccount/nfs-client-provisioner created
clusterrole.rbac.authorization.k8s.io/nfs-client-provisioner-runner created
clusterrolebinding.rbac.authorization.k8s.io/run-nfs-client-provisioner created
role.rbac.authorization.k8s.io/leader-locking-nfs-client-provisioner created
rolebinding.rbac.authorization.k8s.io/leader-locking-nfs-client-provisioner
created
```

3.4. Erstellen Sie die Rolle use-scc-hostmount-anyuid im Namespace nfs-dynamic-namespace. Fügen Sie dem Servicekonto nfs-client-provisioner die Rolle use-scc-hostmount-anyuid hinzu.

```
[lab@utility ~]$ oc create role use-scc-hostmount-anyuid --verb=use \
> --resource=scc --resource-name=hostmount-anyuid -n nfs-dynamic-namespace
role.rbac.authorization.k8s.io/use-scc-hostmount-anyuid created
```

```
[lab@utility ~]$ oc project nfs-dynamic-namespace
Now using project "nfs-dynamic-namespace" on server "https://api.ocp4.example.com:6443".
```

```
[lab@utility ~]$ oc adm policy add-role-to-user use-scc-hostmount-anyuid \
> -z nfs-client-provisioner --role=namespace='nfs-dynamic-namespace'
role.rbac.authorization.k8s.io/use-scc-hostmount-anyuid added: "nfs-client-
provisioner"
```

3.5. Erstellen Sie die Bereitstellung.

```
[lab@utility ~]$ oc create -f \
> nfs-subdir-external-provisioner/deploy/deployment.yaml
deployment.apps/nfs-client-provisioner created
```

3.6. Überprüfen Sie alle Ressourcen, die im Namespace nfs-dynamic-namespace erstellt wurden.

```
[lab@utility ~]$ oc get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/nfs-client-provisioner-69785747-h66hh   1/1     Running   0          25s

NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nfs-client-provisioner   1/1     1           1           25s

NAME                           DESIRED  CURRENT   READY   AGE
replicaset.apps/nfs-client-provisioner-69785747  1        1         1        25s
```

3.7. Erstellen Sie die NFS StorageClass.

```
[lab@utility ~]$ oc create -f \
> nfs-subdir-external-provisioner/deploy/class.yaml
storageclass.storage.k8s.io/managed-nfs-storage created
```

```
[lab@utility ~]$ oc get storageclass
NAME                                PROVISIONER          RECLAIMPOLICY
VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
managed-nfs-storage (default)      nfs-dynamic-provisioner   Retain
Immediate                  false            46s
```

3.8. Testen Sie die Bereitstellung des dynamischen NFS-Volumes.

```
[lab@utility ~]$ oc create -f \
> nfs-subdir-external-provisioner/deploy/test-claim.yaml
persistentvolumeclaim/test-claim created
```

Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

```
[lab@utility ~]$ oc get pvc
NAME           STATUS  VOLUME
MODES   STORAGECLASS      AGE
test-claim    Bound    pvc-7c628580-6de8-416c-8750-9e3d315e2e69  1Mi        RWX
          managed-nfs-storage  17s
```

```
[lab@utility ~]$ oc create -f \
> nfs-subdir-external-provisioner/deploy/test-pod.yaml
pod/test-pod created
```

```
[lab@utility ~]$ oc get all
NAME                                READY  STATUS    RESTARTS  AGE
pod/nfs-client-provisioner-69785747-h66hh  1/1    Running   0          13m
pod/test-pod                         0/1    Completed  0          9s

NAME                                READY  UP-TO-DATE  AVAILABLE  AGE
deployment.apps/nfs-client-provisioner  1/1    1           1          13m

NAME                           DESIRED  CURRENT  READY  AGE
replicaset.apps/nfs-client-provisioner-69785747  1        1        1       13m
```

- 3.9. Wenn der dynamische NFS-Provisioner ordnungsgemäß funktioniert, befindet sich eine Datei mit dem Namen SUCCESS im NFS-Exportordner.

```
[lab@utility ~]$ tree /exports/
(exports/
└── nfs-dynamic-namespace-test-claim-pvc-7c628580-6de8-416c-8750-9e3d315e2e69
    └── SUCCESS
└── registry

2 directories, 1 file
```

**Anmerkung**

Wenn die Erstellung der Test-PVC oder des Test-Pods fehlschlägt, überprüfen Sie die in den vorherigen Schritten an den YAML-Dateien vorgenommenen Änderungen. Wenn Sie vergessen haben, einen Parameter zu ändern, löschen Sie die Ressourcen mit dem Befehl `oc delete -f`, und erstellen Sie sie mit der richtigen YAML-Datei neu.

- 4. Führen Sie einen Funktionstest des Clusters durch.

- 4.1. Erstellen Sie ein neues Projekt mit dem Namen `etherpad`, und legen Sie es als Arbeitsprojekt fest.

```
[lab@utility ~]$ oc new-project etherpad
...output omitted...
```

- 4.2. Erstellen Sie einen neuen Ordner mit dem Namen `etherpad`, und wechseln Sie zu diesem.

```
[lab@utility ~]$ mkdir etherpad  
[lab@utility ~]$ cd etherpad  
[lab@utility etherpad]$
```

- 4.3. Laden Sie die folgenden Dateien herunter, und speichern Sie den Inhalt im Ordner etherpad.

```
[lab@utility etherpad]$ SOL_URL=http://classroom.example.com/materials/solutions  
[lab@utility etherpad]$ wget ${SOL_URL}/etherpad/etherpad-svc.yaml  
...output omitted...  
[lab@utility etherpad]$ cat etherpad-svc.yaml  
apiVersion: v1  
kind: Service  
metadata:  
  name: etherpad  
  labels:  
    app.kubernetes.io/name: etherpad  
    app.kubernetes.io/version: "latest"  
spec:  
  type: ClusterIP  
  ports:  
    - port: 9001  
      targetPort: http  
      protocol: TCP  
      name: http  
  selector:  
    app.kubernetes.io/name: etherpad
```

```
[lab@utility etherpad]$ wget ${SOL_URL}/etherpad/etherpad-route.yaml  
...output omitted...  
[lab@utility etherpad]$ cat etherpad-route.yaml  
apiVersion: route.openshift.io/v1  
kind: Route  
metadata:  
  annotations:  
    openshift.io/host.generated: "true"  
  name: etherpad  
  labels:  
    app.kubernetes.io/name: etherpad  
    app.kubernetes.io/version: "latest"  
spec:  
  host:  
  port:  
    targetPort: http  
  to:  
    kind: Service  
    name: etherpad  
    weight: 100  
  tls:  
    insecureEdgeTerminationPolicy: Redirect  
    termination: edge
```

Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

```
[lab@utility etherpad]$ wget ${SOL_URL}/etherpad/etherpad-pvc.yaml
...output omitted...
[lab@utility etherpad]$ cat etherpad-pvc.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: etherpad
  labels:
    app.kubernetes.io/name: etherpad
    app.kubernetes.io/version: "latest"
spec:
  accessModes:
    - "ReadWriteOnce"
  resources:
    requests:
      storage: "1Gi"
```

```
[lab@utility etherpad]$ wget ${SOL_URL}/etherpad/etherpad-deployment.yaml
...output omitted...
[lab@utility etherpad]$ cat etherpad-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: etherpad
  labels:
    app.kubernetes.io/name: etherpad
    app.kubernetes.io/version: "latest"
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: etherpad
  template:
    metadata:
      labels:
        app.kubernetes.io/name: etherpad
    spec:
      securityContext:
        {}
      containers:
        - env:
            - name: TITLE
              value: D0322 Etherpad
            - name: DEFAULT_PAD_TEXT
              value: Etherpad for sharing ideas between the students.
        name: etherpad
        securityContext:
          {}
      image: "quay.io/redhattraining/etherpad:latest"
      imagePullPolicy: IfNotPresent
      ports:
        - name: http
          containerPort: 9001
```

```

        protocol: TCP
livenessProbe:
  httpGet:
    path: /
    port: http
readinessProbe:
  httpGet:
    path: /
    port: http
resources:
  {}
volumeMounts:
  - name: etherpad-data
    mountPath: /opt/etherpad-lite/var
volumes:
  - name: etherpad-data
    persistentVolumeClaim:
      claimName: etherpad

```

- 4.4. Erstellen Sie die OpenShift-Ressourcen mit den Dateien, die Sie im Ordner `etherpad` erstellt haben.

```
[lab@utility etherpad]$ oc create -f etherpad-pvc.yaml
persistentvolumeclaim/etherpad created
```

```
[lab@utility etherpad]$ oc create -f etherpad-svc.yaml
service/etherpad created
```

```
[lab@utility etherpad]$ oc create -f etherpad-route.yaml
route.route.openshift.io/etherpad created
```

```
[lab@utility etherpad]$ oc create -f etherpad-deployment.yaml
deployment.apps/etherpad created
```

- 4.5. Verifizieren Sie die Erstellung der Ressourcen:

```
[lab@utility etherpad]$ oc get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/etherpad-c7476d8d8-8b8j5       1/1     Running   0          25s

NAME              TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/etherpad   ClusterIP   172.30.15.113 <none>        9001/TCP   31s

NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/etherpad  1/1     1           0           25s

NAME                           DESIRED   CURRENT   READY   AGE
replicaset.apps/etherpad-c7476d8d8  1         1         1       25s

NAME          HOST/PORT   PATH
SERVICES     PORT   TERMINATION   WILDCARD
```

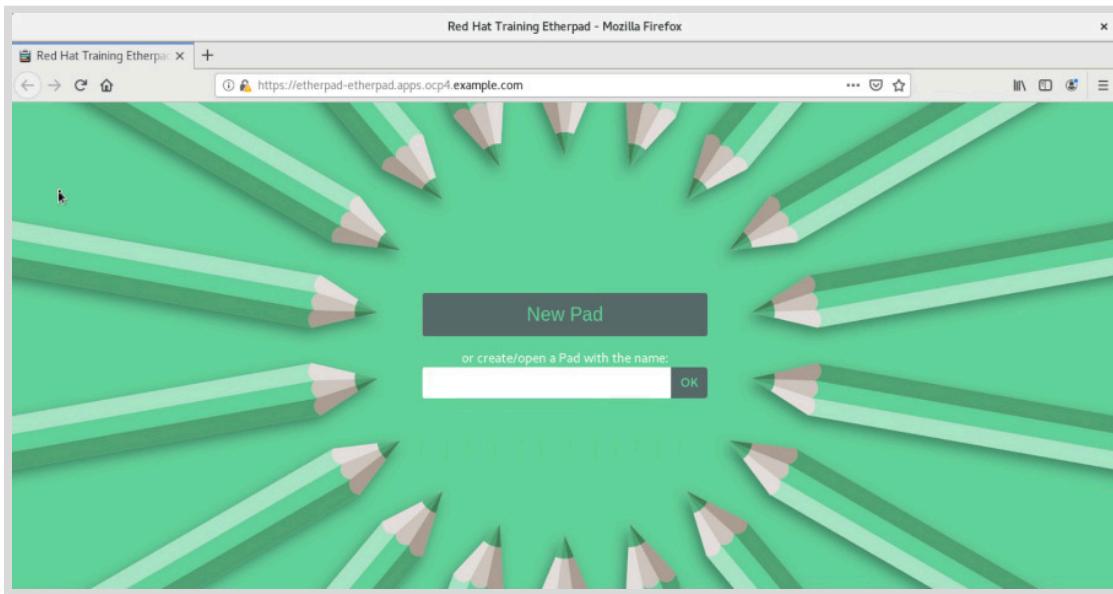
Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

```
route.route.openshift.io/etherpad etherpad-etherpad.apps.ocp4.example.com
  etherpad http edge/Redirect None

[lab@utility etherpad]$ oc get events -o template --template \
> '{{range .items}}{{.message}}\n{{end}}'
...output omitted...
Created pod: etherpad-c7476d8d8-pnv4r
waiting for a volume to be created, either by external provisioner "nfs-dynamic-provisioner" or manually created by system administrator
External provisioner is provisioning volume for claim "etherpad/etherpad"
Successfully provisioned volume pvc-17d90bb5-ae24-46e5-a264-4e057383cdd4
...output omitted...
```

Das PV wurde automatisch von nfs-dynamic-provisioner erstellt.

- 4.6. Öffnen Sie auf dem Rechner „workstation“ den Firefox-Webbrowser, und navigieren Sie zu `etherpad-etherpad.apps.ocp4.example.com`.



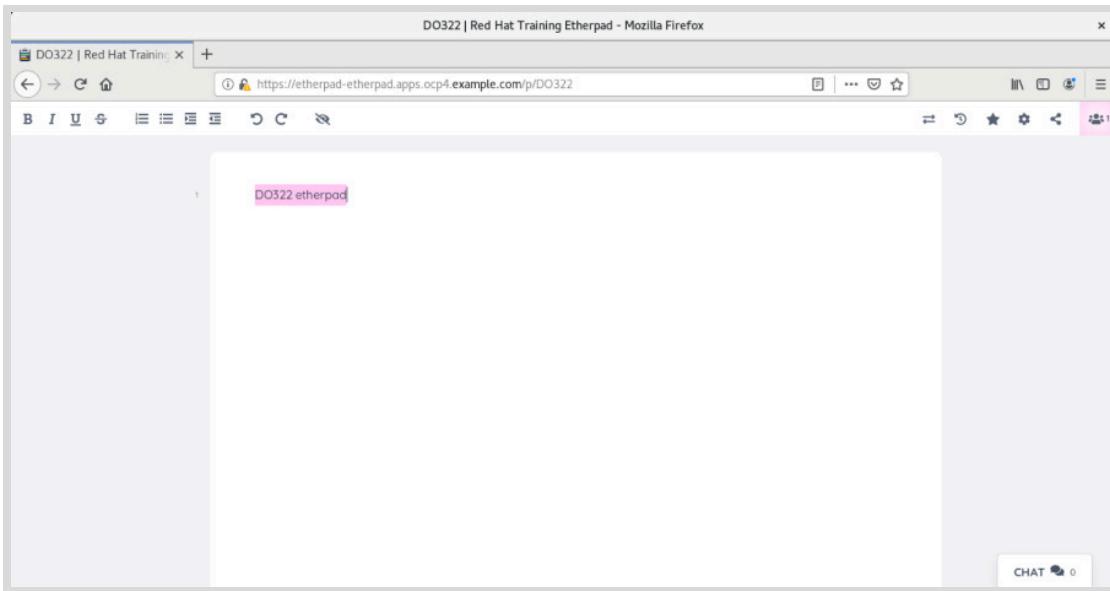
- 4.7. Erstellen Sie ein neues Etherpad mit dem Namen „DO322“.

**Anmerkung**

Wenn Sie von Firefox über ein potenzielles Sicherheitsrisiko informiert werden, klicken Sie auf **Advanced** und dann auf **Accept the Risk and Continue**.

Entfernen Sie den Text im Etherpad, und geben Sie „DO322 etherpad“ ein.

Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter



- 4.8. Löschen Sie im Terminal den Pod „etherpad“, und stellen Sie sicher, dass die Datenbank erhalten bleibt.

```
[lab@utility etherpad]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
etherpad-c7476d8d8-pnv4r   1/1     Running   0          23m
```

```
[lab@utility etherpad]$ oc delete pod etherpad-c7476d8d8-pnv4r
pod "etherpad-c7476d8d8-pnv4r" deleted
```

Verifizieren Sie die Erstellung des neuen Pods:

```
[lab@utility etherpad]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
etherpad-c7476d8d8-sfvsj   1/1     Running   0          26s
```

- 4.9. Kehren Sie zum Firefox-Webbrowser zurück. Laden Sie die Webseite neu, und stellen Sie sicher, dass der Text „DO322 etherpad“ weiterhin vorhanden ist.

- 5. Kehren Sie zum Ordner „home“ von lab zurück, und bereinigen Sie die Umgebung:

```
[lab@utility etherpad]$ cd ~
[lab@utility ~]$ oc delete project etherpad
project.project.openshift.io "etherpad" deleted
```

Beenden

Nehmen Sie erst in der nächsten angeleiteten Übung weitere Änderungen an der Übungsumgebung vor. Sie verwenden die Dateien in späteren angeleiteten Übungen.

Hiermit ist die angeleitete Übung beendet.

Ersetzen eines Control Plane-Knotens

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, einen Control Plane-Knoten zu sichern und wiederherzustellen.

Erläuterung von etcd-Backups

Die OpenShift-etcd-Datenbank ist ein Schlüssel-Wert-Speicher mit Cluster-Statusdaten. Um einen OpenShift-Cluster aus einem vorherigen Zustand wiederherzustellen, müssen Administratoren zunächst ein etcd-Backup durchführen.

Ein etcd-Backup enthält den Status des Clusters und die Konfiguration der statischen Pods, die auf einem Control Plane-Knoten ausgeführt werden. Es ist nicht erforderlich, ein Backup von jedem der drei Control Plane-Knoten zu speichern. Das Backup eines der Control Plane-Knoten ist ausreichend.

Regelmäßige etcd-Backups sind erforderlich, um sicherzustellen, dass ein Cluster wiederhergestellt werden kann, wenn alle Control Plane-Knoten verloren gehen oder der Cluster aufgrund von menschlichen Fehlern oder einem Hardware-Ausfall instabil wird.



Anmerkung

Die standardmäßige Größe des etcd-Speicherplatzkontingents beträgt etwa 7,5 GB und wird durch die Variable `etcd_server_quota_backend_bytes` definiert. Wenn eine etcd-Datenbank dieses Kontingent überschreitet, müssen Sie sie defragmentieren und komprimieren.

Weitere Informationen finden Sie im Abschnitt *Recommended etcd practices* im *Scalability and Performance Guide* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/scalability_and_performance/index#recommended-etcd-practices_

Wiederherstellen des Clusters in einen vorherigen Zustand

OpenShift bietet automatische Reparaturfunktionen aufgrund der Art von Kubernetes-Operatoren. Manchmal, wenn ein Cluster beispielsweise falsch konfiguriert oder Daten beschädigt wurden, kann sich ein OpenShift-Cluster möglicherweise nicht selbst reparieren. Wenn sich der Cluster nicht selbst reparieren kann, ermöglicht OpenShift die Wiederherstellung des Clusters in einen früheren Zustand mithilfe eines etcd-Backups.

Das Wiederherstellungsverfahren eines OpenShift-Clusters erfordert:

- SSH-Zugriff auf die Control Plane-Knoten
- Benutzer mit `cluster-admin`-Berechtigungen
- Ein etcd-Backup

Weitere Informationen zu Disaster Recovery finden Sie im Kapitel *Disaster recovery* im *Red Hat OpenShift Container Platform Backup and Restore Guide* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/backup_and_restore/index#disaster-recovery.

Wiederherstellen eines verlorenen Control Plane-Knotens

In Produktionsumgebungen können Control Plane-Knoten aufgrund von menschlichen Fehlern oder einem Hardware-Ausfall verloren gehen.

Zum Ersetzen eines fehlerhaften Control Plane-Knotens oder etcd-Mitglieds müssen Cluster-Administratoren beim Erstellen des neuen Knotens die gleiche Rechnerkonfiguration verwenden wie bei der Erstellung des fehlerhaften, zu ersetzenen Knotens.

Die Rechnerkonfiguration wird in Abhängigkeit der ursprünglichen Installationsmethode in verschiedenen Formaten gespeichert. Rufen Sie bei OpenShift-Installationen mit der Full-Stack-Automatisierung die Rechnerkonfiguration über die OpenShift-Rechner-API ab. Bei Installationen auf bereits vorhandener Infrastruktur sind die Ignition-Dateien erforderlich.

An anderer Stelle in diesem Kapitel erfahren Sie, wie Sie einen verlorenen Control Plane-Knoten mithilfe der Kursumgebung erneut bereitstellen.



Anmerkung

Für die Zukunft plant Red Hat das automatische Handling und Ersetzen von verlorenen Control Plane-Mitgliedern durch den etcd-Operator.



Literaturhinweise

Weitere Informationen finden Sie im Kapitel *Post-Installation Cluster Tasks* des *Red Hat OpenShift Container Platform Post-Installation Configuration Guide* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/post-installation_configuration/

Weitere Informationen finden Sie zudem im *Red Hat OpenShift Container Platform Backup and Restore Guide* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/backup_and_restore/

► Angeleitete Übung

Ersetzen eines Control Plane-Knotens

In dieser Übung installieren Sie einen fehlgeschlagenen Control Plane-Knoten neu.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Generieren eines etcd-Backups
- Wiederherstellen eines fehlgeschlagenen Control Plane-Knotens

Bevor Sie Beginnen

Um diese Übung durchzuführen, müssen Sie alle vorherigen Übungen in diesem Kurs abgeschlossen haben.

Die Kursumgebung ist abgesehen von den Kursübungen unverändert.

Wenn Sie zusätzliche Änderungen an der Umgebung vorgenommen haben, erstellen Sie eine neue Kursumgebung, bevor Sie beginnen. Sie müssen die Schritte aus den angeleiteten Übungen in den vorherigen Kapiteln wiederholen, um sich auf diese Übung vorzubereiten.

Anweisungen

- 1. Verwenden Sie die kubeconfig-Datei, um sich bei OpenShift über den Server **utility** zu authentifizieren.
- 1.1. Melden Sie sich vom Rechner **workstation** als Benutzer „lab“ beim Rechner **utility** an:

```
[student@workstation ~]$ ssh lab@utility  
...output omitted...
```

- 1.2. Exportieren Sie die Variable **KUBECONFIG** zum Authentifizieren bei OpenShift:

```
[lab@utility ~]$ export KUBECONFIG=~/ocp4upi/auth/kubeconfig
```

- 2. Erstellen Sie das etcd-Backup.
- 2.1. Starten Sie eine Debug-Sitzung auf dem Control Plane-Knoten **master01**. Ändern Sie das Root-Verzeichnis in **/host**.

```
[lab@utility ~]$ oc get nodes  
NAME      STATUS    ROLES     AGE      VERSION  
master01   Ready     master    2d18h    v1.19.0+9f84db3  
master02   Ready     master    2d18h    v1.19.0+9f84db3  
master03   Ready     master    2d18h    v1.19.0+9f84db3  
worker01   Ready     worker    2d17h    v1.19.0+9f84db3  
worker02   Ready     worker    2d17h    v1.19.0+9f84db3
```

```
[lab@utility ~]$ oc debug node/master01
Creating debug namespace/openshift-debug-node-sws5d ...
Starting pod/master01-debug ...
To use host binaries, run chroot /host
Pod IP: 192.168.50.10
If you don't see a command prompt, try pressing enter.
sh-4.4# chroot /host
sh-4.4#
```

- 2.2. Führen Sie das etcd-Backup-Skript in /usr/local/bin/cluster-backup.sh aus. Speichern Sie die Ausgabe in /home/core/backup.

```
sh-4.4# /usr/local/bin/cluster-backup.sh /home/core/backup
...output omitted...
snapshot db and kube resources are successfully saved to /home/core/backup
```

- 2.3. Listen Sie die Ressourcen auf, die vom Backup-Skript erstellt wurden.

```
sh-4.4# ls /home/core/backup/
snapshot_2020-12-03_110352.db  static_kuberestores_2020-12-03_110352.tar.gz
```

- Die Datei snapshot_2020-12-03_110352.db ist der etcd-Snapshot.
- Die Datei „static_kuberestores_2020-12-03_110352.tar.gz“ enthält die statischen Pod-Ressourcen: kube-apiserver, kube-controller-manager, kube-scheduler und etcd.

- 2.4. Ändern Sie die Berechtigungen des Backup-Ordners so, dass der Benutzer core und die Gruppe core die neuen Besitzer sind.

```
sh-4.4# chown -R core:core /home/core/backup
```

- 2.5. Schließen Sie die Debug-Sitzung.

```
sh-4.4# exit
exit
sh-4.4# exit
exit

Removing debug pod ...
Removing debug namespace/openshift-debug-node-85dg7 ...
```

- 2.6. Erstellen Sie den neuen Ordner /home/lab/etc_backup, und speichern Sie die Backup-Dateien auf dem Server utility. Verwenden Sie den SSH-Schlüssel ocp4upi, den Sie als Voraussetzung generiert haben. Geben Sie yes ein, um die Verbindung zum Server master01 herzustellen.

```
[lab@utility ~]$ mkdir /home/lab/etc_backup
[lab@utility ~]$ scp -i .ssh/ocp4upi \
> core@192.168.50.10:/home/core/backup/* /home/lab/etc_backup/
The authenticity of host '192.168.50.10 (192.168.50.10)' can't be established.
```

Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

```
ECDSA key fingerprint is SHA256:1K205E7PCvFYMAcnHH26Sy1wHowbuEN1bACvyu+WjCU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.50.10' (ECDSA) to the list of known hosts.
snapshot_2020-12-16_105145.db
 100% 73MB 181.3MB/s 00:00
static_kubernetes_2020-12-16_105145.tar.gz
 100% 65KB 52.1MB/s 00:00
```

- 3. Entfernen Sie einen Control Plane-Knoten.

**Anmerkung**

Dieses Verfahren zum Wiederherstellen eines fehlgeschlagenen Control Plane-Mitglieds gilt, wenn der ausgestorbene Rechner nicht ausgeführt wird oder sich im Status **NotReady** befindet.

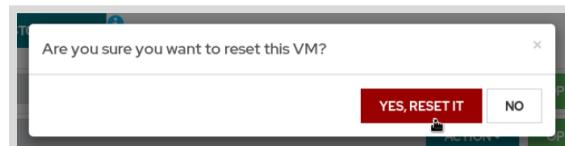
Wenn der Rechner ausgeführt wird und das Control Plane-Mitglied den Status **Ready** aufweist, der etcd-Pod jedoch den Status **Error** aufweist, müssen Sie das im Abschnitt *Replacing an unhealthy etcd member whose etcd pod is crashlooping* des *Red Hat OpenShift Container Platform Backup and Restore Guide* beschriebene Verfahren unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/backup_and_restore/ befolgen.

- 3.1. Setzen Sie den Rechner **master01** in der Übungsumgebung zurück.

Navigieren Sie auf dem Rechner **workstation** mit Firefox zur Registerkarte **Lab Environment** auf der Webseite "Red Hat OpenShift installation Lab". Klicken Sie für den Server **master01** auf **ACTION**, und wählen Sie dann **Reset** aus, wie in der folgenden Abbildung gezeigt:

		DELETE	STOP	i
bastion	active		ACTION	OPEN CONSOLE
bootstrap	active		ACTION	OPEN CONSOLE
classroom	active		ACTION	OPEN CONSOLE
master01	active		ACTION	OPEN CONSOLE
master02	active		Start	OPEN CONSOLE
master03	active		Shutdown Power Off	OPEN CONSOLE
utility	active		Reset	OPEN CONSOLE
worker01	active		ACTION	OPEN CONSOLE

Wenn Sie aufgefordert werden, die Zurücksetzung zu bestätigen, klicken Sie auf **YES, RESET IT**.



Dadurch wird der Server **master01** auf den Anfangsstatus zurückgesetzt, als die Übungsumgebung erstellt wurde (eine leere Disk, für PXE-Boot bereit).

- 3.2. Überprüfen Sie auf dem Server **utility** den Status des Servers **master01** und des etcd-Clusters. Warten Sie, bis NotReady für den Server **master01** angezeigt wird. Drücken Sie zum Beenden **Strg+C**.

```
[lab@utility ~]$ oc get nodes -w
NAME      STATUS    ROLES   AGE     VERSION
master01  NotReady master   73m    v1.19.0+9f84db3
master02  Ready     master   73m    v1.19.0+9f84db3
master03  Ready     master   73m    v1.19.0+9f84db3
worker01  Ready     worker   47m    v1.19.0+9f84db3
worker02  Ready     worker   47m    v1.19.0+9f84db3
```

```
[lab@utility ~]$ oc get etcd -o=jsonpath='{range .items[0].status.conditions[?(@.type=="EtcdMembersAvailable")]}{.message}{"\n"}'
2 of 3 members are available, master01 is unhealthy
```

```
[lab@utility ~]$ oc get nodes -o jsonpath='{range .items[*]}{"\n"}{.metadata.name}{"\t"}{range .spec.taints[*]}{.key}{ " "}' | grep unreachable
master01 node-role.kubernetes.io/master node.kubernetes.io/unreachable
node.kubernetes.io/unreachable
```

Der Knoten **master01** ist **NotReady** und hat den Taint **node.kubernetes.io/unreachable**.



Anmerkung

Die auf dem Server **master01** ausgeführten Pods zeigen möglicherweise nach Unterbrechung der Serververbindung einige Zeit lang **Running** als Status. Dies liegt daran, dass das auf dem Server **master01** ausgeführte Kubelet den Status **NotReady** nicht melden konnte.



Anmerkung

Da Sie ein Control Plane-Mitglied gelöscht haben, können temporäre Fehler in Pods oder Services des Clusters auftreten. Wenn dies der Fall ist, ignorieren Sie die Fehlermeldungen, und wiederholen Sie Ihre Befehle, um die Übung abzuschließen.

- 4. Entfernen Sie das fehlerhafte etcd-Mitglied.

- 4.1. Entfernen Sie in einem ausgeführten etcd-Pod das fehlende etcd-Mitglied.

```
[lab@utility ~]$ oc get pods -n openshift-etcd | grep etcd
etcd-master01                      3/3    Running   0          89m
etcd-master02                      3/3    Running   0          88m
etcd-master03                      3/3    Running   0          90m
etcd-quorum-guard-644f5747b8-bb8vw  1/1    Running   0          93m
etcd-quorum-guard-644f5747b8-c9b5k  1/1    Running   0          93m
etcd-quorum-guard-644f5747b8-fbc5f  1/1    Running   0          93m

[lab@utility ~]$ oc rsh -n openshift-etcd etcd-master02
Defaulting container name to etcdctl.
```

Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

```
Use 'oc describe pod/etcdb-master02 -n openshift-etcd' to see all of the containers
in this pod.

sh-4.4# etcdctl member list -w table
+-----+-----+-----+
+-----+-----+
|      ID      | STATUS | NAME   |          PEER ADDRS           |
| CLIENT ADDRS |        | IS LEARNER |
+-----+-----+-----+
+-----+-----+
| 6c05b85443152afa | started | master01 | https://192.168.50.10:2380 |
| https://192.168.50.10:2379 |     false |
| a8fc53b7e8e11c19 | started | master02 | https://192.168.50.11:2380 |
| https://192.168.50.11:2379 |     false |
| f955e62306188c83 | started | master03 | https://192.168.50.12:2380 |
| https://192.168.50.12:2379 |     false |
+-----+-----+-----+
+-----+-----+
```

**Warnung**

Die etcd-Mitglieder werden nach dem Feld ID und nicht nach NAME sortiert.
Vergewissern Sie sich, dass Sie die ID des master01-Mitglieds-ID verwenden. Sie ist möglicherweise nicht die erste in der Liste.

Entfernen Sie dann das fehlerhafte etcd-Mitglied.

```
sh-4.4# etcdctl member remove 6c05b85443152afa
Member 6c05b85443152afa removed from cluster c73435d0ce2db908

sh-4.4# etcdctl member list -w table
+-----+-----+-----+
+-----+-----+
|      ID      | STATUS | NAME   |          PEER ADDRS           |
| CLIENT ADDRS |        | IS LEARNER |
+-----+-----+-----+
+-----+-----+
| a8fc53b7e8e11c19 | started | master02 | https://192.168.50.11:2380 |
| https://192.168.50.11:2379 |     false |
| f955e62306188c83 | started | master03 | https://192.168.50.12:2380 |
| https://192.168.50.12:2379 |     false |
+-----+-----+-----+
+-----+-----+
sh-4.4# exit
exit
```

Das etcd-Mitglied master01 wurde nun entfernt.

4.2. Entfernen Sie die vom entfernten etcd-Mitglied verwendeten Secrets.

```
[lab@utility ~]$ oc get secrets -n openshift-etcd | grep master01
etcd-peer-master01          kubernetes.io/tls            2      102m
etcd-serving-master01        kubernetes.io/tls            2      102m
etcd-serving-metrics-master01 kubernetes.io/tls            2      102m
```

Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

```
[lab@utility ~]$ oc delete secret -n openshift-etcd etcd-peer-master01
secret "etcd-peer-master01" deleted
```

```
[lab@utility ~]$ oc delete secret -n openshift-etcd etcd-serving-master01
secret "etcd-serving-master01" deleted
```

```
[lab@utility ~]$ oc delete secret -n openshift-etcd etcd-serving-metrics-master01
secret "etcd-serving-metrics-master01" deleted
```

4.3. Löschen Sie Knotenobjekt master01.

```
[lab@utility ~]$ oc delete node master01
node "master01" deleted
```

```
[lab@utility ~]$ oc get nodes
NAME      STATUS    ROLES     AGE      VERSION
master02   Ready     master    105m    v1.19.0+9f84db3
master03   Ready     master    106m    v1.19.0+9f84db3
worker01   Ready     worker    79m     v1.19.0+9f84db3
worker02   Ready     worker    79m     v1.19.0+9f84db3
```

- ▶ 5. Verwenden Sie auf dem Rechner **workstation** Firefox, um den neuen Knoten **master01** über die Registerkarte **Lab Environment** im Red Hat Learning-Portal bereitzustellen.
 - 5.1. Öffnen Sie mit der Schaltfläche **Console** die Web Console, und installieren Sie über das PXE-Boot-Menü den Knoten **master01**.

The screenshot shows the Red Hat Learning Portal interface for managing lab environments. At the top, there are tabs for 'Table of Contents', 'Course', and 'Lab Environment'. Below the tabs, there's a section titled '▶ Lab Controls' with instructions to click 'CREATE' to build the lab environment. It also notes that deleting the lab will remove all virtual machines. A red 'WATCH TUTORIAL' button is visible. Below this, a table lists five virtual machines:

Name	Status	Action	Open Console
bastion	active	ACTION ▾	OPEN CONSOLE
bootstrap	active	ACTION ▾	OPEN CONSOLE
classroom	active	ACTION ▾	OPEN CONSOLE
master01	active	ACTION ▾	OPEN CONSOLE ↴
master02	active	ACTION ▾	OPEN CONSOLE

At the bottom, there's a callout box with a pencil icon labeled 'Anmerkung' (Note). It contains the text: 'Wenn die RHCOS-Installation nicht gestartet wird, verwenden Sie Strg+Alt+Entf auf der Web Console des Servers, um einen Neustart auszulösen.'

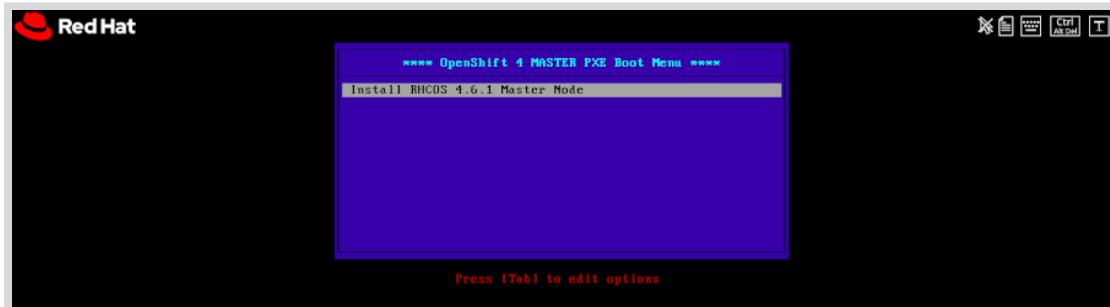


Anmerkung

Im Firefox-Browser wird eine Warnmeldung angezeigt. Klicken Sie auf **Preferences**, und gewähren Sie Firefox die Erlaubnis, die Popup-Fenster für rol.redhat.com anzuzeigen.

Starten Sie den virtuellen Rechner **master01** über die Konsole des Red Hat Learning-Portals neu, indem Sie in der oberen rechten Ecke auf **Ctrl Alt Del** klicken.

Drücken Sie nach dem Neustart die **Eingabetaste**, um die RHCOS-Installation im PXE-Boot-Menü zu starten.



Nach zwei Neustarts wird der neue Control Plane-Knoten **master01** installiert.

```
Red Hat Enterprise Linux CoreOS 46.82.202010091720-0 (Dotpa) 4.6
SSH host key: SHA256:iw129cjI18uuua3rrggCeq1W0Sj4bU0na,jtrIFY+EEs4 (ECDSA)
SSH host key: SHA256:7X530uu79HWyHqHC2H7Eu0EPkqaHb/J0+1gC+Dfyzyg (ED25519)
SSH host key: SHA256:642ig0QbYoGMpkrWFZ57Geb04q5SqCPHKBJuYW9oaQk (RSA)
ens3: 192.168.50.10
master01 login:
```

- 6. Verwenden Sie das Terminalfenster auf dem Server **utility**, um dem Cluster den neuen Knoten **master01** hinzuzufügen.

- 6.1. Genehmigen Sie die Kubelet- und Knotenzertifikate.

```
[lab@utility ~]$ oc get csr | grep Pending
csr-7qqzn    2m1s    kubernetes.io/kube-apiserver-client-kubelet
  system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
  Pending
```

```
[lab@utility ~]$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}\n{{end}}{{end}}' | xargs oc adm certificate approve
certificatesigningrequest.certificates.k8s.io/csr-7qqzn approved
```

```
[lab@utility ~]$ oc get csr | grep Pending
csr-48rhz    4s      kubernetes.io/kubelet-serving
  system:node:master01
  Pending
```

```
[lab@utility ~]$ oc get csr -o go-template='{{range .items}}{{if not .status}} {{.metadata.name}}{{"\n"}}{{end}}{{end}}' | xargs oc adm certificate approve certificatesigningrequest.certificates.k8s.io/csr-48rhz approved
```



Anmerkung

Es können mehrere CSRs im Cluster ausstehen. Machen Sie sich keine Sorgen, wenn die Ausgabe von `oc get csr | grep Pending` mehr als einen Eintrag enthält.

6.2. Überprüfen Sie den Status der Knoten im Cluster.

```
[lab@utility ~]$ oc get nodes
NAME      STATUS    ROLES     AGE      VERSION
master01   Ready     master    6m2s    v1.19.0+9f84db3
master02   Ready     master    120m    v1.19.0+9f84db3
master03   Ready     master    120m    v1.19.0+9f84db3
worker01   Ready     worker    93m     v1.19.0+9f84db3
worker02   Ready     worker    93m     v1.19.0+9f84db3
```

6.3. Überprüfen Sie den Status der etcd-Pods im Cluster.

```
[lab@utility ~]$ oc get pods -n openshift-etcd | grep etcd
etcd-master01                      3/3     Running   0          9m7s
etcd-master02                      3/3     Running   0          7m35s
etcd-master03                      3/3     Running   0          8m37s
etcd-quorum-guard-644f5747b8-bb8vw  1/1     Running   0          121m
etcd-quorum-guard-644f5747b8-c9b5k  1/1     Running   0          121m
etcd-quorum-guard-644f5747b8-dfrcj  1/1     Running   0          13m
```

6.4. Überprüfen Sie die Mitglieder des etcd-Clusters im etcd-Pod in `master01`.

```
[lab@utility ~]$ oc rsh -n openshift-etcd etcd-master01
Defaulting container name to etcdctl.
Use 'oc describe pod/etcd-master01 -n openshift-etcd' to see all of the containers
in this pod.
sh-4.4# etcdctl member list -w table
+-----+-----+-----+
+-----+-----+-----+
|       ID      | STATUS  | NAME   |           PEER ADDRS      |
| CLIENT ADDRS |          | IS LEARNER |           |
+-----+-----+-----+
+-----+-----+-----+
| a8fc53b7e8e11c19 | started | master02 | https://192.168.50.11:2380 |
| https://192.168.50.11:2379 |      false |
| f9423fa20fccb5de | started | master01 | https://192.168.50.10:2380 |
| https://192.168.50.10:2379 |      false |
| f955e62306188c83 | started | master03 | https://192.168.50.12:2380 |
| https://192.168.50.12:2379 |      false |
+-----+-----+-----+
```

Kapitel 6 | Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

```
sh-4.4# etcdctl endpoint health --cluster
https://192.168.50.11:2379 is healthy: successfully committed proposal: took =
8.418937ms
https://192.168.50.10:2379 is healthy: successfully committed proposal: took =
8.842156ms
https://192.168.50.12:2379 is healthy: successfully committed proposal: took =
8.2554ms
sh-4.4# exit
exit
```

- 6.5. Vergewissern Sie sich, dass die drei etcd-Mitglieder für den OpenShift-Cluster verfügbar sind.

```
[lab@utility ~]$ oc get etcd -o=jsonpath='{range .items[0].status.conditions[?(@.type=="EtcdMembersAvailable")]}{.message}{"\n"}'
3 members are available
```



Anmerkung

Der etcd-Operator erkennt automatisch ein neues etcd-Mitglied und verbindet es mit dem Cluster. In diesem Szenario müssen Sie das etcd-Backup nicht verwenden.

Beenden

Löschen Sie nach Abschluss dieser angeleiteten Übung die gesamte Übungsumgebung, indem Sie im Red Hat Learning-Portal auf die Schaltfläche **DELETE** klicken.

Hiermit ist die angeleitete Übung beendet.

► Quiz

Kapitelwiederholung: Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche zwei Elemente sind erforderlich, um einen fehlgeschlagenen OpenShift Control Plane-Knoten zu ersetzen, wenn das etcd-Quorum nicht verloren gegangen ist?
(Wählen Sie zwei Antworten aus.)
- a. Die Ignition-Konfigurationsdatei
 - b. Ein etcd-Backup
 - c. SSH-Zugriff auf alle Control Planes im Cluster
 - d. Benutzer mit cluster-admin-Berechtigungen
 - e. Benutzer developer
- 2. Unter welchen zwei Umständen ist es von Vorteil, etcd-Backups durchzuführen?
(Wählen Sie zwei Antworten aus.)
- a. Alle sechs Monate.
 - b. Nachdem der Cluster installiert und für Endbenutzer freigegeben wurde.
 - c. Vor der Änderung einer Cluster-Konfiguration, die sich auf die Parameter des Kubelets auswirkt.
 - d. Regelmäßig und vor der Durchführung von Aktionen, die die Stabilität des OpenShift-Clusters gefährden können.
 - e. Nach dem Hinzufügen eines neuen Benutzers.
- 3. Bei welcher der folgenden Backup-Methoden handelt es sich um die effektivste Methode zum Speichern des OpenShift-Clusterstatus und der -konfiguration für Disaster Recovery-Zwecke?
- a. Ein Backup aller Disks der Control Plane-Knoten
 - b. Ein Backup der etcd-Datenbank
 - c. Ein Backup der Partition, auf der die statischen Pods der Control Plane-Knoten gehostet werden
 - d. Ein Backup der kubeconfig-Datei
- 4. Welche Aktion wird von Red Hat nach jeder OpenShift-Installation empfohlen?
- a. Einrichten von Cluster-Netzwerkrichtlinien
 - b. Hinzufügen einer benutzerdefinierten Zertifizierungsstelle
 - c. Installieren des MySQL-Operators
 - d. Speichern einer Kopie der vom Installationsprogramm generierten kubeconfig-Datei

► Lösung

Kapitelwiederholung: Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche zwei Elemente sind erforderlich, um einen fehlgeschlagenen OpenShift Control Plane-Knoten zu ersetzen, wenn das etcd-Quorum nicht verloren gegangen ist?
(Wählen Sie zwei Antworten aus.)
- a. Die Ignition-Konfigurationsdatei
 - b. Ein etcd-Backup
 - c. SSH-Zugriff auf alle Control Planes im Cluster
 - d. Benutzer mit cluster-admin-Berechtigungen
 - e. Benutzer developer
- 2. Unter welchen zwei Umständen ist es von Vorteil, etcd-Backups durchzuführen?
(Wählen Sie zwei Antworten aus.)
- a. Alle sechs Monate.
 - b. Nachdem der Cluster installiert und für Endbenutzer freigegeben wurde.
 - c. Vor der Änderung einer Cluster-Konfiguration, die sich auf die Parameter des Kubelets auswirkt.
 - d. Regelmäßig und vor der Durchführung von Aktionen, die die Stabilität des OpenShift-Clusters gefährden können.
 - e. Nach dem Hinzufügen eines neuen Benutzers.
- 3. Bei welcher der folgenden Backup-Methoden handelt es sich um die effektivste Methode zum Speichern des OpenShift-Clusterstatus und der -konfiguration für Disaster Recovery-Zwecke?
- a. Ein Backup aller Disks der Control Plane-Knoten
 - b. Ein Backup der etcd-Datenbank
 - c. Ein Backup der Partition, auf der die statischen Pods der Control Plane-Knoten gehostet werden
 - d. Ein Backup der kubeconfig-Datei
- 4. Welche Aktion wird von Red Hat nach jeder OpenShift-Installation empfohlen?
- a. Einrichten von Cluster-Netzwerkrichtlinien
 - b. Hinzufügen einer benutzerdefinierten Zertifizierungsstelle
 - c. Installieren des MySQL-Operators
 - d. Speichern einer Kopie der vom Installationsprogramm generierten kubeconfig-Datei

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Typische „Day 1“- und „Day 2“-Vorgänge, die in einem OpenShift-Cluster ausgeführt werden.
- Day 1-Vorgänge, die abgeschlossen werden müssen, bevor ein Cluster für die Produktion freigegeben wird.
- Konfigurieren eines dynamischen Storage-Anbieters
- Durchführen eines Funktionstests des Clusters
- Die Bedeutung regelmäßiger etcd-Backups
- Generieren eines etcd -Backups
- Ersetzen eines fehlgeschlagenen Control Plane-Knotens

Kapitel 7

Red Hat OpenShift Installation Lab - Ausführliche Wiederholung

Ziel

Wiederholen von Aufgaben aus *Red Hat OpenShift Installation Lab*

Ziele

- Wiederholen von Aufgaben aus *Red Hat OpenShift Installation Lab*

Abschnitte

- Installieren eines kompakten OpenShift-Clusters (und Lab)

Praktische Übung

Installieren eines kompakten OpenShift-Clusters

Ausführliche Wiederholung

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die in *Red Hat OpenShift Installation Lab* erworbenen Kenntnisse demonstrieren können.

Wiederholung Red Hat OpenShift Installation Lab

Bevor Sie mit der ausführlichen Wiederholung für diesen Kurs beginnen, sollten Sie mit den in den jeweiligen Kapiteln behandelten Themen vertraut sein.

Für zusätzliche Übungen stehen Ihnen auch die vorherigen Kapitel dieses Lehrbuchs zur Verfügung.

Kapitel 1, Beschreiben des OpenShift-Installationsprozesses

Beschreiben und Vergleichen der Installationsmethoden „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“

- Beschreiben und Vergleichen der Installationsmethoden „Full-Stack-Automatisierung“ und „Bereits vorhandene Infrastruktur“
- Beschreiben des OpenShift-Installationsprogramms und seiner Konfigurationsdateien
- Beschreiben der Unterschiede zwischen einem selbstverwalteten OpenShift-Cluster und gehosteten OpenShift-Angeboten

Kapitel 2, Installieren von OpenShift auf einer Cloud-Anbieter-Plattform

Bereitstellen von OpenShift-Clustern auf IaaS-Cloud-Anbieter-Plattformen mit allgemeinen Anpassungen unter Verwendung der Installationsmethode „Full-Stack-Automatisierung“

- Beschreiben der Architektur und des Workflows zur Installation von OpenShift auf einer IaaS-Cloud-Anbieter-Plattform mit der Full-Stack-Automatisierungsmethode
- Erfüllen der Voraussetzungen für die Installation von OpenShift auf Amazon Web Services (AWS) mit Full-Stack-Automatisierung
- Installieren von OpenShift auf Amazon Web Services (AWS) unter Verwendung von Full-Stack-Automatisierung mit allgemeinen Anpassungen
- Bewerten des Erfolgs einer Installation von OpenShift auf AWS

Kapitel 3, Installieren von OpenShift in einer virtualisierten Umgebung

Bereitstellen von OpenShift-Clustern auf Hypervisors mit allgemeinen Anpassungen unter Verwendung der Installationsmethoden „Full-Stack-Automatisierung“ und „Installation auf bereits vorhandener Infrastruktur“

- Beschreiben der Architektur und des Workflows zur Installation von OpenShift auf Hypervisors mit der Full-Stack-Automatisierungsmethode und der Methode bei bereits vorhandener Infrastruktur.
- Beschreiben der Installation von OpenShift auf vSphere mithilfe der Full-Stack-Automatisierung mit allgemeinen Anpassungen.
- Beschreiben der Installation von OpenShift auf vSphere mithilfe der Installation bei bereits vorhandener Infrastruktur mit allgemeinen Anpassungen.

Kapitel 4, Planen der Installation von OpenShift ohne einen Infrastrukturanbieter

Konfigurieren der Voraussetzungen für die Bereitstellung von OpenShift-Clustern ohne Integration in die zugrunde liegende Infrastruktur.

- Beschreiben der Architektur und des Workflows für die Installation von OpenShift ohne Integration in die zugrunde liegende Infrastruktur.
- Bereitstellen der Voraussetzungen für die Installation von OpenShift ohne Integration in die zugrunde liegende Infrastruktur.

Kapitel 5, Installieren von OpenShift ohne Infrastrukturanbieter

Bereitstellen von OpenShift-Clustern ohne Integration in die zugrunde liegende Infrastruktur.

Installieren Sie OpenShift, indem Sie den Bootstrap-Rechner und die Cluster-Knoten starten und den Fortschritt des Installationsvorgangs überwachen.

Kapitel 6, Abschließen der Installation von OpenShift ohne Infrastrukturanbieter

Durchführen wesentlicher Aufgaben, die vor dem Onboarding von Benutzern und Anwendungen in einem neu bereitgestellten OpenShift-Cluster erforderlich sind.

- Durchführen der erforderlichen Anpassungen, bevor Benutzer und Anwendungen in einen neu installierten Cluster integriert werden.
- Sichern und Wiederherstellen eines Control Plane-Knotens.

► Praktische Übung

Installieren eines kompakten OpenShift-Clusters

In dieser Wiederholung installieren Sie einen kompakten Cluster (drei Control Plane-Knoten) und überprüfen die Installation.

Ergebnisse

Sie sollten in der Lage sein, die Installation eines kompakten OpenShift-Clusters abzuschließen und die Installation zu überprüfen.

Bevor Sie Beginnen

Klicken Sie im Red Hat Learning-Portal auf **DELETE**, um die gesamte Übungsumgebung zu löschen.

Klicken Sie nach Löschen der Umgebung auf **CREATE**, und warten Sie, bis für alle Server **active** angezeigt wird.

Sobald für alle Server **active** angezeigt wird, klicken Sie auf **+**, um den Auto-Stop-Timer zu erweitern und das Intervall auf mindestens zwei Stunden festzulegen. Dadurch wird sichergestellt, dass Ihre Umgebung nicht mitten in der Übung angehalten wird. Führen Sie auf dem Rechner **workstation** als Benutzer **student** den Befehl **lab** aus, um Ihr System für diese Übung vorzubereiten.

Mit diesem Befehl wird sichergestellt, dass die zum Abschließen der ausführlichen Wiederholung erforderlichen Dateien auf dem Rechner „**workstation**“ verfügbar sind.

```
[student@workstation ~]$ lab comprehensive-review start
```



Anmerkung

Die Firefox-Version auf dem Rechner „**workstation**“ rendert die Website <https://cloud.redhat.com/openshift/> nicht ordnungsgemäß. Um diese angeleitete Übung durchzuführen, installieren und verwenden Sie den Chromium-Webbrowser auf dem Rechner „**workstation**“:

```
[student@workstation ~]$ sudo yum install chromium
```

Anweisungen

1. Ermitteln Sie den FQDN der lokalen Registry für die Region Ihrer Kursumgebung. Sie können die Region Ihrer Übungsumgebung auf der Registerkarte **Lab Environment** auf der Webseite des Kurses "Red Hat OpenShift Installation Lab" finden.

In der folgenden Tabelle wird der FQDN der lokalen Registry für die einzelnen Regionen aufgeführt.

FQDN der lokalen Registry nach Kursumgebungsregion

Region	FQDN der lokalen Registry
Nordamerika	nexus-registry-int.apps.tools-na.prod.nextcle.com
EMEA	nexus-registry-int.apps.tools-eu.prod.nextcle.com
APAC	nexus-registry-int.apps.tools-ap.prod.nextcle.com

In dieser Übung verwenden Sie eine lokale Registry, die die aus `quay.io` gespiegelten OpenShift-Release-Images enthält. Die lokale Registry wird in derselben Region wie die Kursumgebung ausgeführt, und ihr vollständig qualifizierter Domain-Name (FQDN) hängt von dieser Region ab.

- Erstellen Sie ein gültiges Pull Secret für die Installation von OpenShift in der Kursumgebung, indem Sie ein Pull Secret von `cloud.redhat.com` verwenden. Die Anmelddaten für die Registry `quay.io` müssen durch die Anmelddaten für Ihre lokale Registry ersetzt werden.

Angenommen, die Region Ihrer Kursumgebung ist `northamerica`. In diesem Fall lautet der Inhalt des `pull-secret` für die lokale Registry wie folgt:

```
"nexus-registry-int.apps.tools-na.prod.nextcle.com":  
{ "auth": "cmVndXNlcjpJbnN0YWxsTTM=", "email": "nobody@example.com"}
```

Kopieren Sie zum Schluss die Pull Secret-Datei als Benutzer `lab` auf den Server `utility`.

- Führen Sie das Ansible-Playbook im Ordner `/home/student/D0322/labs/install-compreview/ansible/` unter Verwendung der in diesem Ordner enthaltenen Inventardatei aus.
- Erstellen Sie auf dem Server `utility` einen SSH-Schlüssel. Speichern Sie ihn in der Datei `/home/lab/.ssh/ocp4upi`.
- Vervollständigen Sie die Datei `install-config.yaml` im Ordner `/home/lab/ocp4upi/`. Verwenden Sie das an anderer Stelle in dieser Übung generierte Pull Secret. Verwenden Sie den SSH-Schlüssel, den Sie an anderer Stelle in dieser Übung erstellt haben. Schließen Sie die `imageContentSources`-Informationen ein, um die lokale Registry zum Abrufen der OpenShift-Release-Images zu verwenden. Es dürfen nur drei Control Plane-Knoten ohne Server-Knoten vorhanden sein.
- Erstellen Sie die Kubernetes-Manifeste und Ignition-Konfigurationsdateien mit dem CLI-Tool `openshift-install`. Sie müssen das Installationsprogramm so konfigurieren, dass die lokale Kurs-Registry verwendet wird, um die OpenShift-Release-Version abzurufen. Exportieren Sie dazu die folgende Variable:

```
OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE=<LOCAL_REGISTRY_FQDN>/openshift/  
ocp4:4.6.4-x86_64"
```

Legen Sie die Ignition-Dateien, die von jedem gelesen werden können, in `/var/www/html/openshift4/4.6.4/ignitions/` ab.

- Installieren Sie über das PXE-Menü RHCOS auf den Rechnern `bootstrap`, `master01`, `master02` und `master03`, um die OpenShift-Cluster-Bereitstellung zu starten.

8. Konfigurieren Sie den persistenten Storage für die lokale Registry mit dem NFS-Server, der auf dem Server **utility** ausgeführt wird.
Verwenden Sie den neuen Ordner **registry** im Ordner **NFS /exports** auf dem Server **utility**. Die Ordnerberechtigungen müssen 777 lauten.
9. Überprüfen der Integrität des Clusters.
10. Stellen Sie eine **rails-postgresql-example**-Anwendung im **test**-Projekt zum Durchführen eines Funktionstests des Clusters bereit.

Bewertung

Verwenden Sie als Benutzer **student** auf dem Rechner **workstation** den Befehl **lab**, um Ihre Arbeit zu bewerten. Beheben Sie sämtliche gemeldeten Fehler, und führen Sie den Befehl so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[lab@utility ~]$ exit  
...output omitted...  
[student@workstation ~]$ lab comprehensive-review grade
```

Beenden

Führen Sie auf dem Rechner **workstation** als Benutzer **student** den Befehl **lab** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab comprehensive-review finish
```



Anmerkung

Mit dem Befehl **lab comprehensive-review finish** wird der Cluster nicht entfernt. Das Ausführen des Skripts **lab comprehensive-review grade** nach dem Befehl **lab comprehensive-review finish** zeigt an, dass der Cluster mit drei Knoten vorhanden ist.

Wenn Sie von vorne beginnen möchten, setzen Sie die Rechner **bootstrap**, **master01**, **master02** und **master03** zurück, und starten Sie dann erneut mit dem Schritt, der Sie anweist, RHCOS auf den Rechnern „bootstrap“, „master01“, „master02“ und „master03“ über das PXE-Menü zu installieren.

Hiermit ist die praktische Übung beendet.

► Lösung

Installieren eines kompakten OpenShift-Clusters

In dieser Wiederholung installieren Sie einen kompakten Cluster (drei Control Plane-Knoten) und überprüfen die Installation.

Ergebnisse

Sie sollten in der Lage sein, die Installation eines kompakten OpenShift-Clusters abzuschließen und die Installation zu überprüfen.

Bevor Sie Beginnen

Klicken Sie im Red Hat Learning-Portal auf **DELETE**, um die gesamte Übungsumgebung zu löschen.

Klicken Sie nach Löschen der Umgebung auf **CREATE**, und warten Sie, bis für alle Server **active** angezeigt wird.

Sobald für alle Server **active** angezeigt wird, klicken Sie auf **+**, um den Auto-Stop-Timer zu erweitern und das Intervall auf mindestens zwei Stunden festzulegen. Dadurch wird sichergestellt, dass Ihre Umgebung nicht mitten in der Übung angehalten wird. Führen Sie auf dem Rechner **workstation** als Benutzer **student** den Befehl **lab** aus, um Ihr System für diese Übung vorzubereiten.

Mit diesem Befehl wird sichergestellt, dass die zum Abschließen der ausführlichen Wiederholung erforderlichen Dateien auf dem Rechner „**workstation**“ verfügbar sind.

```
[student@workstation ~]$ lab comprehensive-review start
```



Anmerkung

Die Firefox-Version auf dem Rechner „**workstation**“ rendert die Website <https://cloud.redhat.com/openshift/> nicht ordnungsgemäß. Um diese angeleitete Übung durchzuführen, installieren und verwenden Sie den Chromium-Webbrowser auf dem Rechner „**workstation**“:

```
[student@workstation ~]$ sudo yum install chromium
```

Anweisungen

- Ermitteln Sie den FQDN der lokalen Registry für die Region Ihrer Kursumgebung. Sie können die Region Ihrer Übungsumgebung auf der Registerkarte **Lab Environment** auf der Webseite des Kurses "Red Hat OpenShift Installation Lab" finden.

In der folgenden Tabelle wird der FQDN der lokalen Registry für die einzelnen Regionen aufgeführt.

FQDN der lokalen Registry nach Kursumgebungsregion

Region	FQDN der lokalen Registry
Nordamerika	nexus-registry-int.apps.tools-na.prod.nextcle.com
EMEA	nexus-registry-int.apps.tools-eu.prod.nextcle.com
APAC	nexus-registry-int.apps.tools-ap.prod.nextcle.com

In dieser Übung verwenden Sie eine lokale Registry, die die aus `quay.io` gespiegelten OpenShift-Release-Images enthält. Die lokale Registry wird in derselben Region wie die Kursumgebung ausgeführt, und ihr vollständig qualifizierter Domain-Name (FQDN) hängt von dieser Region ab.

- 1.1. Installieren Sie den Chromium-Webbrowser über einen Terminal auf dem Rechner `workstation`.

```
[student@workstation ~]$ sudo yum install chromium
...output omitted...
```

Wenn Sie dazu aufgefordert werden, geben Sie `yes` ein und drücken Sie die **Eingabetaste**, um die Installation zu akzeptieren.

- 1.2. Ermitteln Sie die Region Ihrer Kursumgebung.

Navigieren Sie auf dem Rechner `workstation` mit dem Chromium-Webbrowser zur Registerkarte **Lab Environment** auf der Webseite des Kurses „Red Hat OpenShift Installation Lab“.

Klicken Sie auf **Information**, um Informationen zur Kursumgebung anzuzeigen. Die Kursumgebungsregion finden Sie im Feld **Published region**.

The screenshot shows the 'Lab Environment' tab of the Red Hat OpenShift Installation Lab interface. It lists three virtual machines: bastion, bootstrap, and classroom, all currently stopped. A tooltip for the 'bastion' row provides specific details about the lab setup:

- Project id: 2ebdcefb42a144ada7e8cfc3fcbe8b56
- Project name: ole-3030f6f7-9253-47ed-9ea1-73c0aa7e62c9
- Project state: stopped
- Lab definition id: do322ea-4.6
- Published region: northamerica (highlighted in red)
- openstack region: us-east-1

- 1.3. Suchen Sie anhand der vorherigen Tabelle nach dem FQDN der lokalen Registry für Ihre Region.
- 1.4. Überprüfen Sie als Benutzer `lab` auf dem Server `utility` mit dem Befehl `curl`, ob Sie mit dem FQDN der lokalen Registry für Ihre Kursumgebung kommunizieren können.

Der folgende Befehl geht davon aus, dass die Kursumgebungsregion `northamerica` ist, die als FQDN der lokalen Registry `nexus-registry-int.apps.tools-na.prod.nextcle.com` verwendet.

```
[student@workstation ~]$ ssh lab@utility
```

```
[lab@utility ~]$ curl -s \  
> https://nexus-registry-int.apps.tools-na.prod.nextcle.com -o /dev/null; echo $?  
0
```

Wenn Sie den falschen FQDN für die lokalen Registry verwendet haben, ist die Ausgabe des Befehls `curl` nicht 0.



Anmerkung

Die lokale Registry der Region `northamerica` (`nexus-registry-int.apps.tools-na.prod.nextcle.com`) wird in den folgenden Schritten verwendet.

Stellen Sie sicher, dass Sie den FQDN der lokalen Registry der Region Ihrer Kursumgebung verwenden. Wenn Sie eine lokale Registry aus anderen Regionen verwenden, schlagen die angeleiteten Übungen fehl.

2. Erstellen Sie ein gültiges Pull Secret für die Installation von OpenShift in der Kursumgebung, indem Sie ein Pull Secret von `cloud.redhat.com` verwenden. Die Anmelde Daten für die Registry `quay.io` müssen durch die Anmelde Daten für Ihre lokale Registry ersetzt werden.
Angenommen, die Region Ihrer Kursumgebung ist `northamerica`. In diesem Fall lautet der Inhalt des `pull-secret` für die lokale Registry wie folgt:

```
"nexus-registry-int.apps.tools-na.prod.nextcle.com":  
{ "auth": "cmVndXNlcjpJbnN0YWxsTTM=", "email": "nobody@example.com" }
```

Kopieren Sie zum Schluss die Pull Secret-Datei als Benutzer `lab` auf den Server `utility`.

- 2.1. Rufen Sie ein Pull Secret von `cloud.redhat.com` ab.

Navigieren Sie auf dem Rechner `workstation` mit dem Chromium-Webbrowser zu `https://cloud.redhat.com/openshift/install/metal/user-provisioned`. Melden Sie sich mit den Anmelde Daten Ihres Red Hat-Benutzerkontos an.

Klicken Sie auf **Download pull secret**. Wählen Sie dann **Save File** aus, und klicken Sie auf **OK**.

Die Datei wird unter `/home/student/Downloads/pull-secret` gespeichert.

- 2.2. Formatieren Sie das Pull Secret als JSON-Datei.

Öffnen Sie auf dem Rechner `workstation` ein Terminal, und formatieren Sie die `pull-secret`-Datei im Ordner `/home/student/Downloads` als JSON-Objekt.

```
[lab@utility ~]$ exit  
...output omitted...  
[student@workstation ~]$
```

```
[student@workstation ~]$ python3 -m json.tool \
> Downloads/pull-secret > pull-secret.json

[student@workstation ~]$ cat pull-secret.json
{
    "auths": {
        "cloud.openshift.com": {
            "auth": "UxUUjYwSTMyb3BlUXUUjYwSTMybnNUxUUjYwSTM...YMy3NfNGUXUUjYw==",
            "email": "student@redhat.com"
        },
        "quay.io": {
            "auth": "UxUUjYwSTMyb3BlUXUUjYwSTMybnNUxUUjYwSTM...YMy3NfNGUXUUjYw==",
            "email": "student@redhat.com"
        },
        "registry.connect.redhat.com": {
            "auth": "CvmsWaJUROSkhCkEQ71NiM1BsracE9ZOVmBMIDrS3R20K0H8Eq...Tx09ph...",
            "email": "student@redhat.com"
        },
        "registry.redhat.io": {
            "auth": "CvmsWaJUROSkhCkEQ71NiM1BsracE9ZOVmBMIDrS3R20K0H8Eq...Tx09ph...",
            "email": "student@redhat.com"
        }
    }
}
```

- 2.3. Bearbeiten Sie die Datei `pull-secret.json`, indem Sie die Anmelddaten für `quay.io` durch die Anmelddaten für Ihre lokale Registry ersetzen.

```
[student@workstation ~]$ vi pull-secret.json
{
    "auths": {
        "cloud.openshift.com": {
            "auth": "UxUUjYwSTMyb3BlUXUUjYwSTMybnNUxUUjYwSTM...YMy3NfNGUXUUjYw==",
            "email": "student@redhat.com"
        },
        "nexus-registry-int.apps.tools-na.prod.nextcle.com": {
            "auth": "cmVndXNlcjpJbnN0YWxsTTM=",
            "email": "nobody@example.com"
        },
        "registry.connect.redhat.com": {
            "auth": "CvmsWaJUROSkhCkEQ71NiM1BsracE9ZOVmBMIDrS3R20K0H8Eq...Tx09ph...",
            "email": "student@redhat.com"
        },
        "registry.redhat.io": {
            "auth": "CvmsWaJUROSkhCkEQ71NiM1BsracE9ZOVmBMIDrS3R20K0H8Eq...Tx09ph...",
            "email": "student@redhat.com"
        }
    }
}
```

```
        "email": "student@redhat.com"
    }
}
```

Speichern Sie die Datei, und schließen Sie den Editor.

- 2.4. Generieren Sie mithilfe des Tools jq die kompakte Version der Datei `pull-secret.json` in der Datei `pull-secret-oneline.json`.

```
[student@workstation ~]$ cat pull-secret.json | jq . -c > pull-secret-oneline.json
```



Anmerkung

Mit der im vorherigen Befehl verwendeten jq-Analyse wird zudem überprüft, ob die Datei `pull-secret.json` gültig ist. Wenn beim Ausführen dieses Befehls keine Fehler auftreten, sind die Dateien `pull-secret.json` und `pull-secret-oneline.json` gültig.

```
[student@workstation ~]$ cat pull-secret-oneline.json
{"auths":{"cloud.openshift.com":
 {"auth":"UxUUj...UUjYw==","email":"student@redhat.com"},"nexus-
 registry-int.apps.tools-na.prod.nextcle.com":
 {"auth":"cmVnd...STTM=","email":"nobody@example.com"},"registry.connect.redhat.com":
 {"auth":"CvmsW...qKLJN=","email":"student@redhat.com"},"registry.redhat.io":
 {"auth":"CvmsW...qKLJN=","email":"student@redhat.com"}}}
```

- 2.5. Kopieren Sie die Datei `pull-secret-oneline.json` auf den Server `utility`.

```
[student@workstation ~]$ scp pull-secret-oneline.json lab@utility:
...output omitted...
pull-secret-oneline.json      100% 2624     4.6MB/s   00:00
```

Sie werden die Datei `pull-secret-oneline.json` später in diesem Lab verwenden, um die Datei `install-config.yaml` zu vervollständigen.

3. Führen Sie das Ansible-Playbook im Ordner `/home/student/D0322/labs/install-compreview/ansible/` unter Verwendung der in diesem Ordner enthaltenen Inventardatei aus.

- 3.1. Überprüfen Sie den Inhalt des Ordners `/home/student/D0322/`.

```
[student@workstation ~]$ tree D0322/
D0322/
└── labs
    └── comprehensive-review
        └── ansible
            ├── inventory
            ├── prereq.yaml
            └── roles
                └── ocp_install_prereq
                    ├── files
                    └── haproxy.cfg
```

```
|   └── install-config.yaml  
└── tasks  
    └── main.yaml  
└── vars  
    └── main.yml
```

- 3.2. Wechseln Sie zum Ordner /home/student/D0322/labs/comprehensive-review/ansible/.

```
[student@workstation ~]$ cd D0322/labs/comprehensive-review/ansible/  
[student@workstation ansible]$
```

- 3.3. Führen Sie das Ansible-Playbook prereq.yaml aus. Verwenden Sie die im selben Ordner enthaltene Inventardatei.

```
[student@workstation ansible]$ ansible-playbook -i inventory prereq.yaml  
  
PLAY [Configure Prerequisites]  
*****  
*****  
  
TASK [ocp_install_prereq : Download "oc"]  
*****  
*****  
changed: [utility.lab.example.com]  
  
TASK [ocp_install_prereq : Download "openshift-install"]  
*****  
*****  
changed: [utility.lab.example.com]  
  
TASK [ocp_install_prereq : Extract "oc" to /usr/bin/]  
*****  
*****  
changed: [utility.lab.example.com]  
  
TASK [ocp_install_prereq : Extract "openshift-install" to /usr/bin/]  
*****  
*****  
changed: [utility.lab.example.com]  
  
TASK [ocp_install_prereq : Download RHCOS rootfs]  
*****  
*****  
changed: [utility.lab.example.com]  
  
TASK [ocp_install_prereq : Download RHCOS kernel]  
*****  
*****  
changed: [utility.lab.example.com]  
  
TASK [ocp_install_prereq : Download RHCOS initramfs]  
*****  
*****
```

```
changed: [utility.lab.example.com]

TASK [ocp_install_prereq : Download bootstrap PXE Boot file]
*****
*****
changed: [utility.lab.example.com]

TASK [ocp_install_prereq : Download master01 PXE boot file]
*****
*****
changed: [utility.lab.example.com]

TASK [ocp_install_prereq : Download master02 PXE Boot file]
*****
*****
changed: [utility.lab.example.com]

TASK [ocp_install_prereq : Download master03 PXE Boot file]
*****
*****
changed: [utility.lab.example.com]

TASK [ocp_install_prereq : Configure haproxy.cfg file]
*****
*****
changed: [utility.lab.example.com]

TASK [ocp_install_prereq : Reload HAProxy service]
*****
*****
changed: [utility.lab.example.com]

TASK [ocp_install_prereq : Create the OpenShift
installation directory if it does not exist]
*****
changed: [utility.lab.example.com]

TASK [ocp_install_prereq : Create custom install-config.yaml file]
*****
*****
changed: [utility.lab.example.com]
```

PLAY RECAP

```
*****
utility.lab.example.com      : ok=15    changed=15    unreachable=0    failed=0
  skipped=0    rescued=0    ignored=0
```

Wie Sie sehen können, führt dieses Playbook mehrere erforderliche Aufgaben in `utility.lab.example.com` aus.

- Es lädt die CLI-Tools `oc` und `openshift-install` herunter und installiert sie
- Es lädt RHCOS-Dateien herunter
- Es lädt PXE-Boot-Dateien für den Bootstrap-Knoten und die Control Plane-Knoten herunter
- Es konfiguriert den HAProxy-Load Balancer für den API- und ingress -Datenverkehr
- Es lädt die Basisdatei `install-config.yaml` herunter (die Sie später vervollständigen)

3.4. Kehren Sie zum Ordner `home` zurück, der zum Benutzer `student` gehört.

```
[student@workstation ansible]$ cd ~
[student@workstation ~]$
```

4. Erstellen Sie auf dem Server `utility` einen SSH-Schlüssel. Speichern Sie ihn in der Datei `/home/lab/.ssh/ocp4upi`.

4.1. Stellen Sie eine Verbindung zum Server `utility` über SSH als Benutzer `lab` her.

```
[student@workstation ~]$ ssh lab@utility
...output omitted...
[lab@utility ~]$
```

4.2. Generieren Sie den SSH-Schlüssel, und speichern Sie ihn dann in der Datei `/home/lab/.ssh/ocp4upi`.

```
[lab@utility ~]$ ssh-keygen -t rsa -b 4096 -N '' -f .ssh/ocp4upi
Generating public/private rsa key pair.
Your identification has been saved in .ssh/ocp4upi.
Your public key has been saved in .ssh/ocp4upi.pub.
...output omitted...
[lab@utility ~]$
```

Der öffentliche Schlüssel, den Sie später verwenden, befindet sich in der Datei `/home/lab/.ssh/ocp4upi.pub`.

5. Vervollständigen Sie die Datei `install-config.yaml` im Ordner `/home/lab/ocp4upi/`. Verwenden Sie das an anderer Stelle in dieser Übung generierte Pull Secret. Verwenden Sie den SSH-Schlüssel, den Sie an anderer Stelle in dieser Übung erstellt haben. Schließen Sie die `imageContentSources`-Informationen ein, um die lokale Registry zum Abrufen der OpenShift-Release-Images zu verwenden. Es dürfen nur drei Control Plane-Knoten ohne Server-Knoten vorhanden sein.

- 5.1. Bearbeiten Sie die Datei `install-config.yaml` im Ordner `/home/lab/ocp4upi/`. Verwenden Sie das kompakte Pull Secret in der Datei `pull-secret-oneline.json`.

```
[lab@utility ~]$ vi ocp4upi/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 0
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: ocp4
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
fips: false
pullSecret: |
  {"auths":{"cloud.openshift.com": {
    "auth":"UxUUj...UUjYw==", "email":"student@redhat.com"}, "nexus-registry-int.apps.tools-na.prod.nextcle.com": {
    "auth":"cmVnd...sTTM=", "email":"nobody@example.com"}, "registry.connect.redhat.com": {
    "auth":"CvmsW...qKLJN=", "email":"student@redhat.com"}, "registry.redhat.io": {
    "auth":"CvmsW...qKLJN=", "email":"student@redhat.com"}}}
  sshKey: |
    <CHANGE_ME_KEEPING_THE_INDENTATION_LEVEL>
imageContentSources:
- mirrors:
  - nexus-registry-int.apps.tools-na.prod.nextcle.com/openshift/ocp4
    source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - nexus-registry-int.apps.tools-na.prod.nextcle.com/openshift/ocp4
    source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

Speichern Sie die Datei, und schließen Sie den Editor.



Anmerkung

Die Anzahl der Replikate der Server-Rechner ist 0. Dadurch können die Control Plane-Knoten automatisch geplant werden.

- 5.2. Kopieren Sie den öffentlichen SSH-Schlüssel aus der Datei `/home/lab/.ssh/ocp4upi.pub`. Bearbeiten Sie die Datei `install-config.yaml` erneut, um den Schlüssel im Abschnitt `sshKey` hinzuzufügen.

```
[lab@utility ~]$ cat .ssh/ocp4upi.pub  
ssh-rsa AAAA...UgUsz2w== lab@utility.lab.example.com
```

```
[lab@utility ~]$ vi ocp4upi/install-config.yaml  
apiVersion: v1  
baseDomain: example.com  
compute:  
- hyperthreading: Enabled  
  name: worker  
  replicas: 0  
controlPlane:  
  hyperthreading: Enabled  
  name: master  
  replicas: 3  
metadata:  
  name: ocp4  
networking:  
  clusterNetwork:  
  - cidr: 10.128.0.0/14  
    hostPrefix: 23  
  networkType: OpenShiftSDN  
  serviceNetwork:  
  - 172.30.0.0/16  
platform:  
  none: {}  
fips: false  
pullSecret: |  
  {"auths":{"cloud.openshift.com":  
{"auth":"UxUUj...UUjYw==","email":"student@redhat.com"},"nexus-  
registry-int.apps.tools-na.prod.nextcle.com":  
{"auth":"cmVnd...STTM=","email":"nobody@example.com"}, "registry.connect.redhat.com":  
{"auth":"CvmsW...qKLJN=","email":"student@redhat.com"}, "registry.redhat.io":  
{"auth":"CvmsW...qKLJN=","email":"student@redhat.com"}}}  
sshKey: |  
  ssh-rsa AAAA...UgUsz2w== lab@utility.lab.example.com  
imageContentSources:  
- mirrors:  
  - nexus-registry-int.apps.tools-na.prod.nextcle.com/openshift/ocp4  
    source: quay.io/openshift-release-dev/ocp-release  
- mirrors:  
  - nexus-registry-int.apps.tools-na.prod.nextcle.com/openshift/ocp4  
    source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

Speichern Sie die Datei, und schließen Sie den Editor.

- 5.3. Erstellen Sie zu Fehlerbehebungszwecken eine Kopie der Datei „install-config.yaml“ im Ordner /home/lab/.

```
[lab@utility ~]$ cp ocp4upi/install-config.yaml .
```

6. Erstellen Sie die Kubernetes-Manifeste und Ignition-Konfigurationsdateien mit dem CLI-Tool `openshift-install`. Sie müssen das Installationsprogramm so konfigurieren, dass

die lokale Kurs-Registry verwendet wird, um die OpenShift-Release-Version abzurufen. Exportieren Sie dazu die folgende Variable:

```
OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE=<LOCAL_REGISTRY_FQDN>/openshift/
ocp4:4.6.4-x86_64"
```

Legen Sie die Ignition-Dateien, die von jedem gelesen werden können, in /var/www/html/openshift4/4.6.4/ignitions/ ab.

6.1. Legen Sie die Umgebungsvariable

OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE so fest, dass die lokale Registry für Ihre Cluster-Region verwendet wird. Das Release-Image befindet sich unter dem Pfad openshift/ocp4:4.6.4-x86_64.

```
[lab@utility ~]$ reg="nexus-registry-int.apps.tools-na.prod.nextcle.com"
[lab@utility ~]$ releaseimg="/openshift/ocp4:4.6.4-x86_64"
[lab@utility ~]$ export OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE=$reg$releaseimg
```

Verifizieren Sie den vollständigen URL-Pfad zum Release-Image wie folgt:

```
[lab@utility ~]$ echo $OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE
nexus-registry-int.apps.tools-na.prod.nextcle.com/openshift/ocp4:4.6.4-x86_64
```



Anmerkung

Die obigen Befehle setzen voraus, dass sich Ihr Cluster in der Region northamerica befindet. Verwenden Sie den FQDN der lokalen Registry für Ihre Region.

6.2. Generieren der Kubernetes-Manifeste

```
[lab@utility ~]$ openshift-install create manifests --dir=./ocp4upi
INFO Consuming Install Config from target directory
WARNING Making control-plane schedulable by setting MastersSchedulable to true for
Scheduler cluster settings
INFO Manifests created in: ocp4upi/manifests and ocp4upi/openshift
```

6.3. Generieren der Ignition-Konfigurationsdateien

```
[lab@utility ~]$ openshift-install create ignition-configs --dir=./ocp4upi
WARNING Found override for release image. Please be warned, this is not advised
INFO Consuming Master Machines from target directory
INFO Consuming Common Manifests from target directory
INFO Consuming OpenShift Install (Manifests) from target directory
INFO Consuming Worker Machines from target directory
INFO Consuming Openshift Manifests from target directory
INFO Ignition-Configs created in: ocp4upi and ocp4upi/auth
```

6.4. Kopieren Sie die Ignition-Dateien in den Ordner /var/www/html/openshift4/4.6.4/ignitions/. Legen Sie die Dateiberechtigungen so fest, damit sie von jedem Benutzer gelesen werden können.

```
[lab@utility ~]$ sudo cp ./ocp4upi/*.ign /var/www/html/openshift4/4.6.4/ignitions/  
[lab@utility ~]$ sudo chmod +r /var/www/html/openshift4/4.6.4/ignitions/*.ign
```

Überprüfen Sie die Leseberechtigungen der Ignition-Konfigurationsdateien.

```
[lab@utility ~]$ ls -l /var/www/html/openshift4/4.6.4/ignitions/  
total 296  
-rw-r--r--. 1 root root 293214 Feb 12 05:37 bootstrap.ign  
-rw-r--r--. 1 root root 1718 Feb 12 05:37 master.ign  
-rw-r--r--. 1 root root 1718 Feb 12 05:37 worker.ign
```

7. Installieren Sie über das PXE-Menü RHCOS auf den Rechnern `bootstrap`, `master01`, `master02` und `master03`, um die OpenShift-Cluster-Bereitstellung zu starten.
 - 7.1. Starten Sie den `bootstrap`-Rechner über die Konsole des Red Hat Learning-Portals neu, indem Sie in der oberen rechten Ecke auf `Ctrl Alt Del` klicken.
 - 7.2. Warten Sie, bis der Neustart abgeschlossen ist und der Rechner mit dem PXE-Menü bootet.
 - 7.3. Drücken Sie im PXE-Menü die **Eingabetaste**, um mit der Installation des Red Hat Enterprise Linux CoreOS auf dem Rechner fortzufahren.
 - 7.4. Wiederholen Sie diese Schritte, um `master01`, `master02` und `master03` zu installieren.
 - 7.5. Führen Sie zu Informationszwecken den Befehl `openshift-install` aus, um auf die Bootstrap-Installation zu warten.

```
[lab@utility ~]$ openshift-install --dir=./ocp4upi wait-for bootstrap-complete  
INFO Waiting up to 20m0s for the Kubernetes API at https://  
api.ocp4.example.com:6443...  
INFO API v1.19.0+9f84db3 up
```



Anmerkung

Sobald Sie den vorherigen Schritt ausgeführt haben, fahren Sie mit den nächsten Schritten im Lab fort. Es dauert etwa 10 Minuten, bis der vorherige Befehl abgeschlossen ist.

- 7.6. Öffnen Sie auf dem Rechner `workstation` ein neues Terminalfenster, und stellen Sie über SSH als Benutzer `lab` eine Verbindung zum Server `utility` her.

```
[student@workstation ~]$ ssh lab@utility  
...output omitted...  
[lab@utility ~]$
```

- 7.7. Sehen Sie sich die Ausgabe des Befehls `openshift-install` an, und warten Sie, bis die API ausgeführt wird.
Konfigurieren Sie anschließend die Umgebungsvariable `KUBECONFIG` für die Interaktion mit dem Cluster.

```
[lab@utility ~]$ export KUBECONFIG=~/ocp4upi/auth/kubeconfig
```

- 7.8. Beobachten Sie die Erstellung und Konfiguration der Control Plane-Knoten mit dem Befehl `watch oc get nodes`.



Anmerkung

Es kann einige Minuten dauern, bis die Control Plane-Knoten erstellt werden.

```
[lab@utility ~]$ watch oc get nodes
```

Der Status in der Ausgabe wird standardmäßig alle zwei Sekunden aktualisiert.

Wenn Sie den Befehl `watch` zum ersten Mal ausführen, werden keine Knoten angezeigt.

Nach ein paar Minuten wird in der Ausgabe Folgendes angezeigt:

```
Every 2.0s: oc get nodes
utility.lab.example.com: Wed Feb  3 08:10:51 2021

NAME      STATUS     ROLES      AGE      VERSION
master01  NotReady  master     5s       v1.19.0+9f84db3
master02  NotReady  master     12s      v1.19.0+9f84db3
master03  NotReady  master     13s      v1.19.0+9f84db3
```

Die Control Plane-Knoten wurden erstellt, sind aber noch nicht bereit.

Nach ein paar weiteren Minuten wird in der Ausgabe der aktualisierte Status angezeigt.

```
Every 2.0s: oc get nodes
utility.lab.example.com: Wed Feb  3 08:11:33 2021

NAME      STATUS     ROLES      AGE      VERSION
master01  Ready     master     47s      v1.19.0+9f84db3
master02  Ready     master     54s      v1.19.0+9f84db3
master03  Ready     master     55s      v1.19.0+9f84db3
```

Die Control Plane-Knoten sind noch nicht als „schedulable“ konfiguriert. Schließlich sieht die Ausgabe wie folgt aus:

```
Every 2.0s: oc get nodes
utility.lab.example.com: Wed Feb  3 08:11:59 2021

NAME      STATUS     ROLES           AGE      VERSION
master01  Ready     master,worker  73s      v1.19.0+9f84db3
master02  Ready     master,worker  80s      v1.19.0+9f84db3
master03  Ready     master,worker  81s      v1.19.0+9f84db3
```

Die Control Plane-Knoten sind an dieser Stelle ordnungsgemäß als „schedulable“ konfiguriert.



Anmerkung

Die CSRs für die Control Plane-Knoten werden während der Installation automatisch genehmigt.

- 7.9. Sobald die Control Plane-Knoten Ready und schedulable sind, zeigt der Befehl `openshift-install` die folgende Ausgabe an und wird dann beendet.

```
INFO Waiting up to 20m0s for the Kubernetes API at https://api.ocp4.example.com:6443...
INFO API v1.19.0+9f84db3 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
INFO Time elapsed: 8m1s
```

Verwenden Sie den Befehl `openshift-install`, um die restliche Installation zu überwachen.

```
[lab@utility ~]$ openshift-install --dir=./ocp4upi \
> wait-for install-complete --log-level=debug
DEBUG OpenShift Installer 4.6.4
DEBUG Built from commit 6e02d049701437fa81521fe981405745a62c86c5
DEBUG Loading Install Config...
DEBUG   Loading SSH Key...
DEBUG   Loading Base Domain...
DEBUG     Loading Platform...
DEBUG   Loading Cluster Name...
DEBUG     Loading Base Domain...
DEBUG     Loading Platform...
DEBUG   Loading Pull Secret...
DEBUG   Loading Platform...
DEBUG Using Install Config loaded from state file
INFO Waiting up to 40m0s for the cluster at https://api.ocp4.example.com:6443 to initialize...
DEBUG Still waiting for the cluster to initialize: Working towards 4.6.4: 99% complete
DEBUG Still waiting for the cluster to initialize: Multiple errors are preventing progress:
* Cluster operator authentication is reporting a failure:
  WellKnownReadyControllerDegraded: kube-apiserver oauth endpoint https://192.168.50.12:6443/.well-known/oauth-authorization-server is not yet served and authentication operator keeps waiting (check kube-apiserver operator, and check that instances roll out successfully, which can take several minutes per instance)
* Cluster operator monitoring is reporting a failure: Failed to rollout the stack.
  Error: running task Updating Prometheus-k8s failed: reconciling Prometheus rules PrometheusRule failed: updating PrometheusRule object failed: Internal error occurred: failed calling webhook "prometheusrules.openshift.io": Post "https://prometheus-operator.openshift-monitoring.svc:8080/admission-prometheusrules/validate?timeout=5s": x509: certificate signed by unknown authority
DEBUG Still waiting for the cluster to initialize: Cluster operator authentication is reporting a failure: WellKnownReadyControllerDegraded: need at least 3 kube-apiservers, got 2
```

```
DEBUG Still waiting for the cluster to initialize: Cluster operator authentication
is reporting a failure: WellKnownReadyControllerDegraded: need at least 3 kube-
apiservers, got 2
DEBUG Still waiting for the cluster to initialize: Cluster operator authentication
is reporting a failure: WellKnownReadyControllerDegraded: need at least 3 kube-
apiservers, got 2
DEBUG Cluster is initialized
INFO Waiting up to 10m0s for the openshift-console route to be created...
DEBUG Route found in openshift-console namespace: console
DEBUG Route found in openshift-console namespace: downloads
DEBUG OpenShift console route is created
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/lab/ocp4upi/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.ocp4.example.com
INFO Login to the console with user: "kubeadmin", and password: "rTbDK-j5rqe-
JQKVZ-ftEGH"
DEBUG Time elapsed per stage:
DEBUG Cluster Operators: 12m44s
INFO Time elapsed: 12m44s
```



Anmerkung

Es dauert etwa 20 Minuten, bis die Installation abgeschlossen ist.

An dieser Stelle können Sie den Status der Installation von einem anderen Terminal aus mit den folgenden Befehlen überwachen:

```
oc get clusteroperator
```

```
oc get clusterversion
```

```
watch "oc get pods \
> --all-namespaces | grep -v -E 'Running|Completed'"
```



Anmerkung

Temporäre Fehlermeldungen im Verlauf der Installation sind normal und können problemlos ignoriert werden.

Es dauert etwa 25 Minuten, bis die Installation komplett abgeschlossen ist.

8. Konfigurieren Sie den persistenten Storage für die lokale Registry mit dem NFS-Server, der auf dem Server **utility** ausgeführt wird.

Verwenden Sie den neuen Ordner **registry** im Ordner **NFS /exports** auf dem Server **utility**. Die Ordnerberechtigungen müssen 777 lauten.

- 8.1. Erstellen Sie den Ordner **/exports/registry** mit den Berechtigungen 777.

```
[lab@utility ~]$ mkdir /exports/registry  
[lab@utility ~]$ sudo chmod 777 /exports/registry/
```

8.2. Erstellen Sie die Persistent Volume-Datei (PV) `pv.yaml`.

```
[lab@utility ~]$ vi pv.yaml  
apiVersion: v1  
kind: PersistentVolume  
metadata:  
  name: registry-pv  
spec:  
  capacity:  
    storage: 5Gi  
  accessModes:  
    - ReadWriteMany  
  nfs:  
    path: /exports/registry  
    server: 192.168.50.254  
  persistentVolumeReclaimPolicy: Recycle
```



Anmerkung

Wenn noch nicht geschehen, müssen Sie die Umgebungsvariable `KUBECONFIG` für die Interaktion mit dem Cluster konfigurieren.

```
[lab@utility ~]$ export KUBECONFIG=~/ocp4upi/auth/kubeconfig
```

```
[lab@utility ~]$ oc create -f pv.yaml  
persistentvolume/registry-pv created
```

8.3. Erstellen Sie die Datei mit der Anforderung für ein persistentes Volume (PVC) `pvc.yaml`.

```
[lab@utility ~]$ vi pvc.yaml  
kind: PersistentVolumeClaim  
apiVersion: v1  
metadata:  
  name: registry-claim  
  namespace: openshift-image-registry  
spec:  
  accessModes:  
    - ReadWriteMany  
  resources:  
    requests:  
      storage: 5Gi
```

```
[lab@utility ~]$ oc create -f pvc.yaml  
persistentvolumeclaim/registry-claim created
```

- 8.4. Legen Sie für den Image-Registry-Operator den Status „Managed“ fest. Bearbeiten Sie die Konfiguration der Cluster-Image-Registry, um die PVC hinzuzufügen. Konfigurieren Sie außerdem die Image-Registry so, dass sie über zwei Pod-Replikate verfügt.

```
[lab@utility ~]$ oc edit configs.imageregistry/cluster
...output omitted...
spec:
...output omitted...
managementState: Managed
...output omitted...
proxy: {}
replicas: 2
requests:
...output omitted...
rolloutStrategy: RollingUpdate
storage:
pvc:
  claim: registry-claim
...output omitted...
config.imageregistry.operator.openshift.io/cluster edited
```

- 8.5. Überprüfen Sie die Konfiguration der Registry.

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM
		STORAGECLASS	REASON AGE		
registry-pv	5Gi	RWX	Recycle	Bound	openshift-image-registry/registry-claim
				106s	

NAMESPACE	NAME	STATUS	VOLUME	CAPACITY
ACCESS MODES	STORAGECLASS	AGE		
openshift-image-registry	registry-claim	Bound	registry-pv	5Gi
		107s		RWX

```
[lab@utility ~]$ oc get pods -n openshift-image-registry \
> -o wide | grep ^image-registry
image-registry-84dbc74f75-bd9zk           1/1     Running   0
  32s  10.128.0.34    master03
image-registry-84dbc74f75-bxdgm          1/1     Running   0
  25s  10.129.0.34    master02
```



Anmerkung

Möglicherweise sehen Sie andere Pods mit dem Status **Terminating**, während die **image-registry**-Pods konfiguriert werden.

9. Überprüfen der Integrität des Clusters.

- 9.1. Überprüfen Sie die NTP-Konfiguration.

```
[lab@utility ~]$ oc debug node/master01
Creating debug namespace/openshift-debug-node-pnwv9 ...
Starting pod/master01-debug ...
To use host binaries, run chroot /host
Pod IP: 192.168.50.10
If you don't see a command prompt, try pressing enter.
sh-4.4# chroot /host
sh-4.4# cat /etc/chrony.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
pool 2.rhel.pool.ntp.org iburst
...output omitted...

sh-4.4# sudo chronyc tracking
Reference ID      : 2D4F6F72 (chl.la)
Stratum          : 3
Ref time (UTC)   : Thu Feb 04 12:52:11 2021
System time      : 0.000170327 seconds slow of NTP time
Last offset      : -0.000190245 seconds
RMS offset       : 0.000336431 seconds
Frequency        : 16.969 ppm slow
Residual freq   : -0.005 ppm
Skew             : 0.184 ppm
Root delay       : 0.041434124 seconds
Root dispersion  : 0.002719518 seconds
Update interval  : 1031.9 seconds
Leap status      : Normal
sh-4.4# exit
exit
sh-4.4# exit
exit
```

Das System verwendet den NTP-Pool `2.rhel.pool.ntp.org`. Es ist mit dem NTP-Server `chl.la` von Stratum 3 synchron, und der Status lautet `Normal`.

Wiederholen Sie die obigen Befehle für die anderen Control Plane-Knoten, um den NTP-Status zu überprüfen.

9.2. Überprüfen Sie die Ressourcennutzungsmetriken des Clusters.

```
[lab@utility ~]$ oc adm top node
NAME      CPU(cores)    CPU%     MEMORY(bytes)   MEMORY%
master01  841m         24%      5803Mi          38%
master02  744m         21%      6282Mi          42%
master03  622m         17%      4431Mi          29%
```

9.3. Vergewissern Sie sich, dass keine Pods mit dem Status Failed angezeigt werden.

```
[lab@utility ~]$ oc get pods --all-namespaces | grep -v -E 'Running|Completed'
NAMESPACE     NAME      READY   STATUS    RESTARTS   AGE
```



Anmerkung

Es ist normal, dass Pods aus dem Namespace `openshift-apiserver` für kurze Zeit nach Abschluss der Installation zu sehen sind.

Nach ein paar Minuten wird vom vorherigen Befehl keine Ausgabe mehr gezeigt.

9.4. Überprüfen Sie die Integrität des etcd-Clusters.

```
[lab@utility ~]$ oc rsh -n openshift-etcd etcd-master01
Defaulting container name to etcdctl.
Use 'oc describe pod/etcd-master01 -n openshift-etcd' to see all of the containers
in this pod.
sh-4.4# etcdctl endpoint health --cluster
https://192.168.50.10:2379 is healthy: successfully committed proposal: took =
8.470285ms
https://192.168.50.12:2379 is healthy: successfully committed proposal: took =
9.379322ms
https://192.168.50.11:2379 is healthy: successfully committed proposal: took =
9.99598ms
sh-4.4# exit
exit
```

9.5. Rufen Sie die API-Version ab, um den Status der Cluster-API zu überprüfen.

```
[lab@utility ~]$ curl -k https://api.ocp4.example.com:6443/version
{
  "major": "1",
  "minor": "19",
  "gitVersion": "v1.19.0+9f84db3",
  "gitCommit": "9f84db336d1a77cba52684ecb51bfb197e9b4533",
  "gitTreeState": "clean",
  "buildDate": "2020-10-30T09:33:51Z",
  "goVersion": "go1.15.2",
  "compiler": "gc",
  "platform": "linux/amd64"
```

9.6. Überprüfen Sie die Verfügbarkeit der OpenShift-Konsole.

```
[lab@utility ~]$ curl -kIs \
> https://console-openshift-console.apps.ocp4.example.com
HTTP/1.1 200 OK
...output omitted...
```

9.7. Verifizieren Sie die Verfügbarkeit der Image-Registry auf den Cluster-Knoten.

```
[lab@utility ~]$ oc debug node/master01
Creating debug namespace/openshift-debug-node-zjb2h ...
Starting pod/master01-debug ...
To use host binaries, run chroot /host
Pod IP: 192.168.50.10
If you don't see a command prompt, try pressing enter.
```

```
sh-4.4# chroot /host
sh-4.4# curl -kIs https://image-registry.openshift-image-registry.svc:5000/healthz
HTTP/2 200
cache-control: no-cache
date: Tue, 02 Feb 2021 12:01:49 GMT

sh-4.4# exit
exit
sh-4.4# exit
exit

Removing debug pod ...
Removing debug namespace/openshift-debug-node-zjb2h ...
```

Wiederholen Sie diesen Schritt für die Control Plane-Knoten `master02` und `master03`.

10. Stellen Sie eine `rails-postgresql-example`-Anwendung im `test`-Projekt zum Durchführen eines Funktionstests des Clusters bereit.

- 10.1. Stellen Sie eine Testanwendung zur Überprüfung der ordnungsgemäßen Funktion des Build-Prozesses und des Registry-Storage bereit.

```
[lab@utility ~]$ oc new-project test
Now using project "test" on server "https://api.ocp4.example.com:6443".

You can add applications to this project with the 'new-app' command. For example,
use the following command to build a new example application in Ruby.

  oc new-app rails-postgresql-example

Or, you can use kubectl to deploy a simple Kubernetes application.

  kubectl create deployment hello-node --image=k8s.gcr.io/serve_hostname
```

```
[lab@utility ~]$ oc new-app rails-postgresql-example
--> Deploying template "openshift/rails-postgresql-example" to project test

  Rails + PostgreSQL (Ephemeral)
  -----
  An example Rails application with a PostgreSQL database. For more information
  about using this template, including OpenShift considerations, see https://
  github.com/sclorg/rails-ex/blob/master/README.md.

  WARNING: Any data stored will be lost upon pod destruction. Only use this
  template for testing.

  The following service(s) have been created in your project: rails-postgresql-
example, postgresql.

  For more information about using this template, including OpenShift
  considerations, see https://github.com/sclorg/rails-ex/blob/master/README.md.

  * With parameters:
```

```
...output omitted...

--> Creating resources ...
secret "rails-postgresql-example" created
service "rails-postgresql-example" created
route.route.openshift.io "rails-postgresql-example" created
imagestream.image.openshift.io "rails-postgresql-example" created
buildconfig.build.openshift.io "rails-postgresql-example" created
deploymentconfig.apps.openshift.io "rails-postgresql-example" created
service "postgresql" created
deploymentconfig.apps.openshift.io "postgresql" created
--> Success
Access your application via route 'rails-postgresql-example-test.apps.ocp4.example.com'
Build scheduled, use 'oc logs -f buildconfig/rails-postgresql-example' to track its progress.
Run 'oc status' to view your app.
```

10.2. Sehen Sie sich die Build-Prozessprotokolle an.

```
[lab@utility ~]$ oc logs -f buildconfig/rails-postgresql-example
Cloning "https://github.com/sclorg/rails-ex.git" ...

...output omitted...

Successfully pushed image-registry.openshift-image-registry.svc:5000/test/rails-
postgresql-example@sha256:b97b...ff82
Push successful
```

Das Image wird mit dem persistenten Storage erstellt und erfolgreich an die interne Registry übertragen.

10.3. Überprüfen Sie, ob alle Pods im Namespace **test** den Status **Running** oder **Completed** aufweisen.

```
[lab@utility ~]$ oc get pods -n test
NAME                               READY   STATUS    RESTARTS   AGE
postgresql-1-deploy                0/1     Completed  0          2m50s
postgresql-1-pgx5x                 1/1     Running   0          2m47s
rails-postgresql-example-1-build   0/1     Completed  0          2m50s
rails-postgresql-example-1-c4ljh   1/1     Running   0          31s
rails-postgresql-example-1-deploy  0/1     Completed  0          52s
rails-postgresql-example-1-hook-pre 0/1     Completed  0          47s
```

10.4. Rufen Sie den Pfad für die Testanwendung **rails-postgres-example** ab.

```
[lab@utility ~]$ oc get routes -n test
NAME           HOST/PORT
PATH  SERVICES      PORT  TERMINATION  WILDCARD
rails-postgresql-example  rails-postgresql-example-test.apps.ocp4.example.com
                           rails-postgresql-example  <all>           None
```

Kapitel 7 | Red Hat OpenShift Installation Lab - Ausführliche Wiederholung

10.5. Öffnen Sie auf dem Rechner „workstation“ den Chromium-Webbrowser, und navigieren Sie zu `rails-postgresql-example-test.apps.ocp4.example.com`.

Bewertung

Verwenden Sie als Benutzer `student` auf dem Rechner `workstation` den Befehl `lab`, um Ihre Arbeit zu bewerten. Beheben Sie sämtliche gemeldeten Fehler, und führen Sie den Befehl so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[lab@utility ~]$ exit
...output omitted...
[student@workstation ~]$ lab comprehensive-review grade
```

Beenden

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab comprehensive-review finish
```



Anmerkung

Mit dem Befehl `lab comprehensive-review finish` wird der Cluster nicht entfernt. Das Ausführen des Skripts `lab comprehensive-review grade` nach dem Befehl `lab comprehensive-review finish` zeigt an, dass der Cluster mit drei Knoten vorhanden ist.

Wenn Sie vorne beginnen möchten, setzen Sie die Rechner `bootstrap`, `master01`, `master02` und `master03` zurück, und starten Sie dann erneut mit dem Schritt, der Sie anweist, RHCOS auf den Rechnern „bootstrap“, „master01“, „master02“ und „master03“ über das PXE-Menü zu installieren.

Hiermit ist die praktische Übung beendet.