



Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster



OCP 4.6 DO280

Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster

Ausgabe 120210723

Veröffentlicht 20210723

Autoren: Zach Guterman, Dan Kolepp, Eduardo Ramirez Ronco, Jordi Sola Alaball, Richard Allred, Michael Jarrett, Harpal Singh, Federico Fapitalle, Maria Fernanda Ordóñez Casado
Editor: Seth Kenlon, Dave Sacco, Connie Petlitzer, Nicole Muller, Sam Ffrench

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: Forrest Taylor, Manuel Aude Morales, James Mighion, Michael Phillips und Fiona Allen

Dokumentkonventionen	vii
	vii
Einführung	ix
DO280 Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster	ix
Informationen zur Kursumgebung	x
Durchführen von praktischen Übungen	xix
1. Beschreiben von Red Hat OpenShift Container Platform	1
Beschreiben von OpenShift Container Platform-Funktionen	2
Quiz: Beschreiben von OpenShift Container Platform-Funktionen	7
Beschreiben der OpenShift-Architektur	11
Quiz: Beschreiben der OpenShift-Architektur	14
Beschreiben von Cluster-Operatoren	18
Quiz: Beschreiben von Cluster-Operatoren	21
Zusammenfassung	23
2. Überprüfen der Integrität eines Clusters	25
Beschreiben der Installationsmethoden	26
Quiz: Beschreiben der Installationsmethoden	29
Fehlerbehebung in OpenShift-Clustern und -Anwendungen	31
Angeleitete Übung: Fehlerbehebung in OpenShift-Clustern und -Anwendungen	40
Einführung in OpenShift Dynamic Storage	47
Angeleitete Übung: Einführung in OpenShift Dynamic Storage	51
Zusammenfassung	56
3. Konfigurieren von Autorisierung und Authentifizierung	57
Konfigurieren der Identitätsanbieter	58
Angeleitete Übung: Konfigurieren der Identitätsanbieter	66
Definieren und Anwenden von Berechtigungen mit RBAC	76
Angeleitete Übung: Definieren und Anwenden von Berechtigungen mit RBAC	80
Praktische Übung: Überprüfen der Integrität eines Clusters	86
Zusammenfassung	95
4. Konfigurieren der Anwendungssicherheit	97
Verwalten von vertraulichen Informationen mit Secrets	98
Angeleitete Übung: Verwalten von vertraulichen Informationen mit Secrets	103
Kontrollieren von Anwendungsberechtigungen mit Sicherheitskontextbeschränkungen	109
Angeleitete Übung: Kontrollieren von Anwendungsberechtigungen mit Sicherheitskontextbeschränkungen	112
Praktische Übung: Konfigurieren der Anwendungssicherheit	116
Zusammenfassung	124
5. Konfigurieren des OpenShift-Netzwerks für Anwendungen	125
Beheben von Software-Defined Networking-Fehlern in OpenShift	126
Angeleitete Übung: Beheben von Software-Defined Networking-Fehlern in OpenShift	133
Anwendungen für den externen Zugriff bereitstellen	142
Angeleitete Übung: Anwendungen für den externen Zugriff bereitstellen	148
Konfigurieren von Netzwerk-Richtlinien	159
Angeleitete Übung: Konfigurieren von Netzwerk-Richtlinien	163
Praktische Übung: Konfigurieren des OpenShift-Netzwerks für Anwendungen	172
Zusammenfassung	185
6. Steuern der Pod-Zuordnung (Scheduling)	187
Steuern des Pod-Zuordnungsverhaltens	188
Angeleitete Übung: Steuern des Pod-Zuordnungsverhaltens	195
Beschränken der Ressourcennutzung einer Anwendung	201

Angeleitete Übung: Beschränken der Ressourcennutzung einer Anwendung	213
Skalieren einer Anwendung	223
Angeleitete Übung: Skalieren einer Anwendung	227
Praktische Übung: Steuern der Pod-Zuordnung (Scheduling)	233
Zusammenfassung	241
7. Beschreiben von Cluster-Updates	243
Beschreiben des Prozesses für Cluster-Updates	244
Quiz: Beschreiben des Prozesses für Cluster-Updates	255
Zusammenfassung	259
8. Verwalten eines Clusters mit der Web Console	261
Durchführen der Cluster-Verwaltung	262
Angeleitete Übung: Durchführen der Cluster-Verwaltung	266
Verwalten von Workloads und Operatoren	273
Angeleitete Übung: Verwalten von Workloads und Operatoren	278
Untersuchen von Cluster-Metriken	287
Angeleitete Übung: Untersuchen von Cluster-Metriken	291
Praktische Übung: Verwalten eines Clusters mit der Web Console	296
Zusammenfassung	308
9. Ausführliche Wiederholung	309
Ausführliche Wiederholung	310
Praktische Übung: Fehlerbehebung in OpenShift-Clustern und -Anwendungen	312
Praktische Übung: Konfigurieren einer Projektvorlage mit Ressourcen- und Netzwerkbeschränkungen	326

Dokumentkonventionen

In diesem Abschnitt werden verschiedene Konventionen und Praktiken beschrieben, die in allen Red Hat Training-Kursen verwendet werden.

Verweise

Die Red Hat Training-Kurse verwenden folgende Verweisarten:



Literaturhinweise

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



Anmerkung

Diese 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 diesen Feldern werden Details hervorgehoben, die andernfalls leicht übersehen werden könnten: Konfigurationsänderungen, die nur die aktuelle Sitzung betreffen, oder Services, die neu gestartet werden müssen, bevor ein Update angewendet werden kann. Wenn Sie diese ignorieren, führt dies nicht zu Datenverlust, kann jedoch Irritationen und Frustration hervorrufen.



Warnung

Diese sollten nicht ignoriert werden. Wenn Sie diese ignorieren, führt dies mit großer Wahrscheinlichkeit zu einem Datenverlust.

Inklusive Sprache

Red Hat Training überprüft derzeit die Verwendung der Sprache in verschiedenen Bereichen, um potenziell anstößige Begriffe zu entfernen. Dies ist ein fortlaufender Prozess und erfordert die Anpassung an die Produkte und Services, die in Red Hat Training-Kursen behandelt werden. Red Hat schätzt Ihre Geduld während dieses Prozesses.

Einführung

DO280 Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster

Red Hat® OpenShift® Container Platform ist eine Container-Anwendungsplattform, auf der Unternehmen mithilfe von Container-Bereitstellungen Anwendungen verwalten und skalieren können. OpenShift bietet vordefinierte, auf Kubernetes basierende Anwendungsumgebungen zur Unterstützung von DevOps-Prinzipien wie schnelle Markteinführung, Continuous Integration (CI) und Continuous Delivery (CD).

Im Kurs „Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster“ (DO280) lernen die Kursteilnehmer, wie die Red Hat® OpenShift® Container Platform konfiguriert und verwaltet wird und wie auftretende Fehler behoben werden. In diesem praxisorientierten Kurs erfahren die Teilnehmer, wie sie die Installation eines Clusters überprüfen, ihn konfigurieren und im Alltag verwalten können.

Lerninhalte

Installieren, Konfigurieren und Verwalten sowie Fehlerbehebung von OpenShift-Clustern. Dieser Kurs dient zusammen mit dem Kurs „Red Hat OpenShift I: Containers & Kubernetes (DO180)“ als Vorbereitung des Kursteilnehmers auf die Prüfung „Red Hat Certified Specialist in OpenShift Administration (EX280)“.

Zielgruppe

System- und Softwarearchitekten, Systemadministratoren, Cluster Operators und Site Reliability Engineers

Voraussetzungen

Abschluss des Kurses „Red Hat OpenShift I: Containers & Kubernetes (DO180)“ oder gleichwertige Kenntnisse
Zertifizierung als Red Hat Certified System Administrator (RHCSA) oder gleichwertige Kenntnisse

Informationen zur Kursumgebung

Der Workstation-Rechner

In diesem Kurs wird `workstation` als primäres Computersystem für praktische Übungen verwendet.

Der Rechner `workstation` verfügt über das standardmäßige Benutzerkonto `student` mit dem Passwort `student`. Sie müssen sich bei keiner Übung in diesem Kurs als `root` anmelden. Wenn dies trotzdem erforderlich ist, lautet das `root`-Passwort auf dem `workstation`-Rechner `redhat`.

Sie geben auf dem `workstation`-Rechner `oc`-Befehle ein, um den OpenShift-Cluster zu verwalten. Dieser ist in der Kursumgebung bereits vorinstalliert.

Außerdem führen Sie auf dem `workstation`-Rechner Shell-Skripts und Ansible-Playbooks aus, die zur Bearbeitung 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.



Anmerkung

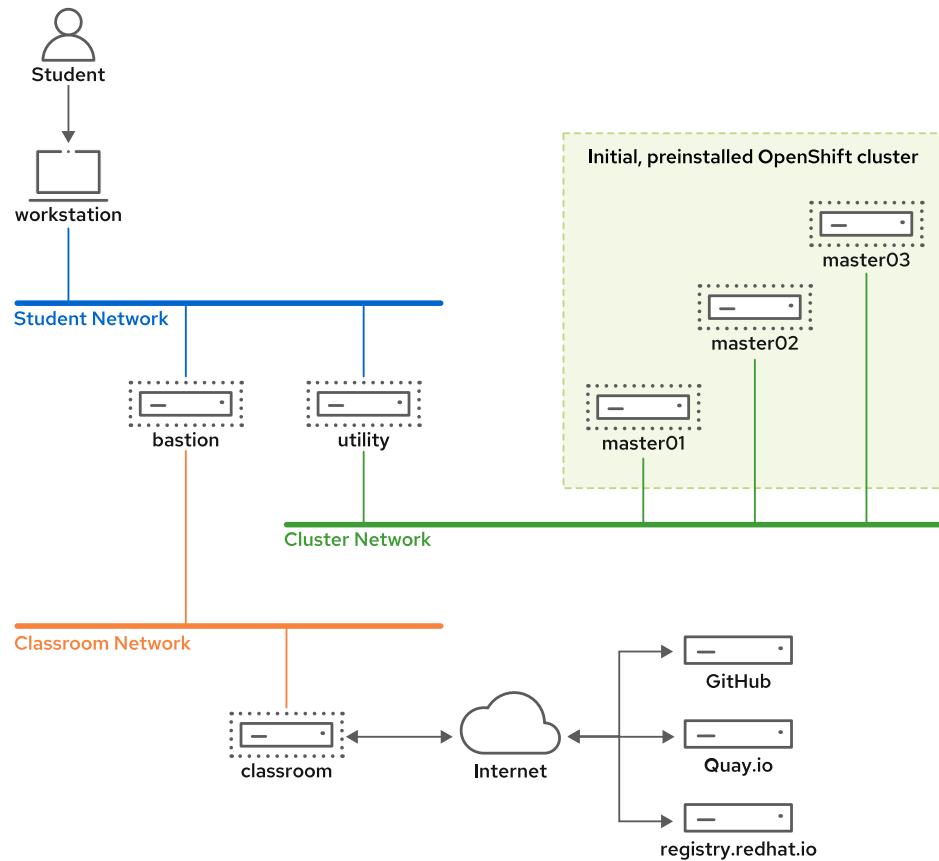
Wenn Sie Ihre Kursumgebung zum ersten Mal starten, dauert es einige Zeit, bis die OpenShift-Cluster verfügbar sind. Der `lab`-Befehl zu Beginn jeder Übung prüft und wartet nach Bedarf. Wenn Sie versuchen, mit dem Befehl `oc` oder der Web Console auf Ihren Cluster zuzugreifen, ohne zunächst einen `lab`-Befehl auszuführen, stellen Sie möglicherweise fest, dass der Cluster noch nicht verfügbar ist. Wenn dies der Fall ist, warten Sie ein paar Minuten, und versuchen Sie es erneut.

Die Kursumgebung

Jeder Teilnehmer erhält eine komplette Remote-Kursumgebung. Als Teil dieser Umgebung erhält jeder Teilnehmer einen dedizierten OpenShift-Cluster für die Durchführung von Administrationsaufgaben.

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.



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. Es wird nicht erwartet, dass Sie direkt mit diesen Systemen interagieren.

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 mit dem Namen **master_XX_** 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

Rechnername	IP-Adressen	Rolle
classroom.example.com	172.25.254.254	Router, der das Classroom Network mit dem Internet verbindet
bastion.lab.example.com	172.25.250.254	Router, der das Student Network mit dem Classroom Network verbindet
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

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 beim Starten der Kursumgebung nicht verfügbar ist, wird der OpenShift-Cluster möglicherweise nicht gestartet oder kann in einen Degraded-Status wechseln. 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 wird mit der vorhandenen Infrastruktur-Installationsmethode vorinstalliert. Alle Knoten werden als Bare-Metal-Server behandelt, obwohl es sich eigentlich um virtuelle Rechner in einem OpenStack-Cluster handelt.

Die Funktionen zur Integration von OpenShift-Cloud-Providern sind nicht aktiviert, und einige Funktionen, die von dieser Integration abhängen, wie z. B. Rechnersätze 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 als `admin`-Benutzer beim OpenShift-Cluster anmelden können, weil Sie die Cluster-Authentifizierungseinstellungen falsch geändert haben, führen Sie den Befehl `lab finish` in der aktuellen Übung aus. Starten Sie die Übung neu, indem Sie den Befehl `lab start` ausführen.

Für Labs, die die Benutzer `admin` und `developer` erwarten, setzt der `Lab`-Befehl die Cluster-Authentifizierungseinstellungen zurück und stellt Passwörter wieder her, sodass der Benutzer `admin` das Passwort `redhat` hat und der Benutzer `developer` das Passwort `developer` besitzt.

Einführung

Wenn die Ausführung eines `lab`-Befehls nicht ausreicht, können Sie den Anweisungen im nächsten Abschnitt folgen, um den `utility`-Rechner für den Zugriff auf Ihren OpenShift-Cluster zu verwenden.

Beheben von Fehlern beim Zugriff auf den OpenShift-Cluster

Der `utility`-Rechner wurde verwendet, um das OpenShift-Installationsprogramm in der Kursumgebung auszuführen. Er ist eine nützliche Ressource zur Behebung von Cluster-Problemen. Sie können die Installationsmanifeste im Ordner `/home/lab/ocp4` des `utility`-Rechners anzeigen.

Zum Ausführen von Übungen ist die Anmeldung beim `utility`-Server nicht erforderlich. Wenn es anscheinend zu lange dauert, bis der OpenShift-Cluster gestartet wird, oder er sich in einem Degraded-Status befindet, können Sie sich auf dem `utility`-Rechner als `lab`-Benutzer anmelden, um Fehler in der Kursumgebung zu beheben.

Der `student`-Benutzer auf dem `workstation`-Rechner ist bereits mit SSH-Schlüsseln konfiguriert. Diese ermöglichen die Anmeldung beim `utility`-Rechner ohne Passwort.

```
[student@workstation ~]$`ssh lab@utility`
```

Auf dem `utility`-Rechner ist der `lab`-Benutzer mit einer `.kube/config`-Datei vorkonfiguriert, die den Zugriff als `system:admin` gewährt, ohne dass zuvor `oc login` erforderlich ist.

Auf diese Weise können Sie Befehle zur Fehlerbehebung ausführen, wie z. B. `oc get node`, wenn diese auf dem `workstation`-Rechner nicht funktionieren.

Es sollte kein SSH-Zugriff auf die OpenShift-Cluster-Knoten für reguläre Administrationsaufgaben notwendig sein, da OpenShift 4 den Befehl `oc debug`-Befehl bereitstellt. Falls erforderlich, ist der `lab`-Benutzer auf dem `utility`-Server mit SSH-Schlüsseln vorkonfiguriert, damit er auf alle Cluster-Knoten zuzugreifen kann. Beispiel:

```
[lab@utility ~]$`ssh -i ~/.ssh/lab_rsa core@master01.ocp4.example.com`
```

Ersetzen Sie im vorherigen Beispiel `master01` durch den Namen des gewünschten Cluster-Knotens.

Genehmigen von Knotenzertifikaten für den OpenShift-Cluster

Red Hat OpenShift Container Platform-Cluster sind für die dauerhafte Ausführung rund um die Uhr konzipiert, bis sie außer Betrieb genommen werden. Im Gegensatz zu einem Produktions-Cluster enthält die Kursumgebung einen Cluster, der nach der Installation angehalten wurde. Außerdem wird er im Laufe des Kurses mehrmals angehalten und neu gestartet. Daher ist dies ein Szenario, das eine spezielle Handhabung erfordert, die bei einem Produktions-Cluster nicht notwendig wäre.

Die Control Plane und die Server-Knoten in einem OpenShift-Cluster kommunizieren häufig miteinander. Die gesamte Kommunikation zwischen Cluster-Knoten wird durch die gegenseitige Authentifizierung auf Basis von TLS-Zertifikaten pro Knoten geschützt.

Das OpenShift-Installationsprogramm erstellt und genehmigt TLS Certificate Signing Requests (CSRs) für die Full-Stack-Automation-Installationsmethode. Es wird erwartet, dass der Systemadministrator die CSRs für die bereits vorhandene Infrastrukturinstallationsmethode manuell genehmigt.

Einführung

Alle TLS-Zertifikate pro Knoten haben eine kurze Gültigkeitsdauer von 24 Stunden (erstmalig) und 30 Tagen (nach Verlängerung). Wenn sie bald ablaufen, erstellen die betroffenen Cluster-Knoten neue CSRs, und die Control Plane genehmigt sie automatisch. Falls die Control Plane offline ist, wenn das TLS-Zertifikat eines Knotens abläuft, muss ein Cluster-Administrator die ausstehende CSR genehmigen.

Der **utility**-Rechner enthält einen Systemservice, der CSRs vom Cluster genehmigt, wenn Sie den Kurs starten. So wird sichergestellt, dass der Cluster bereit ist, wenn Sie mit den Übungen beginnen. Wenn Sie die Kursumgebung erstellen oder starten und zu schnell mit einer Übung beginnen, stellen Sie möglicherweise fest, dass der Cluster noch nicht bereit ist. Wenn dies der Fall ist, warten Sie ein paar Minuten, während der **utility**-Rechner die CSRs verarbeitet, und versuchen Sie es dann erneut.

Manchmal kann der **utility**-Rechner nicht alle erforderlichen CSRs genehmigen. Das ist beispielsweise der Fall, wenn der Cluster zu lange gebraucht hat, um alle erforderlichen CSR-Anforderungen zu generieren, und der Systemservice nicht lange genug gewartet hat. Es ist auch möglich, dass einige OpenShift-Cluster-Knoten nicht lange genug gewartet haben, bis die CSRs genehmigt wurden, und neue CSRs ausgegeben haben, die die vorherigen ersetzen.

Wenn diese Probleme auftreten, werden Sie feststellen, dass der Cluster zu lange braucht, bis er verfügbar ist, und die Befehle `oc login` oder `lab` fehlschlagen. Um das Problem zu beheben, können Sie sich wie bereits oben erläutert auf dem **utility**-Rechner anmelden und das Skript `sign.sh` ausführen, um zusätzliche und ausstehende CSRs zu genehmigen.

```
[lab@utility ~]$ ./sign.sh`
```

Das Skript `sign.sh` wird einige Male ausgeführt, für den Fall, dass die Cluster-Knoten neue CSRs ausgeben, die die genehmigten ersetzen.

Nachdem entweder Sie oder der Systemservice auf dem **utility**-Rechner alle CSRs genehmigt haben, muss OpenShift einige Cluster-Operatoren neu starten. Es dauert einen Augenblick, bis der OpenShift-Cluster bereit ist, Anfragen von Clients zu beantworten. Zur Unterstützung dieses Szenarios stellt der **utility**-Rechner das `wait.sh`-Skript bereit. Dieses wartet, bis der OpenShift-Cluster Authentifizierungs- und API-Anforderungen von Remote-Clients akzeptiert.

```
[lab@utility ~]$ ./wait.sh`
```

Wenn weder der Service auf dem **utility**-Rechner noch die Skripte `sign.sh` und `wait.sh` den OpenShift-Cluster bereitstellen, damit Sie die Übungen starten können, öffnen Sie ein Kundensupport-Ticket.



Anmerkung

Sie können Befehle zur Fehlerbehebung jederzeit über den **utility**-Rechner ausführen, selbst wenn Sie über Control Plane-Knoten verfügen, die nicht bereit sind. Einige nützliche Befehle sind: * `oc get node`, um zu überprüfen, ob alle Ihre Cluster-Knoten bereit sind. * `oc get csr`, um zu überprüfen, ob Ihr Cluster noch ausstehende, nicht genehmigte CSRs aufweist. * `oc get co`, um zu überprüfen, ob einer Ihrer Cluster-Operatoren nicht verfügbar ist, den Degraded-Status aufweist oder die Konfiguration durchläuft und Pods implementiert.

Wenn diese Befehle fehlschlagen, können Sie versuchen, die Kursumgebung als letzten Ausweg zu löschen und neu zu erstellen, bevor Sie ein Kundensupport-Ticket erstellen.

Steuerung Ihrer Systeme

Kursteilnehmern werden Remote-Computer in einem Red Hat Online Learning-Kursraum zugewiesen. Der Zugriff darauf erfolgt über eine Webanwendung, die unter <http://rol.redhat.com/> gehostet wird. Kursteilnehmer sollten sich mithilfe ihrer Anmeldedaten für das Red Hat Customer Portal auf dieser Website anmelden.

Steuern der virtuellen Rechner

Die virtuellen Rechner in Ihrer Kursumgebung werden über eine Webseite gesteuert. Der Status jedes virtuellen Rechners in der Kursumgebung wird auf der unter der Registerkarte Lab Environment befindlichen Seite angezeigt.

Rechnerstatus

VM-Status	Beschreibung
active	Der virtuelle Rechner wird ausgeführt und ist verfügbar (oder wird es bald sein, falls er noch bootet).
stopped	Der virtuelle Rechner ist vollständig heruntergefahren.
building	Der Ersterstellung des virtuellen Rechners wird durchgeführt.

Abhängig vom Status eines Rechners steht eine Auswahl der folgenden Aktionen zur Verfügung.

Aktionen für Kursumgebung/Rechner

Schaltfläche oder Aktion	Beschreibung
CREATE	Erstellt die ROL-Kursumgebung. Hiermit werden sämtliche für die Kursumgebung erforderlichen virtuellen Rechner erstellt und gestartet. Dies dauert ggf. mehrere Minuten.
DELETE	Entfernen Sie den ROL-Kursraum. Hiermit werden alle virtuellen Rechner im Kursraum entfernt. Achtung: Sämtliche auf den Disks gespeicherte Arbeit geht verloren.
START	Startet alle virtuellen Rechner in der Kursumgebung.
STOP	Hält alle virtuellen Rechner in der Kursumgebung an.

Schaltfläche oder Aktion	Beschreibung
OPEN CONSOLE	Öffnet eine neue Registerkarte im Browser und stellt eine Verbindung zwischen Konsole und virtuellem Rechner her. Kursteilnehmer können sich direkt beim virtuellen Rechner anmelden und Befehle ausführen. In den meisten Fällen sollten sich die Kursteilnehmer beim virtuellen Rechner workstation anmelden und ssh verwenden, um mit anderen virtuellen Rechnern eine Verbindung herzustellen.
ACTIONStart	Startet den virtuellen Rechner (d. h. schaltet ihn ein).
ACTIONShutdown	Fährt den virtuellen Rechner ordnungsgemäß herunter, damit die Disk-Inhalte nicht verloren gehen.
ACTIONPower Off	Erzwingt ein Herunterfahren des virtuellen Rechners und behält die Inhalte seiner Disk bei. Dies entspricht der Stromabschaltung bei einem physischen Rechner.
ACTIONReset	Erzwingen Sie das Herunterfahren des virtuellen Rechners, und setzen Sie die Disk in den Ursprungszustand zurück. Achtung: Sämtliche auf der Disk gespeicherte Arbeit geht verloren.

Klicken Sie zu Beginn einer Übung, sofern Sie angewiesen wurden, einen einzelnen Knoten eines virtuellen Rechners zurückzusetzen, nur für den bestimmten virtuellen Rechner auf **ACTION → Reset**.

Klicken Sie zu Beginn einer Übung, sofern Sie angewiesen wurden, alle virtuellen Rechner zurückzusetzen, auf **ACTION → Reset**.

Wenn Sie die Kursumgebung auf ihren ursprünglichen Zustand beim Start des Kurses zurücksetzen möchten, können Sie auf **DELETE** klicken, um die gesamte Kursumgebung zu entfernen.

Nach dem Löschen des Labs können Sie auf **CREATE** klicken, um einen neuen Satz von Kursumgebungssystemen bereitzustellen.



Warnung

Der Vorgang **DELETE** kann nicht rückgängig gemacht werden. Die von Ihnen bis zu diesem Zeitpunkt in der Kursumgebung vorgenommene Arbeit geht verloren.

Der Autostop-Timer

Die Registrierung bei Red Hat Online Learning ermöglicht Kursteilnehmern eine bestimmte Menge Zeit am Computer. Für den sparsamen Umgang mit der vorgegebenen Computerzeit verfügt der ROL-Kursraum über einen verknüpften Zählvorgang, der die Kursumgebung herunterfährt, wenn der Timer abgelaufen ist.

Einführung

Um den Timer anzupassen, klicken Sie auf +. Damit fügen Sie eine Stunde zum Timer hinzu. Beachten Sie, dass die maximale Zeit zwölf Stunden beträgt.

Steuerung Ihrer Systeme

Ihnen werden Remote-Computer in einem Red Hat Online Learning-Kursraum zugewiesen. Der Zugriff darauf erfolgt über eine unter rol.redhat.com [<http://rol.redhat.com>] gehostete Webanwendung. Sie sollten sich mithilfe Ihrer Anmelddaten für das Red Hat Customer Portal auf dieser Website anmelden.

Steuern der virtuellen Rechner

Die virtuellen Rechner in Ihrer Kursumgebung werden über eine Webseite gesteuert. Der Status jedes virtuellen Rechners im Kursraum wird auf der unter der Registerkarte **Online Lab** befindlichen Seite angezeigt.

Rechnerstatus

VM-Status	Beschreibung
STARTING	Der virtuelle Rechner wird hochgefahren.
STARTED	Der virtuelle Rechner wird ausgeführt und ist verfügbar (oder, falls noch hochgefahren wird, wird es bald sein.)
STOPPING	Der virtuelle Rechner wird heruntergefahren.
STOPPED	Der virtuelle Rechner ist vollständig heruntergefahren. Beim Starten fährt der virtuelle Rechner in denselben Status hoch, in dem er sich vor dem Herunterfahren befand. (Die Disk wurde nicht gelöscht.)
PUBLISHING	Der virtuelle Rechner wird anfänglich erstellt.
WAITING_TO_START	Der virtuelle Rechner wartet auf den Start anderer virtueller Rechner.

In Abhängigkeit des Status eines Rechners steht eine Auswahl der folgenden Aktionen zur Verfügung.

Aktionen für Kursumgebung/Rechner

Schaltfläche oder Aktion	Beschreibung
PROVISION LAB	Erstellen Sie den ROL-Kursraum. Hiermit werden sämtliche für die Kursumgebung erforderlichen virtuellen Rechner erstellt und gestartet. Dies dauert ggf. mehrere Minuten.
DELETE LAB	Entfernen Sie den ROL-Kursraum. Hiermit werden alle virtuellen Rechner im Kursraum entfernt. Achtung: Alle auf den Disks gespeicherten Arbeiten gehen verloren.
START LAB	Starten Sie alle virtuellen Rechner im Kursraum.
SHUTDOWN LAB	Halten Sie alle virtuellen Rechner im Kursraum an.

Schaltfläche oder Aktion	Beschreibung
OPEN CONSOLE	Öffnen Sie eine neue Registerkarte im Browser, und stellen Sie eine Verbindung zwischen Konsole und virtuellem Rechner her. Sie können sich direkt beim virtuellen Rechner anmelden und Befehle ausführen. In den meisten Fällen sollten Sie sich beim virtuellen Rechner workstation anmelden und ssh verwenden, um mit anderen virtuellen Rechnern eine Verbindung herzustellen.
ACTION → Start	Startet den virtuellen Rechner (d. h. schaltet ihn ein).
ACTION → Shutdown	Fährt den virtuellen Rechner ordnungsgemäß herunter, damit die Disk-Inhalte nicht verloren gehen.
ACTION → Power Off	Erzwingt ein Herunterfahren des virtuellen Rechners und behält die Inhalte seiner Disk bei. Dies entspricht der Stromabschaltung bei einem physischen Rechner.
ACTION → Reset	Erzwingt das Herunterfahren des virtuellen Rechners und setzt die Disk in den Ursprungszustand zurück. Achtung: Sämtliche auf der Disk gespeicherte Arbeit geht verloren.

Klicken Sie zu Beginn einer Übung, sofern Sie angewiesen wurden, einen einzelnen Knoten eines virtuellen Rechners zurückzusetzen, nur für den bestimmten virtuellen Rechner auf ACTION → Reset.

Klicken Sie zu Beginn einer Übung, sofern Sie angewiesen wurden, alle virtuellen Rechner zurückzusetzen, auf ACTION → Reset.

Wenn Sie die Kursumgebung auf ihren ursprünglichen Zustand beim Start des Kurses zurücksetzen möchten, können Sie auf **DELETE LAB** klicken, um die gesamte Kursumgebung zu entfernen. Nach dem Löschen des Labs können Sie auf **PROVISION LAB** klicken, um einen neuen Satz von Kurssystemen bereitzustellen.



Warnung

Der Vorgang **DELETE LAB** kann nicht rückgängig gemacht werden. Die von Ihnen bis zu diesem Zeitpunkt in der Kursumgebung vorgenommene Arbeit geht verloren.

Der Autostop-Timer

Mit der Registrierung bei Red Hat Online Learning erhalten Sie eine bestimmte Menge Zeit am Computer. Für den sparsamen Umgang mit der vorgegebenen Computerzeit verfügt der ROL-Kursraum über einen verknüpften Zählvorgang, der die Kursumgebung herunterfährt, wenn der Timer abgelaufen ist.

Klicken Sie zum Anpassen des Timers auf **MODIFY**, damit das Dialogfeld **New Autostop Time** angezeigt wird. Legen Sie die Anzahl der Stunden fest, bis der Kursraum automatisch angehalten wird. Beachten Sie, dass die maximale Zeit zehn Stunden beträgt. Klicken Sie auf **ADJUST TIME**, um diese Änderung auf die Timer-Einstellungen anzuwenden.

Durchführen von praktischen Übungen

Durchführen von praktischen Übungen

Führen Sie den Befehl `lab` auf dem Rechner `workstation` aus, um Ihre Umgebung vor jeder praktischen Übung vorzubereiten und nach einer Übung wieder zu bereinigen. Jede praktische Übung hat einen eindeutigen Namen innerhalb eines Kurses. Der Übung ist `lab-` als Dateiname in `/usr/local/lib` vorangestellt. Beispielsweise hat die Übung `instances-cli` den Dateinamen `/usr/local/lib/lab-instances-cli`. Um die verfügbaren Übungen aufzulisten, verwenden Sie die Tab-Vervollständigung im Befehl `lab`. Beachten Sie, dass sich das Wort „Tab“ im folgenden Befehl auf das Drücken der Tabulatortaste auf Ihrer Tastatur bezieht:

```
[student@workstation ~]$ `lab Tab Tab`  
administer-users  deploy-overcloud-lab  prep-deploy-ips      stacks-autoscale  
analyze-metrics   instances-cli        prep-deploy-router  stacks-deploy  
assign-roles       manage-interfaces public-instance-deploy verify-overcloud
```

Es gibt zwei Übungstypen. Der erste Typ, die angeleitete Übung, ist eine praktische Übung zum jeweiligen Kursabschnitt. Wenn ein Abschnitt mit einem Test endet, bedeutet dies normalerweise, dass das Thema keine praktische Übung enthält. Der zweite Typ, eine Übung am Ende eines Kapitels, ist eine bewertbare Übung, mit der Sie Ihre Kenntnisse überprüfen können. Wenn ein Kurs eine ausführliche Wiederholung umfasst, sind die Wiederholungsübungen als bewertbare Übungen strukturiert. Die Syntax für das Ausführen eines Übungsskripts lautet:

```
[student@workstation ~]$ `lab _exercise action_`
```

Bei `action` handelt es sich entweder um `start`, `grade` oder `finish`. Für alle Übungen ist `start` und `finish` möglich. `grade` kann nur für die Übungen am Ende eines Kapitels und für die ausführlichen Wiederholungsübungen verwendet werden. Bei älteren Kursen werden möglicherweise weiterhin `setup` und `cleanup` anstelle der aktuellen Aktionen `start` und `finish` verwendet.

start

Früher `setup`. Die Startlogik des Skripts überprüft die Ressourcen, die zum Starten der Übung erforderlich sind. Dazu gehört die Konfiguration von Einstellungen, das Erstellen von Ressourcen, die Überprüfung der benötigten Services und die Überprüfung der erforderlichen Ergebnisse aus vorherigen Übungen.

grade

Anhand der Übungen am Ende des Kapitels können Sie Ihren Lernerfolg überprüfen, nachdem Sie zuvor die geführten Übungen absolviert haben. Die `grade`-Aktion weist den Befehl `lab` an, eine Liste der Bewertungskriterien mit dem Status `PASS` oder `FAIL` für jedes Kriterium anzuzeigen. Um den Status `PASS` für alle Kriterien zu erzielen, beheben Sie die Fehler, und führen Sie die `grade`-Aktion erneut aus.

finish

Früher `cleanup`. Die Abschlusslogik des Skripts löscht die nicht mehr benötigten Übungsressourcen.

Einführung

Die Übungsskripte sind auf dem Rechner `workstation` erst nach der ersten Ausführung vorhanden. Wenn Sie den Befehl `lab` mit einer gültigen Übung und Aktion ausführen, wird das Skript `lab-exercise` von der Inhaltsfreigabe des Servers `classroom` auf `/usr/local/lib` auf den Rechner `workstation` heruntergeladen. Der Befehl `lab` erstellt zwei Protokolldateien in `/var/tmp/labs` sowie das Verzeichnis, falls es noch nicht existiert. Eine der Dateien mit dem Namen `exercise` erfasst Standardausgabemeldungen, die normalerweise in Ihrem Terminal angezeigt werden. Die andere Datei mit dem Namen `exercise.err` erfasst Fehlermeldungen.

```
[student@workstation ~]$ `ls -l /usr/local/lib`  
-rwxr-xr-x. 1 root root 4131 May  9 23:38 lab-instances-cli  
-rwxr-xr-x. 1 root root 93461 May  9 23:38 labtool.cl110.shlib  
-rwxr-xr-x. 1 root root 10372 May  9 23:38 labtool.shlib  
  
[student@workstation ~]$ `ls -l /var/tmp/labs`  
-rw-r--r--. 1 root root 113 May  9 23:38 instances-cli  
-rw-r--r--. 1 root root 113 May  9 23:38 instances-cli.err
```



Anmerkung

Skripts werden von der Freigabe `http://content.example.com/courses/COURSE/RELEASE/grading-scripts` heruntergeladen, allerdings nur, wenn das Skript auf dem Rechner `workstation` noch nicht vorhanden ist. Wenn Sie ein Skript erneut herunterladen müssen, z. B. weil ein Skript in der Freigabe geändert wurde, löschen Sie das aktuelle Übungsskript manuell unter `/usr/local/lib` auf dem Rechner `workstation`, und führen Sie dann den Befehl `Lab` erneut für die Übung aus. Daraufhin wird das neuere Übungsskript von der Freigabe `grading-scripts` heruntergeladen.

Um alle aktuellen Übungsskripts auf dem Rechner `workstation` zu löschen, verwenden Sie die Option `--refresh` des Befehls `lab`. Eine Aktualisierung löscht alle Skripts in `/usr/local/lib`, jedoch nicht die Log-Dateien.

```
[student@workstation ~]$ `lab --refresh`  
[student@workstation ~]$ `ls -l /usr/local/lib`  
  
[student@workstation ~]$ `ls -l /var/tmp/labs`  
-rw-r--r--. 1 root root 113 May  9 23:38 instances-cli  
-rw-r--r--. 1 root root 113 May  9 23:38 instances-cli.err
```

Übungsskripts senden Ausgaben an Log-Dateien, selbst wenn die Skripts erfolgreich sind. Der Header-Text zum jeweiligen Schritt wird zwischen den Schritten hinzugefügt, und zusätzliche Datums- und Uhrzeit-Header werden beim Start der jeweiligen Skriptausführung eingefügt. Das Übungsprotokoll enthält normalerweise Meldungen zum erfolgreichen Abschluss von Befehlsschritten. Daher ist das Ausgabeprotokoll bei der Übung hilfreich, falls keine Probleme auftreten, bietet jedoch keine zusätzliche Hilfe im Fehlerfall.

Das Übungsfehler-Log ist für die Fehlerbehebung nützlicher. Auch wenn die Skripts erfolgreich sind, werden Meldungen weiterhin an das Übungsfehler-Log gesendet. Beispielsweise sollte ein Skript, das verifiziert, ob ein Objekt bereits vorhanden ist, bevor es versucht, es zu erstellen, die Meldung `object not found` erzeugen, wenn das Objekt noch nicht vorhanden ist. In diesem Szenario wird diese Meldung erwartet und weist nicht auf einen Fehler hin. Tatsächliche Fehlermeldungen sind in der Regel ausführlicher, und erfahrene Systemadministratoren sollten häufige Protokolleinträge kennen.

Obwohl Übungsskripts immer auf dem Rechner `workstation` ausgeführt werden, führen sie Aufgaben auf anderen Systemen in der Kursumgebung aus. Viele Kursumgebungen, einschließlich OpenStack und OpenShift, verwenden eine Befehlszeilenschnittstelle (CLI), die vom Rechner `workstation` aufgerufen wird, um mit Serversystemen über API-Aufrufe zu kommunizieren. Da Skriptaktionen in der Regel Aufgaben an mehrere Systeme verteilen, ist eine zusätzliche Fehlerbehebung erforderlich, um festzustellen, wo eine Aufgabe fehlgeschlagen ist. Melden Sie sich bei diesen anderen Systemen an, und ermitteln Sie mithilfe Ihrer Linux-Diagnosekenntnisse in den lokalen System-Log-Dateien die Ursache des Übungsskriptfehlers.

Kapitel 1

Beschreiben von Red Hat OpenShift Container Platform

Ziel

Beschreiben der Features und Architektur von OpenShift Container Platform

Ziele

- Beschreiben der typischen Nutzung des Produkts und seiner Funktionen
- Beschreiben der Architektur der Red Hat OpenShift Container Platform
- Beschreiben von Cluster-Operatoren und deren Funktionsweise sowie Benennen der wichtigsten Cluster-Operatoren

Abschnitte

- Beschreiben der OpenShift Container Platform-Features (und Test)
- Beschreiben der OpenShift-Architektur (mit Test)
- Beschreiben von Cluster-Operatoren (mit Test)

Beschreiben von OpenShift Container Platform-Funktionen

Ziele

Nach Abschluss dieses Kapitels sollten Sie in der Lage sein, die typische Verwendung des Produkts und dessen Features zu beschreiben.

Einführung in OpenShift Container Platform

Die Container-Orchestrierung ist eine grundlegende Voraussetzung für Initiativen zur digitalen Transformation. Da monolithische Anwendungen auf containerisierte Services umgestellt werden, kann es jedoch mühsam sein, diese Anwendungen mit Legacy-Infrastruktur zu verwalten. Red Hat OpenShift Container Platform (RHOC) ermöglicht Entwicklern und IT-Abteilungen eine bessere Verwaltung des Application Lifecycle.

RHOC basiert auf dem Open Source-Projekt Kubernetes und erweitert die Plattform um Features zur Bereitstellung einer robusten, flexiblen und skalierbaren Containerplattform für Kunden-Rechenzentren, auf der Entwickler Workloads in einer hoch verfügbaren Umgebung auszuführen können.

Ein Werkzeug für die Containerorchestrierung, wie OpenShift Container Platform, verwaltet einen Cluster von Servern, in dem mehrere containerisierte Anwendungen ausgeführt werden. Die Red Hat OpenShift-Produktfamilie umfasst eine Reihe von Lösungen zur Verbesserung der Bereitstellung von Geschäftsanwendungen in einer Vielzahl von Umgebungen.

Red Hat OpenShift Container Platform

Bietet eine für den Einsatz in Unternehmen bereite Kubernetes-Umgebung für die Erstellung, Bereitstellung und Verwaltung containerbasierter Anwendungen in öffentlichen oder privaten Rechenzentren, einschließlich Bare-Metal-Servern. RHOC ist mit mehreren Cloud- und Virtualisierungsanbietern kompatibel, sodass Anwendungsentwickler und Administratoren von Unterschieden zwischen diesen Anbietern unabhängig sind. Sie entscheiden, wann Aktualisierungen auf neuere Versionen durchgeführt und welche zusätzlichen Komponenten aktiviert werden sollen.

Red Hat OpenShift Dedicated

Stellt eine verwaltete OpenShift-Umgebung in einer Public Cloud bereit, z. B. Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure oder IBM Cloud. Dieses Produkt bietet alle Features von RHOC, aber Red Hat verwaltet den Cluster für Sie. Sie behalten die Kontrolle über einige Entscheidungen, z. B. wann eine Aktualisierung auf eine neuere Version von OpenShift oder die Installation von Add-on-Services durchgeführt werden soll.

Red Hat OpenShift Online

Stellt eine gehostete, öffentliche Plattform zur Containerorchestrierung bereit, die eine Lösung für die Entwicklung, Erstellung, Bereitstellung und das Hosting von Anwendungen in einer Cloud-Umgebung bietet. Die Lösung wird von mehreren Kunden gemeinsam genutzt, und Red Hat verwaltet den Cluster-Lebenszyklus, einschließlich der Anwendung von Updates oder der Integration neuer Features.

Red Hat OpenShift Kubernetes Engine

Enthält einen Teil der Features der OpenShift Container Platform, wie z. B. das kompakte transaktionale Red Hat Enterprise Linux CoreOS-Betriebssystem, die CRI-O-Engine, die

Kapitel 1 | Beschreiben von Red Hat OpenShift Container Platform

Kubernetes-Container-Orchestrationsplattform und die zentralen Cluster-Services (Web Console, Over-the-Air-Updates, interne Registry, Operator Lifecycle Manager usw.).

Red Hat Code Ready Containers

Bietet eine Minimalinstallation von OpenShift, die Sie auf einem Laptop zum lokalen Entwickeln und Experimentieren ausführen können.

Einige Cloud-Anbieter bieten auch Lösungen auf Basis von RHOC, die eine enge Integration mit anderen Services ihrer Plattformen ermöglichen und vom Anbieter in Partnerschaft mit Red Hat unterstützt werden. Ein Beispiel hierfür ist Microsoft Azure Red Hat OpenShift.

In der folgenden Abbildung sind die Services und Features von OpenShift dargestellt:

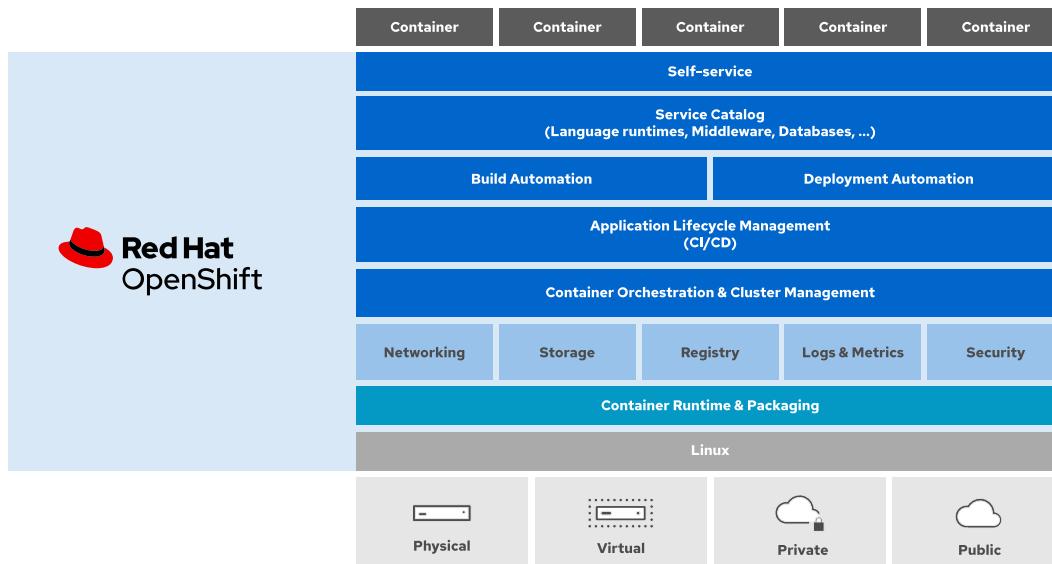


Abbildung 1.1: Services und Features von OpenShift

Die Red Hat OpenShift-Produktfamilie integriert viele Komponenten:

- Das für Container optimierte, stabile Betriebssystem Red Hat Enterprise Linux CoreOS
- Die CRI-O-Engine, eine mit der Open Container Initiative (OCI) konforme Container-Laufzeit-Engine mit reduzierter Angriffsfläche und geringem Footprint
- Kubernetes, eine Open Source-Plattform zur Container-Orchestrierung
- Eine Self-Service-Web-Console
- Eine Reihe vorinstallierter Anwendungsservices, z. B. eine interne Container-Image-Registry und ein Monitoring-Framework
- Zertifizierte Container-Images für mehrere Laufzeiten-Programmiersprachen, Datenbanken und andere Softwarepakete

Einführung in OpenShift-Features

OpenShift bietet viele Features zum Automatisieren, Skalieren und Warten Ihrer Anwendungen. Alle diese Features sind von Kubernetes aktiviert, und die meisten davon benötigen zusätzliche Komponenten, die Sie für eine eigene (Build-Your-Own, BYO) Kubernetes-Einrichtung hinzufügen und konfigurieren müssen.

Kapitel 1 | Beschreiben von Red Hat OpenShift Container Platform

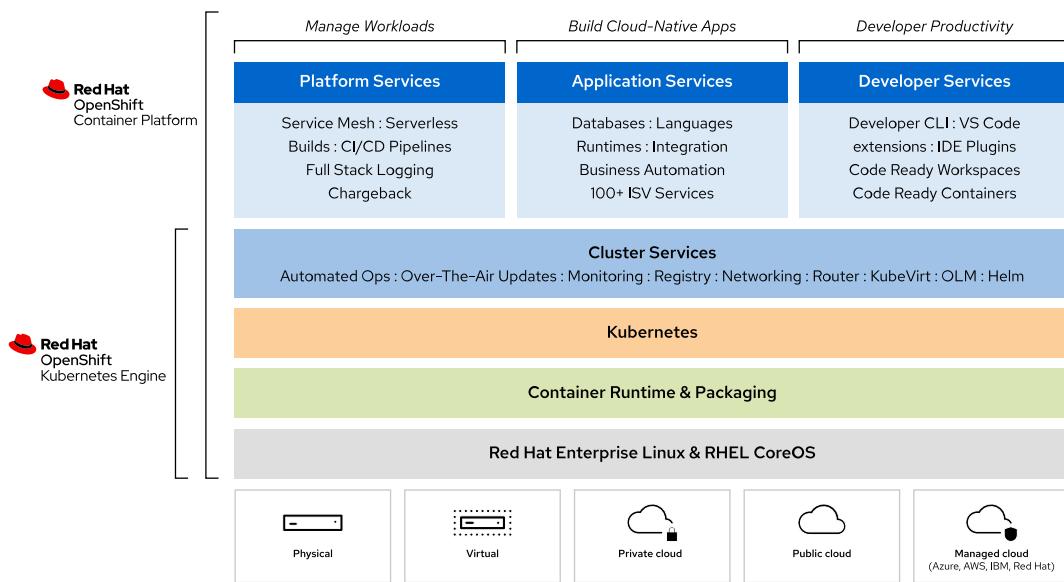


Abbildung 1.2: Funktionsvergleich zwischen OpenShift Container Platform und OpenShift Kubernetes Engine

Hochverfügbarkeit

Kubernetes wurde sowohl für interne Komponenten als auch für Benutzeranwendungen im Hinblick auf Hochverfügbarkeit konzipiert. Ein hoch verfügbarer Etcd-Cluster speichert den Status des OpenShift-Clusters und seiner Anwendungen. In Etcd gespeicherte Ressourcen, z. B. Bereitstellungskonfigurationen, bieten einen automatischen Neustart von Containern, um sicherzustellen, dass die Anwendung immer ausgeführt wird und fehlerhafte Container beendet werden. Dies gilt nicht nur für Ihre Anwendungen, sondern auch für containerisierte Services, aus denen sich der Cluster zusammensetzt, z. B. die Web Console und die interne Image-Registry.

Schlankes Betriebssystem

RHOCP wird auf Red Hat Enterprise Linux CoreOS ausgeführt, dem schlanken Betriebssystem von Red Hat, dessen Schwerpunkt auf Agilität, Portabilität und Sicherheit liegt.

Red Hat Enterprise Linux CoreOS (RHEL CoreOS) ist ein stabiles Betriebssystem, das für die Ausführung von Containeranwendungen optimiert ist. Das gesamte Betriebssystem wird anstatt paketweise als einzelnes Image aktualisiert, und sowohl Benutzeranwendungen als auch Systemkomponenten wie Netzwerkservices werden als Container ausgeführt.

RHOCP steuert Updates für RHEL CoreOS und seine Konfigurationen, sodass die Verwaltung eines OpenShift-Clusters die Verwaltung des Betriebssystems auf Clusterknoten umfasst. Systemadministratoren werden dadurch von diesen Aufgaben befreit, und das Risiko menschlicher Fehler wird verringert.

Load Balancing

Cluster bieten drei Arten von Load Balancing: Einen externen Load Balancer, der den Zugriff auf die OpenShift-API verwaltet, den HAProxy Load Balancer für den externen Zugriff auf Anwendungen und den internen Load Balancer, der Netfilter-Regeln für den internen Zugriff auf Anwendungen und Services verwendet.

Routenressourcen verwenden HAProxy für die Verwaltung des externen Zugriffs auf den Cluster. Service-Ressourcen verwenden Netfilter-Regeln, um den Datenverkehr innerhalb des Clusters zu verwalten. Die Technologie, die externe Load Balancer verwenden, hängt vom Cloud-Anbieter ab, der Ihren Cluster ausführt.

Automatisieren der Skalierung

OpenShift-Cluster können in Echtzeit an den erhöhten Anwendungsdatenverkehr angepasst werden, indem neue Container automatisch gestartet und Container beendet werden, wenn die Last sinkt. Diese Features stellen sicher, dass die Zugriffszeit Ihrer Anwendung unabhängig von der Anzahl der gleichzeitigen Verbindungen oder Aktivitäten optimal bleibt.

OpenShift-Cluster können je nach der aggregierten Last von vielen Anwendungen dem Cluster auch weitere Computing-Knoten hinzufügen oder daraus entfernen, sodass Reaktionsfähigkeit und Kosten für Public und Private Clouds niedrig gehalten werden.

Protokollierung und Monitoring

RHOCP wird mit einer fortschrittlichen Monitoring-Lösung ausgeliefert, die auf Prometheus basiert und Hunderte von Metriken über Ihren Cluster sammelt. Diese Lösung interagiert mit einem Alarmsystem, von dem Sie detaillierte Informationen über Aktivität und Zustand Ihres Cluster erhalten.

RHOCP wird mit einer erweiterten aggregierten Protokollierungslösung ausgeliefert, die auf Elasticsearch basiert und die langfristige Speicherung von Protokollen von Clusterknoten und Containern ermöglicht.

Service-Erkennung

RHOCP führt einen internen DNS-Service auf dem Cluster aus und konfiguriert alle Container für die Verwendung dieses internen DNS für die Namensauflösung. Das bedeutet, dass Anwendungen benutzerfreundliche Namen nutzen können, um andere Anwendungen und Services zu suchen, ohne einen externen Services-Katalog verwenden zu müssen.

Storage

Kubernetes fügt eine Abstraktionsschicht zwischen dem Storage-Back-End und der Storage-Nutzung hinzu. Auf diese Weise können Anwendungen langlebigen, kurzlebigen, block- und dateibasierten Storage mit einheitlichen Storage-Definitionen nutzen, die vom Storage-Back-End unabhängig sind. Ihre Anwendungen hängen so nicht von Storage-APIs bestimmter Cloud-Anbieter ab.

RHOCP integriert eine Reihe von Storage-Anbietern, die eine automatische Bereitstellung von Storage auf gängigen Cloud-Anbietern und Virtualisierungsplattformen ermöglichen. Daher müssen Cluster-Administratoren die genauen Details proprietärer Storage Arrays nicht verwalten.

Anwendungsmanagement

Mit RHOCP können Entwickler die Entwicklung und Bereitstellung ihrer Anwendungen automatisieren. Verwenden Sie das OpenShift-Feature Source-to-Image (S2I), um Container auf Grundlage Ihres Quellcodes automatisch zu erstellen und sie in OpenShift auszuführen. Die interne Registry speichert Anwendungs-Container-Images, die wieder verwendet werden können. Dadurch wird die Zeit bis zur Veröffentlichung Ihrer Anwendungen verkürzt.

Der Entwicklerkatalog, auf den Sie über die Web Console zugreifen können, ist ein Ort für die Veröffentlichung von und den Zugriff auf Anwendungsvorlagen. Er unterstützt viele

Kapitel 1 | Beschreiben von Red Hat OpenShift Container Platform

Laufzeitsprachen wie Python, Ruby, Java und Node.js sowie Datenbank- und Messaging-Server. Sie können den Katalog durch die Installation neuer Operatoren erweitern, bei denen es sich um vorpaketierte Anwendungen und Services handelt, die operative Intelligenz für Bereitstellung, Aktualisierung und Monitoring Ihrer Anwendungen integrieren.

Cluster-Erweiterbarkeit

RHOPC basiert auf standardmäßigen Erweiterungsmechanismen von Kubernetes, z. B. Erweiterungs-APIs und benutzerdefinierte Ressourcendefinitionen, um Features hinzuzufügen, die andernfalls nicht durch das vorgelagerte Kubernetes bereitgestellt werden. OpenShift paketiert diese Erweiterungen als Operatoren, um Installation, Aktualisierung und Verwaltung zu vereinfachen.

OpenShift beinhaltet auch den Operator Lifecycle Manager (OLM), der die Erkennung, Installation und Aktualisierung von Anwendungen und Infrastrukturkomponenten erleichtert, die als Operator paketiert sind.

Red Hat hat in Zusammenarbeit mit AWS, Google Cloud und Microsoft den unter <https://operatorhub.io> verfügbaren OperatorHub auf den Markt gebracht. Die Plattform ist ein öffentliches Repository und ein Marktplatz für Operatoren, die mit OpenShift und anderen Distributionen von Kubernetes kompatibel sind, die den OLM beinhalten.

Der Red Hat Marketplace ist eine Plattform für den Zugriff auf zertifizierte Software als Kubernetes-Operatoren, die in einem OpenShift-Cluster bereitgestellt werden können. Die zertifizierte Software umfasst automatische Bereitstellungen und nahtlose Upgrades für ein integriertes Erlebnis.



Literaturhinweise

Weitere Informationen finden Sie in der Produktdokumentation zu Red Hat OpenShift Container Platform 4.6 unter
https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/

Red Hat OpenShift Kubernetes Engine

<https://www.openshift.com/products/kubernetes-engine>

► Quiz

Beschreiben von OpenShift Container Platform-Funktionen

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Durch welche der folgenden Definitionen werden Plattformen zur Container-Orchestrierung am besten beschrieben?
- a. Sie erweitern das operative Wissen Ihrer Anwendung und bieten eine Möglichkeit, diese zu paketieren und zu verteilen.
 - b. Sie ermöglichen die Verwaltung eines Clusters von Servern, auf denen containerisierte Anwendungen ausgeführt werden. Sie fügen Features wie Self-Service, Hochverfügbarkeit, Überwachung und Automatisierung hinzu.
 - c. Sie ermöglichen die Bereitstellung von Infrastructure-as-a-Service-Clustern auf einer Vielzahl von Cloud-Anbietern, darunter AWS, GCP und Microsoft Azure.
 - d. Sie ermöglichen es Entwicklern, ihre Anwendungen als Operatoren zu schreiben, zu paketieren und im Operatorkatalog zu veröffentlichen.
- 2. Welche drei der folgenden Schlüsselfunktionen ermöglichen die Hochverfügbarkeit Ihrer Anwendungen? (Wählen Sie drei Antworten aus.)
- a. Ein OpenShift-Etcd-Cluster stellt den Clusterstatus für alle Knoten zur Verfügung.
 - b. HAProxy-Load-Balancer von OpenShift ermöglichen den externen Zugriff auf Anwendungen.
 - c. OpenShift-Services führen das Load Balancing beim Zugriff auf Anwendungen im Cluster aus.
 - d. OpenShift-Bereitstellungskonfigurationen stellen sicher, dass Anwendungscontainer bei Szenarien wie dem Verlust eines Knotens neu gestartet werden.
- 3. Welche zwei der folgenden Aussagen über OpenShift sind richtig? (Wählen Sie zwei Antworten aus.)
- a. Entwickler können Cloud-Anwendungen direkt aus einem Quellcode-Repository erstellen und starten.
 - b. OpenShift patcht Kubernetes zum Hinzufügen von Features, die in anderen Distributionen von Kubernetes nicht zur Verfügung stehen.
 - c. OpenShift Dedicated bietet den Zugriff auf eine exklusive Gruppe von Operatoren, die von Red Hat zusammengestellt und gewartet werden. Dies trägt dazu bei, dass die Operatoren sicher sind und zuverlässig in Ihrer Umgebung arbeiten.
 - d. OpenShift-Cluster-Administratoren können neue Operatoren aus dem Operator-Katalog ermitteln und installieren.

- 4. Welche zwei der folgenden Services verwenden OpenShift-Komponenten für das Load Balancing ihres Datenverkehrs? (Wählen Sie zwei Antworten aus.)
- a. Die OpenShift-API, auf die über den externen Load Balancer zugegriffen werden kann
 - b. Services, die Netfilter für das Load Balancing verwenden
 - c. Services, die HAProxy für das Load Balancing verwenden
 - d. Routen, die Netfilter für das Load Balancing verwenden
 - e. Routen, die HAProxy für das Load Balancing verwenden
- 5. Welche zwei der folgenden Aussagen über OpenShift-Hochverfügbarkeit und - Skalierbarkeit sind richtig? (Wählen Sie zwei Antworten aus.)
- a. OpenShift stellt Hochverfügbarkeit nicht standardmäßig bereit. Sie müssen Hochverfügbarkeitsprodukte von Drittanbietern verwenden.
 - b. OpenShift verwendet Metriken von Prometheus, um Anwendungs-Pods dynamisch zu skalieren.
 - c. Hochverfügbarkeit und Skalierbarkeit sind auf Anwendungen beschränkt, die eine REST-API bereitstellen.
 - d. OpenShift kann bei Bedarf Anwendungen hoch- und herunterskalieren.

► Lösung

Beschreiben von OpenShift Container Platform-Funktionen

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Durch welche der folgenden Definitionen werden Plattformen zur Container-Orchestrierung am besten beschrieben?
- a. Sie erweitern das operative Wissen Ihrer Anwendung und bieten eine Möglichkeit, diese zu paketieren und zu verteilen.
 - b. Sie ermöglichen die Verwaltung eines Clusters von Servern, auf denen containerisierte Anwendungen ausgeführt werden. Sie fügen Features wie Self-Service, Hochverfügbarkeit, Überwachung und Automatisierung hinzu.
 - c. Sie ermöglichen die Bereitstellung von Infrastructure-as-a-Service-Clustern auf einer Vielzahl von Cloud-Anbietern, darunter AWS, GCP und Microsoft Azure.
 - d. Sie ermöglichen es Entwicklern, ihre Anwendungen als Operatoren zu schreiben, zu paketieren und im Operatorkatalog zu veröffentlichen.
- 2. Welche drei der folgenden Schlüsselfunktionen ermöglichen die Hochverfügbarkeit Ihrer Anwendungen? (Wählen Sie drei Antworten aus.)
- a. Ein OpenShift-Etcd-Cluster stellt den Clusterstatus für alle Knoten zur Verfügung.
 - b. HAProxy-Load-Balancer von OpenShift ermöglichen den externen Zugriff auf Anwendungen.
 - c. OpenShift-Services führen das Load Balancing beim Zugriff auf Anwendungen im Cluster aus.
 - d. OpenShift-Bereitstellungskonfigurationen stellen sicher, dass Anwendungscontainer bei Szenarien wie dem Verlust eines Knotens neu gestartet werden.
- 3. Welche zwei der folgenden Aussagen über OpenShift sind richtig? (Wählen Sie zwei Antworten aus.)
- a. Entwickler können Cloud-Anwendungen direkt aus einem Quellcode-Repository erstellen und starten.
 - b. OpenShift patcht Kubernetes zum Hinzufügen von Features, die in anderen Distributionen von Kubernetes nicht zur Verfügung stehen.
 - c. OpenShift Dedicated bietet den Zugriff auf eine exklusive Gruppe von Operatoren, die von Red Hat zusammengestellt und gewartet werden. Dies trägt dazu bei, dass die Operatoren sicher sind und zuverlässig in Ihrer Umgebung arbeiten.
 - d. OpenShift-Cluster-Administratoren können neue Operatoren aus dem Operator-Katalog ermitteln und installieren.

► **4. Welche zwei der folgenden Services verwenden OpenShift-Komponenten für das Load Balancing ihres Datenverkehrs? (Wählen Sie zwei Antworten aus.)**

- a. Die OpenShift-API, auf die über den externen Load Balancer zugegriffen werden kann
- b. Services, die Netfilter für das Load Balancing verwenden
- c. Services, die HAProxy für das Load Balancing verwenden
- d. Routen, die Netfilter für das Load Balancing verwenden
- e. Routen, die HAProxy für das Load Balancing verwenden

► **5. Welche zwei der folgenden Aussagen über OpenShift-Hochverfügbarkeit und - Skalierbarkeit sind richtig? (Wählen Sie zwei Antworten aus.)**

- a. OpenShift stellt Hochverfügbarkeit nicht standardmäßig bereit. Sie müssen Hochverfügbarkeitsprodukte von Drittanbietern verwenden.
- b. OpenShift verwendet Metriken von Prometheus, um Anwendungs-Pods dynamisch zu skalieren.
- c. Hochverfügbarkeit und Skalierbarkeit sind auf Anwendungen beschränkt, die eine REST-API bereitstellen.
- d. OpenShift kann bei Bedarf Anwendungen hoch- und herunterskalieren.

Beschreiben der OpenShift-Architektur

Ziele

Am Ende dieses Abschnitts sollten Sie in der Lage sein, die Architektur von Red Hat OpenShift Container Platform zu beschreiben.

Einführung in die deklarative Architektur von Kubernetes

Die Architektur von OpenShift basiert auf dem deklarativen Aufbau von Kubernetes. Die meisten Systemadministratoren sind an imperative Architekturen gewöhnt. In diesen Architekturen werden Aktionen durchgeführt, die indirekt den Status des Systems ändern, z. B. Starten und Stoppen von Containern auf einem bestimmten Server. In einer deklarativen Architektur ändern Sie den Status des Systems, und das System aktualisiert sich so, dass es dem neuen Status entspricht. Mit Kubernetes definieren Sie beispielsweise eine Pod-Ressource, die angibt, dass ein bestimmter Container unter spezifischen Bedingungen ausgeführt werden soll. Anschließend sucht Kubernetes einen Server (einen Knoten), der diesen Container unter diesen spezifischen Bedingungen ausführen kann.

Deklarative Architekturen ermöglichen selbst-optimierende und selbst-reparierende Systeme, die einfacher zu verwalten sind als imperative Architekturen.

Kubernetes definiert den Status seines Clusters, einschließlich der Menge der bereitgestellten Anwendungen, als eine Reihe von Ressourcen, die in der Etcd-Datenbank gespeichert sind. Kubernetes führt zudem Controller aus, die diese Ressourcen überwachen und mit dem aktuellen Status des Clusters vergleichen. Diese Controller treffen alle erforderlichen Maßnahmen, um den Status des Clusters mit dem Status der Ressourcen abzugleichen, z. B. indem ein Knoten mit ausreichender CPU-Kapazität gesucht wird, um einen neuen Container aus einer neuen Pod-Ressource zu starten.

Kubernetes bietet eine REST-API zur Verwaltung dieser Ressourcen. Alle Aktionen, die ein OpenShift-Benutzer in der Befehlszeilenschnittstelle oder der Web Console ausführt, werden durch Aufrufe dieser REST-API durchgeführt.

Einführung in die OpenShift Control Plane

Ein Kubernetes-Cluster besteht aus einer Reihe von Knoten, die den Systemservice kubelet und eine Container-Engine ausführen. OpenShift führt ausschließlich die Container-Engine CRI-O aus. Einige Knoten sind Knoten der Control Plane, auf denen die REST-API, die etcd-Datenbank und die Plattform-Controller ausgeführt werden. OpenShift konfiguriert die Knoten der Control Plane so, dass sie nicht für die Ausführung von Endbenutzer-Anwendungs-Pods zugeordnet werden können und ausschließlich für die Ausführung der Control Plane Services bestimmt sind. OpenShift plant die Ausführung der Endbenutzer-Anwendungs-Pods auf den Computing-Knoten.

Die folgende Grafik zeigt eine Übersicht über einen OpenShift-Knoten der Control Plane zusammen mit den wichtigsten Prozessen, die auf einem regulären Knoten und einem Knoten der Control Plane ausgeführt werden, entweder als Systemservices oder als Container.

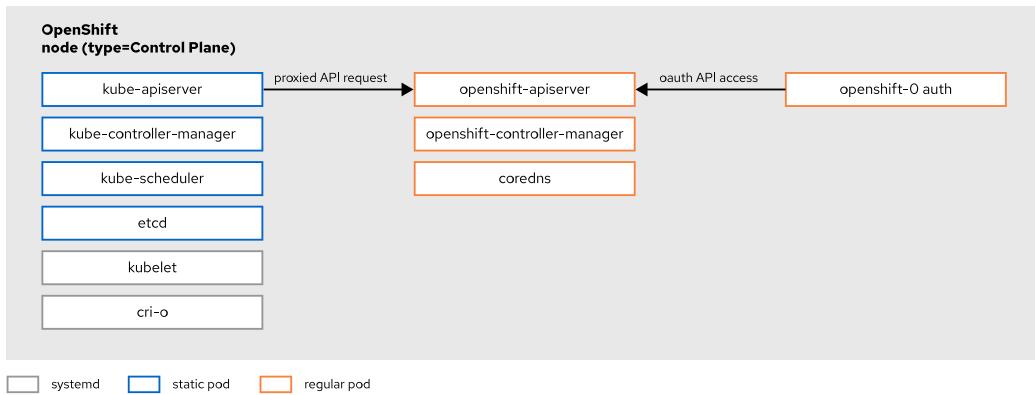


Abbildung 1.3: Architektur eines OpenShift-Knotens der Control Plane

Depending on the node settings, the `kubelet` agent starts different sets of static pods. Statische Pods sind Pods, für deren Start keine Verbindung zum API-Server erforderlich ist. Der `kubelet`-Agent verwaltet den Lebenszyklus des Pods. Statische Pods können entweder Control Plane Services wie den Scheduler oder Knotenservices bereitstellen, z. B. Software-Defined Networking (SDN). OpenShift stellt Operatoren bereit, die Pod-Ressourcen für diese statischen Pods erstellen, sodass sie wie reguläre Pods überwacht werden.

Beschreiben von OpenShift-Erweiterungen

Viele Funktionen von Kubernetes hängen von externen Komponenten ab, wie z. B. Ingress-Controller, Storage-Plugins, Netzwerk-Plugins und Authentifizierungs-Plugins. Durch Auswählen verschiedener Komponenten gibt es ähnlich wie bei Linux-Distributionen viele Möglichkeiten, eine Kubernetes-Distribution zu erstellen.

Viele Funktionen von Kubernetes hängen auch von Erweiterungs-APIs ab, z. B. Zugriffskontrolle und Netzwerkisolation.

OpenShift ist eine Kubernetes-Distribution, die viele dieser Komponenten bereitstellt, die bereits integriert und konfiguriert sind und von Operatoren verwaltet werden. OpenShift stellt auch vorinstallierte Anwendungen bereit, z. B. eine Container-Image-Registry und eine Web Console, die von Operatoren verwaltet werden.

OpenShift ergänzt Kubernetes zudem um eine Reihe von Erweiterungs-APIs und benutzerdefinierten Ressourcen. Beispielsweise zum Erstellen von Konfigurationen für den Source-to-Image-Prozess und für das Routing von Ressourcen, um den externen Zugriff auf den Cluster zu verwalten.

Red Hat entwickelt alle Erweiterungen als Open Source-Projekte und arbeitet mit der größeren Kubernetes-Community zusammen, um nicht nur diese offiziellen Komponenten von Kubernetes zu erstellen, sondern auch die Kubernetes-Plattform weiterzuentwickeln, mit dem Ziel einer einfacheren Verwaltbarkeit und Anpassung.

Bei OpenShift 3 waren diese Erweiterungen manchmal Patches (oder Forks) des vorgelagerten Kubernetes. Bei OpenShift 4 und Operatoren sind diese Erweiterungen standardmäßige Kubernetes-Erweiterungen, die jeder Distribution von Kubernetes hinzugefügt werden können.

Einführung in die Standard-Storage-Klasse von OpenShift

Im Gegensatz zu vielen Containerplattformen mit dem Schwerpunkt auf cloudnativen, zustandslosen Anwendungen unterstützt OpenShift auch zustandsbehaftete Anwendungen, die nicht der standardmäßigen Zwölf-Faktoren-App-Methodik entsprechen. OpenShift unterstützt zustandsbehaftete Anwendungen, indem es einen umfassenden Satz an Storage-Funktionen und unterstützenden Operatoren bietet. OpenShift wird mit integrierten Storage-Plugins und Storage-Klassen ausgeliefert, die sich auf die zugrunde liegende Cloud- oder Virtualisierungsplattform stützen, um dynamisch bereitgestellten Speicher bereitzustellen.

Wenn Sie OpenShift beispielsweise auf Amazon Web Services (AWS) installieren, wird Ihr OpenShift-Cluster mit einer standardmäßigen Storage-Klasse vorkonfiguriert, die den Amazon Elastic Block Store (EBS)-Service automatisch verwendet, um Storage-Volumes bei Bedarf bereitzustellen. Benutzer können eine Anwendung bereitstellen, die persistenten Storage erfordert, z. B. eine Datenbank, und OpenShift erstellt automatisch ein EBS-Volume, um die Anwendungsdaten zu hosten.

OpenShift-Cluster-Administratoren können später zusätzliche Storage-Klassen definieren, die unterschiedliche EBS-Service-Ebenen verwenden. Beispielsweise könnten Sie eine Storage-Klasse für Hochleistungs-Storage mit einer hohen IOPS-Rate (Input-Output-Operationen pro Sekunde) und eine weitere Storage-Klasse für kostengünstigen Storage mit niedriger Leistung definieren. Cluster-Administratoren können dann zulassen, dass nur bestimmte Anwendungen die Hochleistungs-Storage-Klasse verwenden, und Anwendungen für die Datenarchivierung so konfigurieren, dass die Storage-Klasse mit niedriger Leistung verwendet wird.



Literaturhinweise

Weitere Informationen finden Sie in der Produktdokumentation zu Red Hat OpenShift Container Platform 4.6 unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/

Die Zwölf-Faktoren-App

<https://12factor.net/>

► Quiz

Beschreiben der OpenShift-Architektur

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

► 1. Auf welcher der folgenden Technologien zur Container-Orchestrierung basiert OpenShift?

- a. Docker Swarm
- b. Rancher
- c. Kubernetes
- d. Mesosphere Marathon
- e. CoreOS Fleet

► 2. Welche zwei der folgenden Aussagen über OpenShift Container Platform sind richtig?
(Wählen Sie zwei Antworten aus.)

- a. OpenShift stellt einen OAuth-Server bereit, der Aufrufe seiner REST-API authentifiziert.
- b. Für OpenShift ist die Container-Engine CRI-O erforderlich.
- c. Kubernetes folgt einer deklarativen Architektur, OpenShift folgt dagegen einer eher traditionellen imperativen Architektur.
- d. OpenShift-Erweiterungs-APIs werden als Systemservices ausgeführt.

► 3. Auf welchem der folgenden Server werden Kubernetes-API-Komponenten ausgeführt?

- a. Server-Knoten
- b. Knoten
- c. Knoten der Control Plane

► 4. Um welche der folgenden Komponenten wird das vorgelagerte Kubernetes durch OpenShift ergänzt?

- a. Die Etcd-Datenbank
- b. Eine Container-Engine
- c. Einen Registry Server
- d. Einen Scheduler
- e. Das Kubelet

► **5. Welcher der folgenden Sätze trifft auf die Unterstützung von Storage mit OpenShift zu?**

- a. Benutzer können persistente Daten nur in der Etcd-Datenbank speichern.
- b. Benutzer können nur auf Cloud-nativen Anwendungen von OpenShift bereitstellen, die der Zwölf-Faktoren-App-Methodik entsprechen.
- c. Administratoren müssen Storage-Plugins konfigurieren, die für ihre Cloud-Anbieter geeignet sind.
- d. Bevor ein Benutzer Anwendungen bereitstellen kann, die persistenten Storage erfordern, müssen Administratoren persistente Volumes definieren.
- e. Benutzer können Anwendungen bereitstellen, die persistenten Storage erfordern, indem sie sich auf die standardmäßige Storage-Klasse verlassen.

► Lösung

Beschreiben der OpenShift-Architektur

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

► 1. Auf welcher der folgenden Technologien zur Container-Orchestrierung basiert OpenShift?

- a. Docker Swarm
- b. Rancher
- c. Kubernetes
- d. Mesosphere Marathon
- e. CoreOS Fleet

► 2. Welche zwei der folgenden Aussagen über OpenShift Container Platform sind richtig?
(Wählen Sie zwei Antworten aus.)

- a. OpenShift stellt einen OAuth-Server bereit, der Aufrufe seiner REST-API authentifiziert.
- b. Für OpenShift ist die Container-Engine CRI-O erforderlich.
- c. Kubernetes folgt einer deklarativen Architektur, OpenShift folgt dagegen einer eher traditionellen imperativen Architektur.
- d. OpenShift-Erweiterungs-APIs werden als Systemservices ausgeführt.

► 3. Auf welchem der folgenden Server werden Kubernetes-API-Komponenten ausgeführt?

- a. Server-Knoten
- b. Knoten
- c. Knoten der Control Plane

► 4. Um welche der folgenden Komponenten wird das vorgelagerte Kubernetes durch OpenShift ergänzt?

- a. Die Etcd-Datenbank
- b. Eine Container-Engine
- c. Einen Registry Server
- d. Einen Scheduler
- e. Das Kubelet

► **5. Welcher der folgenden Sätze trifft auf die Unterstützung von Storage mit OpenShift zu?**

- a. Benutzer können persistente Daten nur in der Etcd-Datenbank speichern.
- b. Benutzer können nur auf Cloud-nativen Anwendungen von OpenShift bereitstellen, die der Zwölf-Faktoren-App-Methodik entsprechen.
- c. Administratoren müssen Storage-Plugins konfigurieren, die für ihre Cloud-Anbieter geeignet sind.
- d. Bevor ein Benutzer Anwendungen bereitstellen kann, die persistenten Storage erfordern, müssen Administratoren persistente Volumes definieren.
- e. Benutzer können Anwendungen bereitstellen, die persistenten Storage erfordern, indem sie sich auf die standardmäßige Storage-Klasse verlassen.

Beschreiben von Cluster-Operatoren

Ziele

Am Ende dieses Abschnitts sollten Sie beschreiben können, was ein Cluster-Operator ist und wie er funktioniert. Außerdem sollten Sie die wichtigsten Cluster-Operatoren kennen.

Einführung in Kubernetes-Operatoren

Kubernetes-Operatoren sind Anwendungen, die die Kubernetes-API aufrufen, um Kubernetes-Ressourcen zu verwalten. Wie bei jeder Kubernetes-Anwendung stellen Sie einen Operator bereit, indem Sie Kubernetes-Ressourcen wie Services und Bereitstellungen definieren, die auf das Container-Image des Operators verweisen. Da Operatoren im Gegensatz zu gängigen Anwendungen direkten Zugriff auf die Kubernetes-Ressourcen benötigen, erfordern sie in der Regel benutzerdefinierte Sicherheitseinstellungen.

Operatoren definieren in der Regel benutzerdefinierte Ressourcen (CR), die ihre Einstellungen und Konfigurationen speichern. OpenShift-Administratoren können Operatoren verwalten, indem sie deren benutzerdefinierte Ressourcen bearbeiten. Die Syntax einer benutzerdefinierten Ressource wird durch eine benutzerdefinierte Ressourcendefinition (CRD) definiert.

Die meisten Operatoren verwalten eine andere Anwendung. So kann ein Operator beispielsweise einen Datenbankserver verwalten. In diesem Fall erstellt der Operator die Ressourcen, die diese andere Anwendung mit den Informationen aus der benutzerdefinierten Ressource beschreiben.

Der Zweck eines Operators besteht in der Regel darin, Aufgaben zu automatisieren, die ein menschlicher Administrator (oder ein menschlicher Operator) für die Bereitstellung, Aktualisierung und Verwaltung einer Anwendung durchführt.

Einführung in das Operator-Framework

Sie können Operatoren mit Ihrer bevorzugten Programmiersprache entwickeln. Technisch gesehen benötigen Sie kein spezielles SDK für die Entwicklung eines Operators. Alles, was Sie benötigen, ist die Möglichkeit, REST-APIs aufzurufen und Secrets zu nutzen, die Anmelddaten für die Kubernetes-APIs enthalten.

Das Operator-Framework ist ein Open-Source-Toolkit zum Erstellen, Testen und Paketieren von Operatoren. Das Operator-Framework erleichtert diese Aufgaben, indem es die folgenden Komponenten bereitstellt, statt direkt auf Low-Level-Kubernetes-APIs zu programmieren:

Operator Software Development Kit (Operator SDK)

Stellt eine Reihe von GoLang-Bibliotheken und Quellcodebeispielen bereit, in denen gängige Patterns in Operator-Anwendungen implementiert werden. Darüber hinaus bietet es ein Container-Image und Playbook-Beispiele, mit denen Sie Operatoren mit Ansible entwickeln können.

Operator Lifecycle Manager (OLM)

Stellt eine Anwendung bereit, welche die Bereitstellung, die Ressourcennutzung, die Aktualisierungen und das Löschen von Operatoren verwaltet, die über einen Operator-Katalog bereitgestellt wurden. Der OLM selbst ist ein Operator, der mit OpenShift vorinstalliert wird.

Kapitel 1 | Beschreiben von Red Hat OpenShift Container Platform

Das Operator-Framework definiert außerdem eine Reihe empfohlener Vorgehensweisen für die Implementierung von Operatoren und CRDs sowie eine Standardmethode zur Paketierung eines Operator-Manifests als Container-Image, mit dem ein Operator mithilfe eines Operator-Katalogs verteilt werden kann. Die häufigste Form eines Operator-Katalogs ist ein Image-Registry-Server.

Ein Operator-Container-Image, das den Operator-Framework-Standards folgt, enthält alle für die Bereitstellung der Operator-Anwendung erforderlichen Ressourcendefinitionen. Auf diese Weise kann der OLM einen Operator automatisch installieren. Wenn ein Operator nicht gemäß den Operator-Framework-Standards erstellt und paketiert ist, kann der OLM diesen Operator weder installieren noch verwalten.

Einführung in OperatorHub

OperatorHub stellt eine Weboberfläche bereit, um Operatoren zu ermitteln und zu veröffentlichen, die den Operator-Framework-Standards entsprechen. Sowohl Open-Source- als auch kommerzielle Operatoren können im Operator-Hub veröffentlicht werden. Operator-Container-Images können in unterschiedlichen Image-Registries gehostet werden, z. B. quay.io.

Einführung in den Red Hat Marketplace

Der Red Hat Marketplace bietet Zugriff auf eine kuratierte Reihe von Operatoren für Unternehmen, die in einem OpenShift- oder Kubernetes-Cluster bereitgestellt werden können. Die im Red Hat Marketplace verfügbaren Operatoren haben einen Zertifizierungsprozess durchlaufen, um sicherzustellen, dass die Software den Best Practices folgt und dass auch die Container auf Sicherheitslücken überprüft werden.

Der Red Hat Marketplace Operator bietet eine nahtlose Integration zwischen einem OpenShift-Cluster und dem Red Hat Marketplace. Diese Integration verwaltet Aktualisierungen und konsolidiert Abrechnung und Berichterstattung, um die Bereitstellung zertifizierter Operatoren zu vereinfachen. Anbieter bieten verschiedene Preisoptionen für ihre Operatoren an, z. B. kostenlose Testversionen, verschiedene Editionen und Rabatte für Großkunden.

Einführung in OpenShift-Cluster-Operatoren

Cluster-Operatoren sind reguläre Operatoren, mit dem Unterschied, dass Sie nicht vom OLM verwaltet werden. Sie werden vom Cluster-Version-Operator von OpenShift verwaltet, der manchmal als First-Level-Operator bezeichnet wird. Alle Cluster-Operatoren werden auch als Second-Level-Operatoren bezeichnet.

OpenShift-Cluster-Operatoren bieten OpenShift-Extension-APIs und Infrastruktur-Services wie:

- OAuth-Server, der den Zugriff auf die Control Plane- und Extension-APIs authentifiziert.
- Zentraler DNS-Server, der die Service-Ermittlung innerhalb des Clusters verwaltet.
- Web-Konsole, welche die grafische Verwaltung des Clusters ermöglicht.
- Interne Image-Registry, die es Entwicklern ermöglicht, Container-Images innerhalb des Clusters zu hosten, wobei entweder S2I oder ein anderer Mechanismus verwendet wird.
- Monitoring-Stack, der Metriken und Alarne über den Cluster-Zustand generiert.

Einige Cluster-Operatoren verwalten Knoten- oder Control Plane-Einstellungen. Mit vorgelegtem Kubernetes können Sie beispielsweise eine Knoten-Konfigurationsdatei bearbeiten, um Storage- und Netzwerk-Plugins hinzuzufügen, und für diese Plugins sind möglicherweise zusätzliche Konfigurationsdateien erforderlich. OpenShift unterstützt Operatoren, die Konfigurationsdateien auf allen Knoten verwalten, und lädt die Knoten-Services neu, die von Änderungen an diesen Dateien betroffen sind.



Wichtig

OpenShift 4 kündigt die Verwendung von SSH-Sitzungen zum Verwalten von Knoten-Konfigurationen und Systemservices ab. Dadurch wird sichergestellt, dass Sie die Knoten nicht anpassen und sie sicher zu einem Cluster hinzugefügt oder daraus entfernt werden können. Es wird erwartet, dass Sie alle administrativen Aktionen indirekt durchführen, indem Sie benutzerdefinierte Ressourcen bearbeiten und dann warten, bis ihre jeweiligen Operatoren Ihre Änderungen übernehmen.

OpenShift-Cluster-Operatoren

Normalerweise teilen ein Operator und seine verwaltete Anwendung dasselbe Projekt. Im Fall von Cluster-Operatoren befinden sich diese in den Projekten von `openshift-*`. Jeder Cluster-Operator definiert eine benutzerdefinierte Ressource vom Typ `ClusterOperator`. Cluster-Operatoren verwalten den Cluster selbst und einschließlich API-Server, Web-Konsole oder Netzwerk-Stack. Jeder Cluster-Operator definiert eine Reihe von benutzerdefinierten Ressourcen, um seine Komponenten weiter zu steuern. Die `ClusterOperator`-API-Ressource macht Informationen wie den Zustand des Updates oder die Version der Komponente verfügbar.

Operatoren sind anhand ihres Namens erkennbar. So stellt der Cluster-Operator `console` die Web Console bereit, und der Cluster-Operator `ingress` ermöglicht Zugänge und Routen. In der folgenden Liste werden einige der Cluster-Operatoren aufgeführt:

- `network`
- `ingress`
- `storage`
- `authentication`
- `console`
- `monitoring`
- `image-registry`
- `cluster-autoscaler`
- `openshift-apiserver`
- `dns`
- `openshift-controller-manager`
- `cloud-credential`



Literaturhinweise

Weitere Informationen finden Sie in der Produktdokumentation zu Red Hat OpenShift Container Platform 4.6 unter
https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/

Einführung in das Operator-Framework

<https://blog.openshift.com/introducing-the-operator-framework/>

Erste Schritte mit dem Red Hat Marketplace

<https://marketplace.redhat.com/en-us/documentation/getting-started>

► Quiz

Beschreiben von Cluster-Operatoren

Ordnen Sie folgende Elemente ihren jeweiligen Gegenstücken in der Tabelle zu.

Benutzerdefinierte-Ressourcendefinitionen

Operator

Operator Lifecycle Manager (OLM)

Operator-Image

Operator-Katalog

Operator-SDK

OperatorHub

Red Hat Marketplace

Operator-Terminologie	Name
Ein Open-Source-Toolkit zum Erstellen, Testen und Paketieren von Operatoren.	
Ein Repository zum Ermitteln und Installieren von Operatoren.	
Eine Erweiterung der Kubernetes-API, welche die Syntax einer benutzerdefinierten Ressource definiert.	
Das vom Operator-Framework definierte Artefakt, das Sie zur Verwendung durch eine OLM-Instanz publizieren können.	
Eine Anwendung, die Kubernetes-Ressourcen verwaltet.	
Eine Anwendung, die Kubernetes-Operatoren verwaltet.	
Ein öffentlicher Webservice, in dem Sie Operatoren veröffentlichen können, die mit dem OLM kompatibel sind.	
Eine Plattform für den Zugriff auf zertifizierte Software als Kubernetes-Operatoren, die in einem OpenShift-Cluster bereitgestellt werden können.	

► Lösung

Beschreiben von Cluster-Operatoren

Ordnen Sie folgende Elemente ihren jeweiligen Gegenstücken in der Tabelle zu.

Operator-Terminologie	Name
Ein Open-Source-Toolkit zum Erstellen, Testen und Paketieren von Operatoren.	Operator-SDK
Ein Repository zum Ermitteln und Installieren von Operatoren.	Operator-Katalog
Eine Erweiterung der Kubernetes-API, welche die Syntax einer benutzerdefinierten Ressource definiert.	Benutzerdefinierte-Ressourcendefinitionen
Das vom Operator-Framework definierte Artefakt, das Sie zur Verwendung durch eine OLM-Instanz publizieren können.	Operator-Image
Eine Anwendung, die Kubernetes-Ressourcen verwaltet.	Operator
Eine Anwendung, die Kubernetes-Operatoren verwaltet.	Operator Lifecycle Manager (OLM)
Ein öffentlicher Webservice, in dem Sie Operatoren veröffentlichen können, die mit dem OLM kompatibel sind.	OperatorHub
Eine Plattform für den Zugriff auf zertifizierte Software als Kubernetes-Operatoren, die in einem OpenShift-Cluster bereitgestellt werden können.	Red Hat Marketplace

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Red Hat OpenShift Container Platform basiert auf dem Red Hat Enterprise Linux CoreOS, der CRI-O-Container-Engine und Kubernetes.
- RHOCP 4 bietet Services auf Kubernetes, z. B. eine interne Container-Image-Registry, Storage, Netzwerkanbieter und zentralisierte Protokollierung und Überwachung.
- Operatoren, die Kubernetes-Ressourcen verwalten, und der Operator Lifecycle Manager (OLM) verarbeiten die Installation und Verwaltung von Operatoren.
- OperatorHub.io ist ein Online-Katalog zur Erkennung von Operatoren.

Kapitel 2

Überprüfen der Integrität eines Clusters

Ziel

Beschreiben der OpenShift-Installationsmethoden und Überprüfen der Integrität eines neu installierten Clusters

Ziele

- Beschreiben des OpenShift-Installationsvorgangs, der Full-Stack-Automatisierung und sonstiger vorhandener Infrastruktur-Installationsmethode
- Ausführen von Befehlen, die bei der Fehlerbehebung helfen, Überprüfen, ob die OpenShift-Knoten fehlerfrei sind, und Beheben häufiger Probleme bei OpenShift- und Kubernetes-Bereitstellungen.
- Identifizieren der Komponenten und Ressourcen des persistenten Storage und Bereitstellen einer Anwendung mit persistenter Volume-Anforderung

Abschnitte

- Beschreiben der Installationsmethoden (und Quiz)
- Fehlerbehebung in OpenShift-Clustern und -Anwendungen (und angeleitete Übung)
- Einführung in OpenShift Dynamic Storage (und angeleitete Übung)

Beschreiben der Installationsmethoden

Ziele

Nach Abschluss dieses Abschnitts sollten Sie den OpenShift-Installationsvorgang, die Full-Stack-Automatisierung und sonstige vorhandene Infrastruktur-Installationsmethoden beschreiben können.

Einführung in OpenShift-Installationsmethoden

Die Red Hat OpenShift Container Platform bietet zwei wichtige Installationsmethoden:

Full-Stack-Automatisierung

Mit dieser Methode stellt das OpenShift-Installationsprogramm sämtliche Rechen-, Storage- und Netzwerkressourcen eines Cloud- oder Virtualization-Anbieters bereit. Sie stellen dem Installationsprogramm Mindestdaten zur Verfügung, z. B. Anmeldedaten für einen Cloud-Anbieter und die Größe des anfänglichen Clusters. Anschließend stellt das Installationsprogramm einen voll funktionsfähigen OpenShift-Cluster bereit.

Bereits vorhandene Infrastruktur

Mit dieser Methode konfigurieren Sie eine Reihe von Computing-, Storage- und Netzwerkressourcen, und das OpenShift-Installationsprogramm konfiguriert einen initialen Cluster mit diesen Ressourcen. Sie können diese Methode verwenden, um einen OpenShift-Cluster mit Bare-Metal-Servern und Cloud- oder Virtualisierungsanbietern einzurichten, die von der Full-Stack-Automatisierungsmethode nicht unterstützt werden.

Wenn Sie eine bereits vorhandene Infrastruktur verwenden, müssen Sie die gesamte Cluster-Infrastruktur und -Ressourcen einschließlich des Bootstrap-Knotens bereitzustellen. Sie müssen das Installationsprogramm ausführen, um die erforderlichen Konfigurationsdateien zu generieren, und anschließend das Installationsprogramm erneut ausführen, um einen OpenShift-Cluster in Ihrer Infrastruktur bereitzustellen.

Zum Zeitpunkt der Veröffentlichung von Red Hat OpenShift Container Platform 4.6 unterstützt die Full-Stack-Automatisierungsmethode die Cloud-Anbieter Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure und Red Hat OpenStack Platform mit der standardmäßigen Intel-Architektur (x86). Zu den unterstützten Virtualisierungsanbietern und Architekturen für die Full-Stack-Automatisierung zählen VMware, Red Hat Virtualization, IBM Power und IBM System Z.

Jede Nebenversion des 4.x-Streams fügt weitere Funktionen und mehr Unterstützung für Anpassungen hinzu, z. B. die Wiederverwendung von vorerstellten Cloud-Ressourcen.

Vergleich der verschiedenen OpenShift-Installationsmethoden

Für bestimmte OpenShift-Features muss die Full-Stack-Automatisierungsmethode verwendet werden, z. B. die automatische Cluster-Skalierung. Es wird jedoch erwartet, dass zukünftige Releases solche Anforderungen lockern könnten.

Mit der Full-Stack-Automatisierungsmethode wird auf allen Knoten des neuen Clusters Red Hat Enterprise Linux CoreOS (RHEL CoreOS) ausgeführt. Mit der vorhandenen Infrastrukturmethode

können Sie Computing-Knoten mit Red Hat Enterprise Linux (RHEL) einrichten, aber die Kontrollebene (Master-Knoten) benötigt weiterhin RHEL CoreOS.

Beschreiben des Bereitstellungsprozesses

Die Installation erfolgt in mehreren Schritten, beginnend mit der Erstellung eines Bootstrap-Rechners, der Red Hat Enterprise Linux CoreOS mit den vom Installationsprogramm generierten Ressourcen ausführt.

Der Bootstrapping-Prozess für den Cluster lautet wie folgt:

1. Der Bootstrap-Rechner bootet und startet dann das Hosten der Remote-Ressourcen, die für das Booten der Control Plane-Rechner erforderlich sind.
2. Die Control Plane-Rechner holen die Remote-Ressourcen vom Bootstrap-Rechner ab und starten das Booten.
3. Die Control Plane-Rechner bilden einen Etcd-Cluster.
4. Der Bootstrap-Rechner startet eine temporäre Kubernetes-Control Plane mit dem neu erstellten Etcd-Cluster.
5. Die temporäre Control Plane plant die Control Plane für die Control Plane-Rechner.
6. Die temporäre Control Plane fährt herunter und liefert die Control Plane.
7. Der Bootstrap-Knoten injiziert Komponenten, die für OpenShift spezifisch sind, in die Control Plane.
8. Schließlich fährt der Installer den Bootstrap-Rechner herunter.

Das Ergebnis dieses Bootstrapping-Prozesses ist eine vollständig ausgeführte OpenShift-Control Plane, die den API-Server, die Controller (z. B. SDN) und das Etcd-Cluster umfasst. Der Cluster lädt dann die verbleibenden Komponenten, die für den täglichen Betrieb benötigt werden, über den Cluster-Versionsoperator herunter und konfiguriert diese, einschließlich der automatisierten Erstellung von Compute-Rechnern auf unterstützten Plattformen.

Anpassen einer OpenShift-Installation

Das OpenShift-Installationsprogramm ermöglicht eine sehr geringe Anpassung des initialen Clusters. Die meisten Anpassungen werden nach der Installation durchgeführt, darunter:

- Definieren von benutzerdefinierten Storage-Klassen für die dynamische Storage-Bereitstellung.
- Ändern der benutzerdefinierten Ressourcen von Cluster-Operatoren.
- Hinzufügen neuer Operatoren zu einem Cluster.
- Definieren neuer Rechnersätze.



Literaturhinweise

Weitere Informationen zu den verschiedenen Installationsmethoden finden Sie in der Dokumentation „Red Hat OpenShift Container Platform 4.6 *Installing*“ unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing/index

Weitere Informationen zur bereitgestellten Installer-Infrastruktur finden Sie im Video „Red Hat OpenShift Container Platform 4.6 OpenShift 4.x Installation – Quick Overview (IPI Installation)“ unter <https://www.youtube.com/watch?v=uBsibl4cuAl>

Weitere Informationen zur von Benutzern bereitgestellten Infrastruktur finden Sie im Video „Red Hat OpenShift Container Platform 4.6 OpenShift 4 von Benutzern bereitgestellte Infrastruktur mit VMware vSphere“ unter <https://www.youtube.com/watch?v=TsAJEEDv-gg>

► Quiz

Beschreiben der Installationsmethoden

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Für welche der folgenden Installationsmethoden muss das OpenShift-Installationsprogramm verwendet werden, um die Kontrollsicht- und Computing-Knoten zu konfigurieren?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.
- ▶ 2. Welche der folgenden Installationsmethoden ermöglicht die Einrichtung von Knoten unter Verwendung von Red Hat Enterprise Linux?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.
- ▶ 3. Welche der folgenden Installationsmethoden ermöglicht die Verwendung eines nicht unterstützten Virtualisierungsproviders auf Kosten einiger OpenShift-Funktionen?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.
- ▶ 4. Mit welcher Installationsmethode können mehrere unterstützte Cloud-Anbieter mit minimalem Aufwand verwendet werden?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.
- ▶ 5. Welche der folgenden Installationsmethoden ermöglicht eine umfangreiche Anpassung der Cluster-Einstellungen durch Eingabe in das OpenShift-Installationsprogramms?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.

► Lösung

Beschreiben der Installationsmethoden

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Für welche der folgenden Installationsmethoden muss das OpenShift-Installationsprogramm verwendet werden, um die Kontrollsicht- und Computing-Knoten zu konfigurieren?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.
- ▶ 2. Welche der folgenden Installationsmethoden ermöglicht die Einrichtung von Knoten unter Verwendung von Red Hat Enterprise Linux?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.
- ▶ 3. Welche der folgenden Installationsmethoden ermöglicht die Verwendung eines nicht unterstützten Virtualisierungsproviders auf Kosten einiger OpenShift-Funktionen?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.
- ▶ 4. Mit welcher Installationsmethode können mehrere unterstützte Cloud-Anbieter mit minimalem Aufwand verwendet werden?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.
- ▶ 5. Welche der folgenden Installationsmethoden ermöglicht eine umfangreiche Anpassung der Cluster-Einstellungen durch Eingabe in das OpenShift-Installationsprogramms?
 - a. Full-Stack-Automatisierung.
 - b. Bereits vorhandene Infrastruktur.
 - c. Sowohl Full-Stack-Automatisierung als auch bereits vorhandene Infrastruktur.
 - d. Weder Full-Stack-Automatisierung noch bereits vorhandene Infrastruktur.

Fehlerbehebung in OpenShift-Clustern und -Anwendungen

Ziele

Am Ende dieses Abschnitts sollten Sie zu Folgendem in der Lage sein: Ausführen von Befehlen, die bei der Fehlerbehebung helfen, Überprüfen, ob die OpenShift-Knoten fehlerfrei sind, und Beheben häufiger Probleme bei OpenShift- und Kubernetes-Bereitstellungen.

Beheben häufiger Probleme mit einem OpenShift-Cluster

Die meisten Fehlerbehebungen des OpenShift-Clusters ähneln der Fehlerbehebung bei Anwendungsbereitstellungen, da die meisten Komponenten von Red Hat OpenShift 4 Operatoren und Operatoren Kubernetes-Anwendungen sind. Für jeden Operator können Sie das Projekt, in dem er sich befindet, die Bereitstellung, welche die Operator-Anwendung verwaltet, und die zugehörigen Pods identifizieren. Wenn der Operator Konfigurationseinstellungen enthält, die Sie ändern müssen, können Sie die benutzerdefinierte Ressource (CR) oder manchmal die Konfigurations-Map oder Secret-Ressource, die diese Einstellungen speichert, identifizieren.

Die meisten OpenShift-Operatoren verwalten Anwendungen, die auch von standardmäßigen Kubernetes-Workload-API-Ressourcen bereitgestellt werden, z. B. DaemonSets und Bereitstellungen. Die Rolle des Operators besteht in der Regel darin, diese Ressourcen zu erstellen und sie mit der CR synchron zu halten.

Dieser Abschnitt konzentriert sich zunächst auf Cluster-Probleme, die nicht direkt mit Operatoren oder Anwendungsbereitstellungen in Zusammenhang stehen. Weiter unten in diesem Abschnitt erfahren Sie, wie Sie Fehler in Anwendungsbereitstellungen beheben.

Überprüfen des Zustands von OpenShift-Knoten

Die folgenden Befehle zeigen Informationen über den Status und den Zustand von Knoten in einem OpenShift-Cluster an:

`oc get nodes`

Zeigt eine Spalte mit dem Status der einzelnen Knoten an. Wenn ein Knoten nicht Ready ist, kann er nicht mit der OpenShift-Control Plane kommunizieren und ist tatsächlich für den Cluster inaktiv.

`oc adm top nodes`

Zeigt die aktuelle CPU- und Speicherauslastung der einzelnen Knoten an. Hierbei handelt es sich um tatsächliche Auslastungszahlen und nicht um die Ressourcenanforderungen, die der OpenShift-Scheduler als verfügbare und genutzte Kapazität des Knotens betrachtet.

`oc describe node my-node-name`

Zeigt die verfügbaren Ressourcen und andere Informationen an, die im Scheduler-Sichtbereich verwendet werden. Suchen Sie in der Ausgabe nach den Überschriften „Capacity“, „Allocateable“ und „Allocated resources“. Die Überschrift „Conditions“ gibt an, ob der Knoten unter Speicherdruck, Festplattendruck oder einer anderen Bedingung steht, die verhindert, dass der Knoten neue Container startet.

Überprüfen der Cluster-Versionsressource

Das OpenShift-Installationsprogramm erstellt ein auth-Verzeichnis mit den Dateien `kubeconfig` und `kubeadm-password`. Führen Sie den Befehl `oc login` aus, um eine Verbindung zum Cluster mit dem Benutzernamen `kubeadm` herzustellen. Das Passwort des Benutzers `kubeadm` befindet sich in der Datei `kubeadm-password`.

```
[user@host ~]$ oc login -u kubeadm -p MMTUc-TnXjo-NFyh3-aeWmC
>     https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

`ClusterVersion` ist eine benutzerdefinierte Ressource, die allgemeine Informationen zum Cluster enthält, z. B. die Aktualisierungskanäle, den Status der Cluster-Operatoren und die Cluster-Version (z. B. 4.6.29). Verwenden Sie diese Ressource, um die Version des Clusters zu deklarieren, die Sie ausführen möchten. Durch das Definieren einer neuen Version für den Cluster wird der Operator `cluster-version` angewiesen, den Cluster auf diese Version zu aktualisieren.

Mit der Cluster-Version können Sie sicherstellen, dass die gewünschte Version ausgeführt wird und dass der Cluster den richtigen Subskriptionskanal verwendet.

- Führen Sie `oc get clusterversion` aus, um die Cluster-Version abzurufen. Die Ausgabe listet die Version auf, einschließlich Minor Releases, die Cluster-Betriebszeit für eine bestimmte Version und den Gesamtstatus des Clusters.

```
[user@host ~]$ oc get clusterversion
NAME      VERSION   AVAILABLE   PROGRESSING   SINCE      STATUS
version   4.6.29    True        False         4d23h     Cluster version is 4.6.29
```

- Führen Sie `oc describe clusterversion` aus, um detailliertere Informationen zum Cluster-Status zu erhalten.

```
[user@host ~]$ oc describe clusterversion
Name:           version
Namespace:
Labels:          <none>
Annotations:    <none>
API Version:   config.openshift.io/v1
Kind:           ClusterVersion
...output omitted...
Spec:
  Channel:      stable-4.6 ①
  Cluster ID:   f33267f8-260b-40c1-9cf3-ecc406ce035e ②
  Upstream:     https://api.openshift.com/api/upgrades_info/v1/graph ③
Status:
  Available Updates: <nil> ④
  Conditions:
    Last Transition Time: 2020-08-05T18:35:08Z
    Message:             Done applying 4.6.29 ⑤
    Status:              True
    Type:                Available
  ...output omitted...
  Desired:
    Force:       false
```

```

Image: quay.io/openshift-release-dev/ocp-release@sha256:...
Version: 4.6.29
...output omitted...
History:
  Completion Time: 2021-05-24T08:12:13Z ⑥
  Image: quay.io/openshift-release-dev/ocp-release@sha256:...
  Started Time: 2021-05-24T06:03:47Z
  State: Completed ⑦
  Verified: true
  Version: 4.6.29
  Observed Generation: 2
  ...output omitted...

```

- ① Zeigt die Version des Clusters und seinen Kanal an. Je nach Ihrem Abo kann der Kanal unterschiedlich sein.
- ② Zeigt die eindeutige ID für den Cluster an. Red Hat verwendet diese ID, um Cluster und Cluster-Berechtigungen zu identifizieren.
- ③ Diese URL entspricht dem Red Hat Update Server. Der Endpunkt ermöglicht dem Cluster, seinen Aktualisierungspfad zu bestimmen, wenn er auf eine neue Version aktualisiert wird.
- ④ Dieser Eintrag listet die verfügbaren Images für die Aktualisierung des Clusters auf.
- ⑤ Dieser Eintrag listet den Verlauf auf. Die Ausgabe zeigt an, dass ein Update abgeschlossen wurde.
- ⑥ Dieser Eintrag zeigt an, wann der Cluster die im Eintrag **Version** angegebene Version bereitgestellt hat.
- ⑦ Dieser Eintrag gibt an, dass die Version erfolgreich bereitgestellt wurde. Mit diesem Eintrag können Sie die Integrität des Clusters überprüfen.

Überprüfen der Cluster-Operatoren

OpenShift Container Platform- *Cluster-Operatoren* sind Operatoren der obersten Ebene, die den Cluster verwalten. Sie sind für die Hauptkomponenten verantwortlich, z. B. für den API-Server, die Web-Konsole, den Speicher oder das SDN. Auf ihre Informationen kann über die Ressource **ClusterOperator** zugegriffen werden, über die Sie auf eine Übersicht über alle Cluster-Operatoren oder detaillierte Informationen zu einem bestimmten Operator zugreifen können.

Führen Sie `oc get clusteroperators` aus, um eine Liste aller Cluster-Operatoren abzurufen:

```

[user@host ~]$ oc get clusteroperators
NAME          VERSION  AVAILABLE  PROGRESSING  DEGRADED  SINCE
authentication 4.6.29   True       False        False      3h58m ①
cloud-credential 4.6.29  True       False        False      4d23h
cluster-autoscaler 4.6.29  True       False        False      4d23h
config-operator 4.6.29   True       False        False      4d23h
console         4.6.29   True       False        False      3h58m
csi-snapshot-controller 4.6.29  True       False        False      4d23h
dns             4.6.29   True       False        False      4d23h
etcd            4.6.29   True       False        False      4d23h
image-registry 4.6.29   True       False        False      4d23h
...output omitted...

```

Kapitel 2 | Überprüfen der Integrität eines Clusters

- 1 Jede Zeile beschreibt einen Cluster-Operator.

Das Feld NAME gibt den Namen des Operators an. Dieser Operator ist für die Verwaltung der Authentifizierung verantwortlich.

Das Feld AVAILABLE gibt an, dass der authentication-Operator erfolgreich bereitgestellt wurde und für die Verwendung im Cluster zur Verfügung steht. Beachten Sie, dass ein Cluster-Operator möglicherweise den verfügbaren Status zurückgibt, auch wenn er beeinträchtigt wird. Ein Operator meldet *degraded*, wenn sein aktueller Status über einen bestimmten Zeitraum nicht mit dem gewünschten Zustand übereinstimmt. Wenn der Operator beispielsweise drei laufende Pods benötigt, aber ein Pod abstürzt, ist der Operator verfügbar, aber in einem degradierten („*degraded*“) Zustand.

Das Feld PROGRESSING gibt an, ob ein Operator durch den Operator der obersten Ebene auf eine neuere Version aktualisiert wird. Wenn neue Ressourcen durch den Operator `cluster version` bereitgestellt werden, ist der Wert der Spalten True.

Das Feld DEGRADED gibt den Zustand des Operators zurück. Der Eintrag zeigt True an, wenn der Operator auf einen Fehler stößt, der verhindert, dass er ordnungsgemäß funktioniert. Die Operator-Services sind möglicherweise weiterhin verfügbar, allerdings werden möglicherweise nicht alle Anforderungen erfüllt. Dies kann darauf hindeuten, dass der Operator fehlschlägt und ein Benutzereingriff erforderlich ist.

Anzeigen der Protokolle von OpenShift-Knoten

Die meisten Infrastrukturkomponenten von OpenShift sind Container in Pods; Sie können Ihre Protokolle auf dieselbe Weise anzeigen, wie Sie Protokolle für eine beliebige Endbenutzeranwendung anzeigen. Einige dieser Container werden vom Kubelet erstellt und sind daher für die meisten Distributionen von Kubernetes unsichtbar, allerdings erstellen OpenShift-Cluster-Operatoren Pod-Ressourcen für sie.

Ein OpenShift-Knoten, der auf Red Hat Enterprise Linux CoreOS basiert, führt nur sehr wenige lokale Services aus, die einen direkten Zugriff auf einen Knoten benötigen, um den Status zu überprüfen. Die meisten Systemservices in Red Hat Enterprise Linux CoreOS werden als Container ausgeführt. Die wichtigsten Ausnahmen sind die CRI-O-Container-Engine und das Kubelet, welche systemd-Einheiten sind. Um diese Protokolle anzuzeigen, verwenden Sie den Befehl `oc adm node-logs`, wie in den folgenden Beispielen gezeigt:

```
[user@host ~]$ oc adm node-logs -u crio my-node-name
```

```
[user@host ~]$ oc adm node-logs -u kubelet my-node-name
```

Sie können auch alle Journal-Protokolle eines Knotens anzeigen:

```
[user@host ~]$ oc adm node-logs my-node-name
```

Öffnen einer Shell-Eingabeaufforderung auf einem OpenShift-Knoten

Administratoren, welche die Red Hat OpenShift Cluster Platform 3 und andere Distributionen von Kubernetes verwalten, öffnen häufig SSH-Sitzungen auf ihren Knoten, um den Status der Control Plane und der Container-Engine zu überprüfen oder um Änderungen an Konfigurationsdateien vorzunehmen. Auch wenn dies noch möglich ist, wird diese Vorgehensweise nicht mehr für die Red Hat OpenShift Cluster Platform 4 empfohlen.

Wenn Sie Ihren Cluster mit der Full-Stack-Automatisierungsmethode installieren, können Sie nicht direkt über das Internet auf die Cluster-Knoten zugreifen, da sie sich in einem virtuellen privaten Netzwerk befinden, das von AWS als virtuelle private Cloud (VPC) bezeichnet wird. Um SSH-Sitzungen zu öffnen, ist ein Bastion-Server auf derselben VPC Ihres Clusters erforderlich, der auch eine öffentliche IP-Adresse zugewiesen ist. Die Erstellung eines Bastion-Servers hängt von Ihrem Cloud-Anbieter ab und wird in diesem Kurs nicht behandelt.

Der Befehl `oc debug node` bietet eine Möglichkeit, in jedem Knoten Ihres Clusters einen Shell-Prompt zu öffnen. Diese Eingabeaufforderung wird von einem speziellen Tools-Container bereitgestellt, der das Root-Dateisystem des Knotens im Ordner `/host` mountet und Ihnen ermöglicht, alle Dateien vom Knoten zu überprüfen.

Um lokale Befehle direkt über den Knoten auszuführen, müssen Sie in einer `oc debug node`-Sitzung im Ordner `/host` eine chroot-Shell starten. Anschließend können Sie die lokalen Dateisysteme des Knotens und den Status seiner systemd-Services überprüfen und weitere Aufgaben durchführen, für die andernfalls eine SSH-Sitzung erforderlich wäre. Im Folgenden sehen Sie ein Beispiel für eine `oc debug node`-Sitzung:

```
[user@host ~]$ oc debug node/my-node-name
...output omitted...
sh-4.4# chroot /host
sh-4.4# systemctl is-active kubelet
active
```

Eine Shell-Sitzung, die über den Befehl `oc debug node` gestartet wurde, hängt davon ab, dass die OpenShift-Control Plane funktioniert. Sie verwendet dieselbe Tunneling-Technologie, die das Öffnen einer Shell-Eingabeaufforderung in einem ausgeführten Pod ermöglicht (siehe den Befehl `oc rsh` weiter unten in diesem Abschnitt). Der Befehl `oc debug node` basiert nicht auf den SSH- oder RSH-Protokollen.

Wenn Ihre Kontrollsicht nicht funktioniert, ist entweder Ihr Knoten nicht bereit oder aus irgendeinem Grund nicht in der Lage, mit der Control Plane zu kommunizieren; daher können Sie sich nicht auf den Befehl `oc debug node` verlassen und benötigen einen Bastion-Host.



Warnung

Seien Sie bei der Verwendung des Befehls `oc debug node` sehr umsichtig. Einige Aktionen können den Knoten unbrauchbar machen, z. B. das Anhalten des Kubelets, und eine Wiederherstellung ausschließlich mit `oc`-Befehlen ist nicht möglich.

Fehlerbehebung von Container-Engines

Verwenden Sie in einer `oc debug node`-Sitzung den Befehl `crlctl`, um allgemeine Informationen zu allen lokalen Containern, die auf dem Knoten ausgeführt werden, abzurufen. Für diese Aufgabe können Sie nicht den Befehl `podman` verwenden, da er keine Sichtbarkeit für Container besitzt, die von CRI-O erstellt wurden. Das folgende Beispiel listet alle Container auf, die auf einem Knoten ausgeführt werden. Der Befehl `oc describe node` bietet dieselben Informationen, wird jedoch nach Pod statt nach Container organisiert.

```
[user@host ~]$ oc debug node/my-node-name
...output omitted...
sh-4.4# chroot /host
sh-4.4# crictl ps
...output omitted...
```

Fehlerbehebung von Anwendungsumsetzungen

Bei der Fehlerbehebung von Anwendungen können Sie die Unterschiede zwischen Kubernetes-Bereitstellungen und OpenShift-Bereitstellungskonfigurationen in der Regel ignorieren. Die gängigen Fehlerszenarien und die Methoden zur Fehlerbehebung sind im Wesentlichen identisch.

Es gibt viele Szenarien, die in späteren Kapiteln dieses Kurses beschrieben werden, beispielsweise Pods, die nicht geplant werden können. Dieser Abschnitt konzentriert sich auf gängige Szenarien, die für generische Anwendungen gelten, und dieselben Szenarien gelten in der Regel auch für Operatoren.

Fehlerbehebung von nicht startbaren Pods

Ein gängiges Szenario ist, dass OpenShift einen Pod erstellt und der Pod niemals den Status `Running` erreicht. Dies bedeutet, dass OpenShift die Container in diesem Pod nicht starten konnte. Starten Sie die Fehlerbehebung mit den Befehlen `oc get pod` und `oc status`, um zu überprüfen, ob Ihre Pods und Container ausgeführt werden. Zu einem bestimmten Zeitpunkt befinden sich die Pods in einem Fehlerstatus, z. B. `ErrImagePull` oder `ImagePullBackOff`.

Wenn dies geschieht, wird im ersten Schritt mit dem Befehl `oc get events` eine Liste der Ereignisse aus dem aktuellen Projekt angezeigt. Wenn Ihr Projekt viele Pods enthält, erhalten Sie mit dem Befehl `oc describe pod` eine nach Pod gefilterte Liste von Ereignissen. Sie können auch ähnliche `oc describe`-Befehle ausführen, um Ereignisse nach Bereitstellungen und Bereitstellungskonfigurationen zu filtern.

Fehlerbehebung beim Ausführen und Beenden von Pods

Ein weiteres gängiges Szenario ist, dass OpenShift einen Pod erstellt und für kurze Zeit kein Problem auftritt. Der Pod wechselt in den Status `Running`, d. h. es wird mindestens einer seiner Container gestartet. Später funktioniert eine Anwendung, die in einem der Pod-Container ausgeführt wird, nicht mehr. Sie wird entweder beendet oder gibt auf Benutzeranfragen Fehlermeldungen zurück.

Wenn die Anwendung von einer ordnungsgemäß konzipierten Bereitstellung verwaltet wird, sollte sie auch Integritätstests beinhalten, welche die Anwendung schließlich beenden und den Container stoppen. Wenn dies der Fall ist, versucht OpenShift mehrmals, den Container neu zu starten. Wenn die Anwendung aufgrund von Integritätstests oder aus anderen Gründen weiterhin beendet wird, bleibt der Pod im Status `CrashLoopBackOff`.

Ein Container, der auch nur für kurze Zeit ausgeführt wird, generiert Protokolle. Diese Protokolle werden nicht verworfen, wenn der Container beendet wird. Der Befehl `oc logs` zeigt die Protokolle von jedem Container in einem Pod an. Wenn der Pod einen einzelnen Container enthält, erfordert der Befehl `oc logs` nur den Namen des Pods.

```
[user@host ~]$ oc logs my-pod-name
```

Wenn der Pod mehrere Container enthält, erfordert der Befehl `oc logs` die Option `-c`.

```
[user@host ~]$ oc logs my-pod-name -c my-container-name
```

Die Interpretation von Anwendungsprotokollen erfordert spezifische Kenntnisse der jeweiligen Anwendung. Wenn alles gut geht, liefert die Anwendung eindeutige Fehlermeldungen, die Ihnen dabei helfen, das Problem zu finden.

Einführung in die aggregierte Protokollierung von OpenShift

Red Hat OpenShift Container Platform 4 bietet das auf Elasticsearch, Fluent oder Rsyslog und Kibana basierende Cluster-Protokollierungssubsystem, das Protokolle aus dem Cluster und seinen Containern aggregiert.

Die Bereitstellung und Konfiguration des OpenShift-Cluster-Protokollierungssystems über seinen Operator sprengt den Rahmen dieses Kurses. Weitere Informationen finden Sie in den Referenzen am Ende dieses Abschnitts.

Erstellen von Pods zur Fehlerbehebung

Wenn Sie sich nicht sicher sind, ob sich Ihre Probleme auf das Anwendungscontainer-Image oder die Einstellungen beziehen, die es von seinen OpenShift-Ressourcen erhält, dann ist der Befehl `oc debug` sehr hilfreich. Mit diesem Befehl wird ein Pod basierend auf einem vorhandenen Pod, einer Bereitstellungskonfiguration, einer Bereitstellung oder einer anderen Ressource aus der Workloads-API erstellt.

Der neue Pod führt anstelle des Standard-Einstiegpunkts des Container-Images eine interaktive Shell aus. Er wird auch mit deaktivierten Integritätstests ausgeführt. Auf diese Weise können Sie die Umgebungsvariablen, den Netzwerkzugriff auf andere Services und die Berechtigungen im Pod ganz einfach überprüfen.

Mit den Befehlszeilenoptionen des Befehls `oc debug` können Sie Einstellungen festlegen, die nicht geklont werden sollen. Sie können beispielsweise das Container-Image ändern oder eine feste Benutzer-ID festlegen. Für einige Einstellungen sind möglicherweise Cluster-Administratorberechtigungen erforderlich.

Ein gängiges Szenario besteht darin, einen Pod aus einer Bereitstellung zu erstellen, aber als root-Benutzer auszuführen und so zu beweisen, dass die Bereitstellung auf ein Container-Image verweist, das nicht für die Ausführung unter den standardmäßigen Sicherheitsrichtlinien von OpenShift konzipiert wurde:

```
[user@host ~]$ oc debug deployment/my-deployment-name --as-root
```

Ändern eines ausgeführten Containers

Da Container-Images unveränderlich sind und Container kurzlebig sein sollten, wird nicht empfohlen, die Änderungen an den ausgeführten Containern vorzunehmen. Es kann jedoch vorkommen, dass diese Änderungen bei der Behebung von Anwendungsproblemen hilfreich sind. Wenn Sie versuchen, einen ausgeführten Container zu ändern, vergessen Sie nicht, dieselben Änderungen wieder auf das Container-Image und die zugehörigen Anwendungsressourcen anzuwenden, und überprüfen Sie anschließend, ob die nun permanenten Fixes erwartungsgemäß funktionieren.

Mit den folgenden Befehlen können Sie Änderungen an den ausgeführten Containern vornehmen. Diese gehen alle davon aus, dass Pods einen einzelnen Container enthalten. Ist dies nicht der Fall, müssen Sie die Option `-c my-container-name` hinzufügen.

Kapitel 2 | Überprüfen der Integrität eines Clusters

- oc **rsh** *my-pod-name*
Öffnet eine Shell in einem Pod, um die Shell-Befehle interaktiv und nicht interaktiv auszuführen.
- oc **cp** */local/path my-pod-name:/container/path*
Kopiert lokale Dateien an einen Speicherort in einem Pod. Sie können auch Argumente umkehren und Dateien aus einem Pod in Ihr lokales Dateisystem kopieren. Siehe auch den Befehl oc **rsync**, um mehrere Dateien gleichzeitig zu kopieren.
- oc **port-forward** *my-pod-name local-port:remote-port*
Erstellt einen TCP-Tunnel vom *local-port* auf Ihrer Workstation zum *local-port* auf dem Pod. Der Tunnel ist so lange aktiv, wie Sie oc **port-forward** ausführen. Auf diese Weise können Sie den Netzwerkzugriff auf den Pod erhalten, ohne ihn über eine Route bereitzustellen. Da der Tunnel auf Ihrem localhost gestartet wird, kann er nicht von anderen Rechnern verwendet werden.

Befehle für die Fehlerbehebung auf der OpenShift CLI

Manchmal ist nicht klar, warum ein oc-Befehl fehlschlägt, und Sie müssen Fehler in den allgemeinen Aktionen beheben, um die Ursache zu finden. Vielleicht müssen Sie wissen, was ein bestimmter Aufruf des Befehls oc im Hintergrund tut, sodass Sie das Verhalten mit einem Automatisierungstool replizieren können, das OpenShift- und Kubernetes-API-Anforderungen wie Ansible-Manuskripte mithilfe des Moduls k8s erstellt.

Die Option **--loglevel level** zeigt OpenShift-API-Anforderungen ab Stufe 6 an. Wenn Sie die Ebene auf bis zu 10 erhöhen, werden weitere Informationen zu diesen Anforderungen hinzugefügt, beispielsweise die HTTP-Anforderungsheader und die Antwortbodys. Ebene 10 enthält außerdem einen curl-Befehl zum Replizieren jeder einzelnen Anforderung.

Sie können diese beiden Befehle von jedem Projekt aus ausprobieren und ihre Ergebnisse vergleichen.

```
[user@host ~]$ oc get pod --loglevel 6
```

```
[user@host ~]$ oc get pod --loglevel 10
```

Manchmal benötigen Sie nur das Authentifizierungstoken, das der Befehl oc verwendet, um OpenShift-API-Anforderungen zu authentifizieren. Mit diesem Token kann ein Automatisierungstool OpenShift-API-Anforderungen so erstellen, als wäre es als Ihr Benutzer angemeldet. Verwenden Sie zum Abrufen des Tokens die Option **-t** des Befehls oc **whoami**:

```
[user@host ~]$ oc whoami -t
```



Literaturhinweise

Weitere Informationen zu OpenShift-Events finden Sie im Abschnitt *Viewing system event information in an OpenShift Container Platform cluster* des Kapitels *Working with clusters* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 Nodes unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/nodes/index#nodes-containers-events

Weitere Informationen zum Kopieren von Dateien in ausgeführte Container finden Sie im Abschnitt *Copying files to or from an OpenShift Container Platform container* des Kapitels *Working with containers* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 Nodes unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/nodes/index#nodes-containers-copying-files

Weitere Informationen zum Ausführen von Befehlen für ausgeführte Container finden Sie im Abschnitt *Executing remote commands in an OpenShift Container Platform container* des Kapitels *Working with containers* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 Nodes unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/nodes/index#nodes-containers-remote-commands

Weitere Informationen über das Weiterleiten von lokalen Ports an ausgeführte Container finden Sie im Abschnitt *Using port forwarding to access applications in a container* des Kapitels *Working with containers* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 Nodes unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/nodes/index#nodes-containers-port-forwarding

Weitere Informationen zur aggregierten Protokollierung finden Sie in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 Logging unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/logging/index

Benutzerdefinierte ClusterOperator-Ressource

<https://github.com/openshift/enhancements/blob/master/dev-guide/cluster-version-operator/dev/clusteroperator.md>

► Angeleitete Übung

Fehlerbehebung in OpenShift-Clustern und -Anwendungen

In dieser Übung führen Sie Befehle aus, die bei der Behebung häufiger Probleme mit der OpenShift-Control Plane und bei Anwendungsbereitstellungen behilflich sind.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Überprüfen des allgemeinen Status eines OpenShift-Clusters.
- Überprüfen der lokalen Services und Pods, die auf einem OpenShift-Computing-Knoten ausgeführt werden.
- Diagnostizieren und Beheben von Problemen bei der Bereitstellung einer Anwendung.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt die Ressourcendateien für die Übung. Außerdem wird das Projekt `install-troubleshoot` mit einer Anwendung erstellt, die Sie während dieser Übung diagnostizieren und von Fehlern befreien.

```
[student@workstation ~]$ lab install-troubleshoot start
```

Anweisungen

- 1. Melden Sie sich beim OpenShift-Cluster an, und überprüfen Sie den Status Ihrer Cluster-Knoten.
- 1.1. Rufen Sie die Kursumgebungs-Konfigurationsdatei auf, auf die Sie unter `/usr/local/etc/ocp4.config` zugreifen können.

```
[student@workstation ~]$ source /usr/local/etc/ocp4.config
```

- 1.2. Melden Sie sich als Benutzer `kubeadmin` bei dem Cluster an. Akzeptieren Sie das unsichere Zertifikat, wenn Sie dazu aufgefordert werden.

Kapitel 2 | Überprüfen der Integrität eines Clusters

```
[student@workstation ~]$ oc login -u kubeadmin -p ${RHT_OCP4_KUBEADM_PASSWD} \
>   https://api.ocp4.example.com:6443
The server uses a certificate signed by an unknown authority.
You can bypass the certificate check, but any data you send to the server could be
intercepted by others.
Use insecure connections? (y/n): y

Login successful.
...output omitted...
```

- 1.3. Überprüfen Sie, ob alle Knoten auf Ihrem Cluster bereit sind:

```
[student@workstation ~]$ oc get nodes
NAME      STATUS    ROLES     AGE      VERSION
master01   Ready     master,worker   2d      v1.19.3+012b3ec
master02   Ready     master,worker   2d      v1.19.3+012b3ec
master03   Ready     master,worker   2d      v1.19.3+012b3ec
```

- 1.4. Überprüfen Sie, ob sich einer Ihrer Knoten an der Belastungsgrenze von CPU und verfügbarem Speicher befindet.

Wiederholen Sie den folgenden Befehl einige Male, um sicherzustellen, dass Ihnen die tatsächliche Nutzung von CPU und Speicher durch Ihre Knoten angezeigt wird. Die Zahlen, die Sie sehen, sollten bei jedem wiederholten Befehl leicht abweichen.

```
[student@workstation ~]$ oc adm top node
NAME      CPU(cores)   CPU%     MEMORY(bytes)   MEMORY%
master01   499m        14%     3235Mi          21%
master02   769m        21%     4933Mi          33%
master03   1016m       29%     6087Mi          40%
```

- 1.5. Verwenden Sie den Befehl `oc describe`, um sicherzustellen, dass keine Bedingungen vorliegen, die auf Probleme hindeuten könnten.

```
[student@workstation ~]$ oc describe node master01
...output omitted...
Conditions:
  Type      Status  ...  Message
  ----      -----  ... 
  MemoryPressure  False  ...  kubelet has sufficient memory available
  DiskPressure   False  ...  kubelet has no disk pressure
  PIDPressure    False  ...  kubelet has sufficient PID available
  Ready         True   ...  kubelet is posting ready status
Addresses:
  ...output omitted...
```

- 2. Überprüfen Sie die Protokolle des internen Registry-Operators, des internen Registry-Servers und des Kubelets eines Knotens.
- 2.1. Listen Sie alle Pods im Projekt `openshift-image-registry` auf, und identifizieren Sie dann den Pod, der den Operator ausführt, und den Pod, der den internen Registry-Server ausführt.

Kapitel 2 | Überprüfen der Integrität eines Clusters

```
[student@workstation ~]$ oc get pod -n openshift-image-registry
NAME                                READY   STATUS    ...
cluster-image-registry-operator-564bd5dd8f-s46bz   1/1     Running   ...
image-registry-794dfc7978-w7w69                  1/1     Running   ...
...output omitted...
```

- 2.2. Folgen Sie den Protokollen des Operator-Pods (`cluster-image-registry-operator-xxx`). Ihre Ausgabe unterscheidet sich möglicherweise vom folgenden Beispiel.

```
[student@workstation ~]$ oc logs --tail 3 -n openshift-image-registry \
>   cluster-image-registry-operator-564bd5dd8f-s46bz
I0614 15:31:29.316773      1 imageregistrycertificates.go:97]
  ImageRegistryCertificatesController: event from workqueue successfully processed
I0614 15:31:29.317055      1 controllerimagepruner.go:323] event from image
  pruner workqueue successfully processed
I0614 15:31:29.341756      1 controller.go:333] event from workqueue successfully
  processed
```

- 2.3. Folgen Sie den Protokollen des Image-Registry-Server-Pods (`image-registry-xxx` aus der Ausgabe des Befehls `oc pod`, der zuvor ausgeführt wurde). Ihre Ausgabe unterscheidet sich möglicherweise vom folgenden Beispiel.

```
[student@workstation ~]$ oc logs --tail 1 -n openshift-image-registry \
>   image-registry-794dfc7978-w7w69
time="2021-06-10T16:11:55.871435967Z" level=info msg=response
go.version=g01.11.6 http.request.host="10.129.2.44:5000"
http.request.id=f4d83df5-8ed7-4651-81d4-4ed9f758c67d http.request.method=GET
http.request.remoteaddr="10.129.2.50:59500" http.request.uri=/extensions/v2/
metrics http.request.useragent=Prometheus/2.11.0 http.response.contenttype="text/
plain; version=0.0.4" http.response.duration=12.141585ms http.response.status=200
http.response.written=2326
```

- 2.4. Folgen Sie den Protokollen des Kubelets aus demselben Knoten, den Sie im vorherigen Schritt auf CPU und Speicherauslastung überprüft haben. Ihre Ausgabe unterscheidet sich möglicherweise vom folgenden Beispiel.

```
[student@workstation ~]$ oc adm node-logs --tail 1 -u kubelet master01
-- Logs begin at Tue 2021-05-25 16:53:09 UTC, end at Thu 2021-06-10 16:14:58 UTC.
--
Jun 09 21:26:11.244996 master01 systemd[1]: kubelet.service: Consumed 6min 24.649s
CPU time
-- Logs begin at Tue 2021-05-25 16:53:09 UTC, end at Thu 2021-06-10 16:14:58 UTC.
--
Jun 10 16:14:58.104396 master01 hyperkube[1892]: I0610 16:14:58.104356      1892
  prober.go:126] Readiness probe for "console-operator-6d89b76984-wd5t8_openshift-
  console-operator(6e9ddc9d-aacd-462d-81c3-cfe154e8287f):console-operator" succeeded
```

- 3. Starten Sie eine Shell-Sitzung mit demselben Knoten, den Sie zuvor verwendet haben, um die OpenShift-Services und -Pods zu überprüfen. Nehmen Sie keine

Kapitel 2 | Überprüfen der Integrität eines Clusters

Änderungen am Knoten vor, z. B. zum Beenden von Services oder zum Bearbeiten von Konfigurationsdateien.

- 3.1. Starten Sie eine Shell-Sitzung auf dem Knoten, und verwenden Sie anschließend den Befehl `chroot`, um auf das lokale Dateisystem des Hosts zuzugreifen.

```
[student@workstation ~]$ oc debug node/master01
Creating debug namespace/openshift-debug-node-5zsch ...
Starting pod/master01-debug ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.50.10
If you do not see a command prompt, try pressing enter.
sh-4.4# chroot /host
sh-4.4#
```

- 3.2. Überprüfen Sie, ob Kubelet und CRI-O-Container-Engine weiterhin mit derselben Shell-Sitzung ausgeführt werden. Geben Sie `q` ein, um den Befehl zu beenden.

```
sh-4.4# systemctl status kubelet
● kubelet.service - Kubernetes Kubelet
  Loaded: loaded (/etc/systemd/system/kubelet.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/kubelet.service.d
            └─10-mco-default-env.conf, 20-nodenet.conf
  Active: active (running) since Thu 2021-06-10 15:22:22 UTC; 1h 2min ago
...output omitted...
q
```

Führen Sie denselben Befehl für den cri-o-Service erneut aus. Geben Sie `q` ein, um den Befehl zu beenden.

```
sh-4.4# systemctl status cri-o
● crio.service - Open Container Initiative Daemon
  Loaded: loaded (/usr/lib/systemd/system/crio.service; disabled; vendor preset: disabled)
  Drop-In: /etc/systemd/system/crio.service.d
            └─10-mco-default-env.conf, 20-nodenet.conf
  Active: active (running) since Thu 2021-06-10 15:21:56 UTC; 1h 5min ago
...output omitted...
q
```

- 3.3. Überprüfen Sie, ob der openvswitch-Pod ausgeführt wird; verwenden Sie dabei immer noch dieselbe Shell-Sitzung.

```
sh-4.4# crictl ps --name openvswitch
CONTAINER ID      ...      STATE      NAME          ATTEMPT      POD ID
13f0b0ed3497a    ...      Running    openvswitch   0           4bc278dddf007
```

- 3.4. Beenden Sie die `chroot`-Sitzung und die Shell-Sitzung mit dem Knoten. Dadurch wird auch der Befehl `oc debug node` beendet.

Kapitel 2 | Überprüfen der Integrität eines Clusters

```
sh-4.4# exit
exit
sh-4.4# exit
exit

Removing debug pod ...
[student@workstation ~]$
```

- 4. Rufen Sie das Projekt `install-troubleshoot` auf, um einen Pod zu diagnostizieren, der sich in einem Fehlerstatus befindet.

4.1. Verwenden Sie das Projekt `install-troubleshoot`.

```
[student@workstation ~]$ oc project install-troubleshoot
Now using project "install-troubleshoot" on server
"https://api.ocp4.example.com:6443"+.
```

4.2. Überprüfen Sie, ob das Projekt über einen einzelnen Pod im Status `ErrImagePull` oder `ImagePullBackOff` verfügt.

```
[student@workstation ~]$ oc get pod
NAME          READY   STATUS        ...
pgsql-7d4cc9d6d-m5r59  0/1     ImagePullBackOff  ...
```

4.3. Überprüfen Sie, ob das Projekt eine Kubernetes-Bereitstellung enthält, die den Pod verwaltet.

```
[student@workstation ~]$ oc status
...output omitted...
deployment/pgsql deploys registry.redhat.io/rhel8/postgresq-13:1
  deployment #1 running for 8 minutes - 0/1 pods
...output omitted...
```

4.4. Listen Sie alle Ereignisse aus dem aktuellen Projekt auf, und suchen Sie nach Fehlermeldungen im Zusammenhang mit dem Pod.

```
[student@workstation ~]$ oc get events
LAST SEEN    TYPE      REASON           OBJECT                MESSAGE
112s         Normal    Scheduled        pod/pgsql-7d4cc9d6d-m5r59  Successfully
assigned install-troubleshoot/pgsql-7d4cc9d6d-m5r59 to master03
112s         Normal    AddedInterface   pod/pgsql-578f78ccb-nbm8q  Add eth0
[10.9.0.87/23]
21s          Normal    Pulling          pod/pgsql-7d4cc9d6d-m5r59  Pulling
image "registry.redhat.io/rhel8/postgresq-13:1"
21s          Warning   Failed           pod/pgsql-7d4cc9d6d-m5r59  Failed
to pull image "registry.redhat.io/rhel8/postgresq-13:1": rpc error: code =
Unknown desc = Error reading manifest 1 in registry.redhat.io/rhel8/postgresq-13:
unknown: Not Found
21s          Warning   Failed           pod/pgsql-7d4cc9d6d-m5r59  Error:
ErrImagePull
```

Kapitel 2 | Überprüfen der Integrität eines Clusters

```

8s      Normal   BackOff        pod/sql-7d4cc9d6d-m5r59  Back-off
  pulling image "registry.redhat.io/rhel8/postgresq-13:1"
8s      Warning  Failed        pod/sql-7d4cc9d6d-m5r59  Error:
    ImagePullBackOff
112s     Normal   SuccessfulCreate replicaset/sql-7d4cc9d6d  Created pod:
    sql-7d4cc9d6d-m5r59
112s     Normal   ScalingReplicaSet deployment/sql           Scaled up
    replica set sql-7d4cc9d6d to 1

```

Diese Ausgabe zeigt auch ein Problem beim Abrufen des Images für die Bereitstellung des Pods an.

- 4.5. Melden Sie sich mit Ihrem Red Hat-Benutzerkonto beim Red Hat Container Catalog an.

```
[student@workstation ~]$ podman login registry.redhat.io
Username: your_username
Password: your_password
Login Succeeded!
```

- 4.6. Verwenden Sie Skopeo, um Informationen über das Container-Image in den Ereignissen zu finden.

```
[student@workstation ~]$ skopeo inspect \
> docker://registry.redhat.io/rhel8/postgresq-13:1
FATA[0000] Error parsing image name "docker://registry.redhat.io/rhel8/
postgresq-13:1": Error reading manifest 1 in registry.redhat.io/rhel8/
postgresq-13: unknown: Not Found
```

- 4.7. Es sieht so aus, als ob das Container-Image falsch geschrieben wurde. Überprüfen Sie, ob es funktioniert, wenn Sie `postgresq-13` durch `postgresql-13` ersetzen.

```
[student@workstation ~]$ skopeo inspect \
> docker://registry.redhat.io/rhel8/postgresql-13:1
{
  "Name": "registry.redhat.io/rhel8/postgresql-13",
  ...output omitted...
```

- 4.8. Um zu überprüfen, ob der Image-Name die eigentliche Ursache des Fehlers ist, bearbeiten Sie die `sql`-Bereitstellung, um den Namen des Container-Images zu korrigieren. Die Befehle `oc edit` verwenden `vi` als Standard-Editor.

**Warnung**

In einem realen Szenario würden Sie sich fragen, wer die PostgreSQL-Datenbank bereitgestellt hat, um die YAML zu korrigieren und die Anwendung erneut bereitzustellen.

```
[student@workstation ~]$ oc edit deployment/sql
...output omitted...
spec:
  containers:
```

Kapitel 2 | Überprüfen der Integrität eines Clusters

```
- env:  
  - name: POSTGRESQL_DATABASE  
    value: db  
  - name: POSTGRESQL_PASSWORD  
    value: pass  
  - name: POSTGRESQL_USER  
    value: user  
image: registry.redhat.io/rhel8/postgresql-13:1-7  
...output omitted...
```

- 4.9. Stellen Sie sicher, dass eine neue Bereitstellung aktiv ist.

```
[student@workstation ~]$ oc status  
...output omitted...  
deployment #2 running for 10 seconds - 0/1 pods  
deployment #1 deployed 5 minutes ago
```

- 4.10. Listen Sie alle Pods im aktuellen Projekt auf. Es kann sein, dass für einige Augenblicke sowohl der alte Pod als auch der neue Pod angezeigt werden. Wiederholen Sie den folgenden Befehl, bis Sie sehen, dass der neue Pod bereit ist und ausgeführt wird und der alte Pod nicht mehr angezeigt wird.

```
[student@workstation ~]$ oc get pods  
NAME          READY   STATUS    RESTARTS   AGE  
pgsql-544c9c666f-btlw8  1/1     Running   0          55s
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab install-troubleshoot finish
```

Hiermit ist die angeleitete Übung beendet.

Einführung in OpenShift Dynamic Storage

Ziele

Nach Abschluss dieses Abschnitts sollten Sie die Komponenten und Ressourcen des persistenten Storage identifizieren und eine Anwendung mit persistenter Volume-Anforderung bereitstellen können.

Persistenter Storage: Übersicht

Container verwenden standardmäßig einen temporären Storage. Wenn beispielsweise ein Container gelöscht wird, werden alle darin enthaltenen Dateien und Daten ebenfalls gelöscht. Um die Dateien beizubehalten, bieten Container zwei Hauptmethoden zum Verwalten von persistentem Storage: Volumes und Bind-Mounts. Volumes sind die bevorzugte OpenShift-Methode zum Verwalten von persistentem Storage. Volumes werden manuell vom Administrator oder dynamisch über eine Storage-Klasse verwaltet. Entwickler, die mit Containern auf einem lokalen System arbeiten, können mit einer Bind-Bereitstellung ein lokales Verzeichnis in einen Container mounten.

OpenShift-Clusteradministratoren verwenden das persistente Volume-Framework von Kubernetes, um den persistenten Storage für die Benutzer eines Clusters zu verwalten. Storage für den Cluster kann entweder statisch oder dynamisch bereitgestellt werden. Für die statische Bereitstellung muss ein Cluster-Administrator persistente Volumes manuell erstellen. Bei der dynamischen Bereitstellung werden Storage-Klassen verwendet, um die persistenten Volumes bei Bedarf zu erstellen.

Die OpenShift Container Platform verwendet Storage-Klassen, mit denen Administratoren persistente Storage bereitstellen können. Storage-Klassen beschreiben Storage-Typen für den Cluster werden verwendet, um bei Bedarf dynamischen Storage bereitzustellen.

Entwickler verwenden persistente Volume-Ansprüche, um persistente Volumes dynamisch zu Ihren Anwendungen hinzuzufügen. Dazu müssen die Entwickler die Details der Storage-Infrastruktur nicht kennen. Bei der statischen Bereitstellung verwenden Entwickler vorab erstellte PVs oder bitten einen Cluster-Administrator, persistente Volumes für Ihre Anwendungen manuell zu erstellen.

Jede Anforderung für ein persistentes Volume (Persistent Volume Claim, PVC) gehört zu einem bestimmten Projekt. Um eine PVC zu erstellen, müssen Sie neben anderen Optionen den Zugriffsmodus und die Größe festlegen. Nach der Erstellung können PVCs nicht zwischen Projekten geteilt werden. Entwickler verwenden ein PVC für den Zugriff auf ein persistentes Volume (PV). Persistente Volumes sind nicht exklusiv für Projekte und sind für den gesamten OpenShift-Cluster erreichbar. Wenn ein persistentes Volume an eine persistente Volume-Anforderung gebunden wird, kann das persistente Volume nicht mehr an eine andere persistente Volume-Anforderung gebunden werden.

Lebenszyklus von persistenten Volumes (PV) und Anforderungen für persistente Volumes (PVC)

Anforderungen für persistente Volumes werden verwendet, um persistente Volume-Ressourcen anzufordern. Dafür gilt die Voraussetzung, dass das PV nicht an eine andere PVC gebunden sein darf. Darüber hinaus muss das PV den in der PVC angegebenen Zugriffsmodus anbieten

und mindestens die in der PVC angeforderte Größe haben. PVCs können zusätzliche Kriterien definieren, z. B. den Namen einer Storage-Klasse. Wenn eine PVC kein PV findet, das alle Kriterien erfüllt, wird die PVC als ausstehend festgelegt und wartet, bis ein entsprechendes PV verfügbar wird. Ein Cluster-Administrator kann das PV manuell erstellen, oder eine Storage-Klasse kann das PV dynamisch erstellen. Ein gebundenes persistentes Volume kann als Volume an einen bestimmten Mount-Punkt im Pod gemountet werden (z. B. /var/lib/pgsql für eine PostgreSQL-Datenbank).

Überprüfen des dynamischen bereitgestellten Storage

Verwenden Sie den Befehl `oc get storageclass`, um die verfügbaren Storage-Klassen anzuzeigen. Die Ausgabe identifiziert die standardmäßige Storage-Klasse. Wenn eine Storage-Klasse vorhanden ist, werden persistente Volumes dynamisch erstellt, um die Anforderungen für persistente Volumes zu erfüllen. Für persistente Volume-Anforderungen ohne Angabe einer Storage-Klasse wird die standardmäßige Storage-Klasse verwendet.

```
[user@host ~]$ oc get storageclass
NAME          PROVISIONER
nfs-storage (default)  nfs-storage-provisioner ...
```



Anmerkung

Die Kursumgebung verwendet einen externen Open Source NFS-Provisioner. Der Provisioner erstellt dynamische persistente NFS-Volumes auf einem vorhandenen NFS-Server. Red Hat rät davon ab, diesen Provisioner in Produktionsumgebungen einzusetzen.

Bereitstellen von dynamisch bereitgestelltem Storage

Um ein Volume zu einer Anwendung hinzuzufügen, erstellen Sie eine `PersistentVolumeClaim`-Ressource und fügen Sie sie als Volume zur Anwendung hinzu. Erstellen Sie die Anforderung für das persistente Volume entweder mit einem Kubernetes-Manifest oder mit dem Befehl `oc set volumes`. Zusätzlich zum Erstellen einer neuen Anforderung für ein persistentes Volume oder zur Verwendung einer vorhandenen Anforderung für ein persistentes Volume können Sie mit dem Befehl `oc set volumes` eine Bereitstellung ändern, um die Anforderung für das persistente Volume als ein Volume innerhalb des Pods zu mounten.

Verwenden Sie den Befehl `oc set volumes`, um ein Volume zu einer Anwendung hinzuzufügen:

```
[user@host ~]$ oc set volumes deployment/example-application \
>   --add --name example-storage --type pvc --claim-class nfs-storage \
>   --claim-mode rwo --claim-size 15Gi --mount-path /var/lib/example-app \
>   --claim-name example-storage
```

Dieser Befehl erstellt eine PVC-Ressource und fügt sie als Volume innerhalb des Pods zur Anwendung hinzu.

Im folgenden YAML-Beispiel wird eine Anforderung für ein persistentes Volume angegeben.

So erstellen Sie ein `PersistentVolumeClaim`-API-Objekt:

```

apiVersion: v1
kind: PersistentVolumeClaim 1
metadata:
  name: example-pv-claim 2
  labels:
    app: example-application
spec:
  accessModes:
    - ReadWriteOnce 3
  resources:
    requests:
      storage: 15Gi 4

```

- 1** Gibt an, dass es sich um eine Anforderung für ein persistentes Volume handelt.
- 2** Der Name, der im Feld `claimName` des `persistentVolumeClaim`-Elements im Abschnitt `Volumes` eines Bereitstellungsmanifests verwendet werden muss.
- 3** Der Provisioner der Storage-Klasse muss diesen Zugriffsmodus anbieten. Wenn persistente Volumes statisch erstellt werden, muss ein geeignetes persistentes Volume existieren, das diesen Zugriffsmodus bereitstellt.
- 4** Die Storage-Klasse erstellt ein persistentes Volume für diese Größenanforderung. Wenn persistente Volumes statisch erstellt werden, muss ein geeignetes persistentes Volume existieren, das mindestens die angeforderte Größe hat.

OpenShift definiert die in der folgenden Tabelle zusammengefassten drei Zugriffsmodi.

Zugriffsmodus	CLI-Abkürzung	Beschreibung
ReadWriteMany	RWX	Kubernetes kann das Volume auf vielen Knoten mit Lese- und Schreibzugriff mounten.
ReadOnlyMany	ROX	Kubernetes kann das Volume auf vielen Knoten schreibgeschützt mounten.
ReadWriteOnce	RWO	Kubernetes kann das Volume auf einem einzigen Knoten mit Lese- und Schreibzugriff mounten.

Es ist wichtig zu erwähnen, dass die Anforderungen dem besten verfügbaren PV zugeordnet werden, normalerweise mit einem ähnlichen Zugriffsmodus. Die unterstützten Modi hängen jedoch von den Funktionen des Anbieters ab. Beispielsweise kann ein PVC mit RWO vorliegen, der ein NFS PV anfordert. Dieser wird zugeordnet, da RWO von NFS unterstützt wird. Umgekehrt ist dies nicht möglich. Die Anforderung bleibt im Status pending.

So fügen Sie die PVC zur Anwendung hinzu:

```
...output omitted...
spec:
  volumes:
    - name: example-pv-storage
      persistentVolumeClaim:
        claimName: example-pv-claim
  containers:
    - name: example-application
      image: registry.redhat.io/rhel8/example-app
      ports:
        - containerPort: 1234
      volumeMounts:
        - mountPath: "/var/lib/example-app"
          name: example-pv-storage
...output omitted...
```

Anforderungen für persistente Volumes löschen

Um ein Volume zu löschen, verwenden Sie den Befehl `oc delete`, um die Anforderung für das persistente Volume zu löschen. Die Storage-Klasse nimmt das Volume zurück, nachdem die PVC entfernt wurde.

```
[user@host ~]$ oc delete pvc/example-pvc-storage
```



Literaturhinweise

Weitere Informationen zum persistenten Storage finden Sie im Kapitel *Understanding persistent storage* in der Red Hat OpenShift Container Platform 4.6 Storage-Dokumentation unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/storage/index#understanding-persistent-storage

Weitere Informationen zum temporären Storage finden Sie im Kapitel *Understanding ephemeral storage* in der Red Hat OpenShift Container Platform 4.6 Storage-Dokumentation unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/storage/index#understanding-ephemeral-storage

► Angeleitete Übung

Einführung in OpenShift Dynamic Storage

In dieser Übung stellen Sie eine PostgreSQL-Datenbank mit einer Anforderung für ein persistentes Volume bereit und identifizieren das dynamisch zugewiesene Volume.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Identifizieren der standardmäßigen Storage-Einstellungen eines OpenShift-Clusters.
- Anforderungen für persistente Volumes erstellen
- Persistente Volumes verwalten.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Mit dem Befehl wird sichergestellt, dass die Cluster-API erreichbar ist und die für diese Übung erforderlichen Dateien heruntergeladen werden.

```
[student@workstation ~]$ lab install-storage start
```

Anweisungen

► 1. Melden Sie sich beim OpenShift-Cluster an.

- 1.1. Rufen Sie die Kursumgebungs-Konfigurationsdatei auf, auf die unter `/usr/local/etc/ocp4.config` zugegriffen werden kann.

```
[student@workstation ~]$ source /usr/local/etc/ocp4.config
```

- 1.2. Melden Sie sich als Benutzer `kubeadmin` bei dem Cluster an. Akzeptieren Sie das unsichere Zertifikat, wenn Sie dazu aufgefordert werden.

```
[student@workstation ~]$ oc login -u kubeadmin -p ${RHT_OCP4_KUBEADM_PASSWD} \
>   https://api.ocp4.example.com:6443
...output omitted...
```

► 2. Erstellen Sie ein neues Projekt namens `install-storage`.

```
[student@workstation ~]$ oc new-project install-storage
Now using project "install-storage" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

Kapitel 2 | Überprüfen der Integrität eines Clusters

- 3. Überprüfen Sie die standardmäßige Storage-Klasse.

```
[student@workstation ~]$ oc get storageclass
NAME          PROVISIONER      RECLAIMPOLICY  ...
nfs-storage (default)  nfs-storage-provisioner Delete  ...
```

- 4. Erstellen Sie eine neue Datenbankbereitstellung mit dem Container-Image unter `registry.redhat.io/rhel8/postgresql-12:1-43`.

```
[student@workstation ~]$ oc new-app --name postgresql-persistent \
>   --docker-image registry.redhat.io/rhel8/postgresql-13:1-7 \
>   -e POSTGRESQL_USER=redhat \
>   -e POSTGRESQL_PASSWORD=redhat123 \
>   -e POSTGRESQL_DATABASE=persistentdb
...output omitted...
--> Creating resources ...
imagestream.image.openshift.io "postgresql-persistent" created
deployment.apps "postgresql-persistent" created
service "postgresql-persistent" created
--> Success
...output omitted...
```

**Anmerkung**

Zur Vereinfachung enthält die Datei `~/D0280/labs/install-storage/commands.txt` einige Befehle, die Sie kopieren und einfügen können.

- 5. Fügen Sie ein persistentes Volume für die PostgreSQL-Datenbank hinzu.

- 5.1. Erstellen Sie eine neue Anforderung für ein persistentes Volume, um ein neues Volume zur Bereitstellung `postgresql-persistent` hinzuzufügen.

```
[student@workstation ~]$ oc set volumes deployment/postgresql-persistent \
>   --add --name postgresql-storage --type pvc --claim-class nfs-storage \
>   --claim-mode rwo --claim-size 10Gi --mount-path /var/lib/pgsql \
>   --claim-name postgresql-storage
deployment.apps/postgresql-persistent volume updated
```

- 5.2. Überprüfen Sie, ob die neue PVC erfolgreich erstellt wurde.

```
[student@workstation ~]$ oc get pvc
NAME          STATUS    ...  CAPACITY  ACCESS MODES  STORAGECLASS  AGE
postgresql-storage  Bound    ...  10Gi       RWO          nfs-storage  25s
```

- 5.3. Überprüfen Sie, ob das neue PV erfolgreich erstellt wurde.

```
[student@workstation ~]$ oc get pv \
>   -o custom-columns=NAME:.metadata.name,CLAIM:.spec.claimRef.name
NAME           CLAIM
pvc-26cc804a-4ec2-4f52-b6e5-84404b4b9def  image-registry-storage
pvc-65c3cce7-45eb-482d-badf-a6469640bd75  postgresql-storage
```

Kapitel 2 | Überprüfen der Integrität eines Clusters

- 6. Füllen Sie die Datenbank mit dem Skript ~/D0280/labs/install-storage/init_data.sh.

- 6.1. Führen Sie das Skript init_data.sh aus.

```
[student@workstation ~]$ cd ~/D0280/labs/install-storage  
[student@workstation install-storage]$ ./init_data.sh  
Populating characters table  
CREATE TABLE  
INSERT 0 5
```

- 6.2. Verwenden Sie das Skript ~/D0280/labs/install-storage/check_data.sh, um zu überprüfen, ob die Datenbank erfolgreich mit Daten gefüllt wurde.

```
[student@workstation install-storage]$ ./check_data.sh  
Checking characters table  


| id | name                    | nationality                      |
|----|-------------------------|----------------------------------|
| 1  | Wolfgang Amadeus Mozart | Prince-Archbishopric of Salzburg |
| 2  | Ludwig van Beethoven    | Bonn, Germany                    |
| 3  | Johann Sebastian Bach   | Eisenach, Germany                |
| 4  | José Pablo Moncayo      | Guadalajara, México              |
| 5  | Niccolò Paganini        | Genoa, Italy                     |



(5 rows)


```

- 7. Entfernen Sie die Bereitstellung postgresql-persistent, und erstellen Sie eine weitere Bereitstellung namens postgresql-deployment2 mit demselben persistenten Volume. Überprüfen Sie, ob die Daten persistent sind.

- 7.1. Löschen Sie alle Ressourcen mit der Bezeichnung app=postgresql-persistent.

```
[student@workstation install-storage]$ oc delete all -l app=postgresql-persistent  
service "postgresql-persistent" deleted  
deployment.apps "postgresql-persistent" deleted  
imagestream.image.openshift.io "postgresql-persistent" deleted
```

- 7.2. Erstellen Sie die PostgreSQL-persistent2- Bereitstellung mit den gleichen Initialisierungsdaten wie die PostgreSQL-persistente Bereitstellung.

```
[student@workstation install-storage]$ oc new-app --name postgresql-persistent2 \  
> --docker-image registry.redhat.io/rhel8/postgresql-13:1-7 \  
> -e POSTGRESQL_USER=redhat \  
> -e POSTGRESQL_PASSWORD=redhat123 \  
> -e POSTGRESQL_DATABASE=persistentdb  
...output omitted...  
--> Creating resources ...  
imagestream.image.openshift.io "postgresql-persistent2" created  
deployment.apps "postgresql-persistent2" created  
service "postgresql-persistent2" created  
--> Success  
...output omitted...
```

Kapitel 2 | Überprüfen der Integrität eines Clusters

- 7.3. Verwenden Sie das Skript `~/D0280/labs/install-storage/check_data.sh`, um sich zu vergewissern, dass die Datenbank keine Zeichentabelle enthält.

```
[student@workstation install-storage]$ ./check_data.sh
Checking characters table
ERROR: 'characters' table does not exist
```

- 7.4. Fügen Sie die vorhandene Anforderung für das persistente Volume `postgresql-persistent` zur Bereitstellung `postgresql-persistent2` hinzu.

```
[student@workstation install-storage]$ oc set volumes \
>   deployment/postgresql-persistent2 \
>   --add --name postgresql-storage --type pvc \
>   --claim-name postgresql-storage --mount-path /var/lib/pgsql
deployment.apps/postgresql-persistent2 volume updated
```

- 7.5. Überprüfen Sie mit dem Skript `~/D0280/labs/install-storage/check_data.sh`, ob das persistente Volume erfolgreich hinzugefügt wurde und ob der Pod `postgresql-persistent2` auf die zuvor erstellten Daten zugreifen kann.

```
[student@workstation install-storage]$ ./check_data.sh
Checking characters table
+-----+
| id | name | nationality |
+-----+
| 1 | Wolfgang Amadeus Mozart | Prince-Archbishopric of Salzburg |
| 2 | Ludwig van Beethoven | Bonn, Germany |
...output omitted...
```

- 8. Entfernen Sie die Bereitstellung `postgresql-persistent2` und die Anforderung für das persistente Volume.

- 8.1. Löschen Sie alle Ressourcen mit der Bezeichnung `app=postgresql-persistent2`.

```
[student@workstation install-storage]$ oc delete all -l app=postgresql-persistent2
service "postgresql-persistent2" deleted
deployment.apps "postgresql-persistent2" deleted
imagestream.image.openshift.io "postgresql-persistent2" deleted
```

- 8.2. Löschen Sie das persistente Volume, indem Sie die Anforderung für das persistente Volume `postgresql-storage` entfernen, und kehren Sie zu Ihrem Benutzerverzeichnis zurück.

```
[student@workstation install-storage]$ oc delete pvc/postgresql-storage
persistentvolumeclaim "postgresql-storage" deleted
```

- 8.3. Kehren Sie zum Verzeichnis `/home/student/` zurück.

```
[student@workstation install-storage]$ cd ~
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab install-storage finish
```

Hiermit ist die angeleitete Übung beendet.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Die Red Hat OpenShift Container Platform bietet zwei wichtige Installationsmethoden: Full-Stack-Automatisierung und bereits vorhandene Infrastrukturen.
- In zukünftigen Releases wird erwartet, dass weitere Cloud- und Virtualisierungsanbieter wie VMware, Red Hat Virtualization und IBM System Z hinzugefügt werden.
- Ein OpenShift-Knoten, der auf Red Hat Enterprise Linux CoreOS basiert, führt nur sehr wenige lokale Services aus, die einen direkten Zugriff auf einen Knoten benötigen, um den Status zu überprüfen. Die meisten Systemdienste werden als Container ausgeführt; wichtigste Ausnahmen hierbei sind die CRI-O-Container-Engine und das Kubelet.
- Die Befehle `oc get node`, `oc adm top`, `oc adm node-logs` und `oc debug` enthalten Fehlerbehebungsinformationen zu OpenShift-Knoten.

Kapitel 3

Konfigurieren von Autorisierung und Authentifizierung

Ziel

Konfigurieren der Authentifizierung mit dem HTPasswd-Identitätsanbieter und Zuweisen von Rollen zu Benutzern und Gruppen

Ziele

- Konfigurieren des HTPasswd-Identitätsanbieters für die OpenShift-Authentifizierung
- Rollenbasierte Zugriffskontrollen definieren und Berechtigungen für Benutzer anwenden.

Abschnitte

- Konfigurieren der Identitätsanbieter (und angeleitete Übung)
- Definieren und Anwenden von Berechtigungen mit RBAC (und angeleitete Übung)

Praktische Übung

Überprüfen der Integrität eines Clusters

Konfigurieren der Identitätsanbieter

Ziele

Nach Abschluss dieses Abschnitts sollten Sie den HTPasswd-Identitätsanbieter für die OpenShift-Authentifizierung konfigurieren können.

Beschreiben von OpenShift-Benutzern und -Gruppen

Es gibt eine Reihe von OpenShift-Ressourcen zu den Themen Authentifizierung und Autorisierung. Die folgende Liste enthält die wichtigsten Ressourcentypen und deren Definitionen:

Benutzer

In der OpenShift Container Platform-Architektur sind Benutzer Entitäten, die mit dem API-Server interagieren. Die Benutzerressource stellt einen Akteur innerhalb des Systems dar. Sie weisen Berechtigungen hinzu, indem Sie Rollen direkt zum Benutzer oder zu den Gruppen, zu denen der Benutzer gehört, hinzufügen.

Identität

Die Identitätsressource protokolliert die erfolgreichen Authentifizierungsversuche eines bestimmten Benutzers und Identitätsanbieters. Alle Daten zur Authentifizierungsquelle werden in der Identität gespeichert. Nur eine einzelne Benutzerressource ist einer Identitätsressource zugeordnet.

Servicekonto

In OpenShift können Anwendungen unabhängig voneinander mit der API kommunizieren, wenn keine Benutzeranmeldedaten abgerufen werden können. Um die Integrität der Anmeldedaten eines regulären Benutzers beizubehalten, werden die Anmeldedaten nicht freigegeben, und stattdessen werden Dienstkonten verwendet. Mit Servicekonten können Sie den API-Zugriff steuern, ohne die Anmeldedaten eines regulären Benutzers verwenden zu müssen.

Gruppe

Eine Gruppe stellt bestimmte Benutzer dar. Benutzer werden einer oder mehreren Gruppen zugeordnet. Gruppen werden eingesetzt, wenn Autorisierungsrichtlinien implementiert werden, um mehreren Benutzern gleichzeitig Berechtigungen zuzuweisen. Wenn Sie z. B. zwanzig Benutzern den Zugriff auf Objekte in einem Projekt erlauben möchten, können Sie eine Gruppe verwenden, anstatt jedem Benutzer den Zugriff einzeln zu gewähren. OpenShift Container Platform ermöglicht zudem Systemgruppen oder virtuelle Gruppen, die automatisch vom Cluster bereitgestellt werden.

Rolle

Eine Rolle definiert aus einer Reihe von Berechtigungen, mit denen ein Benutzer API-Operationen über einen oder mehrere Ressourcentypen durchführen kann. Sie gewähren Benutzern, Gruppen und Servicekonten Berechtigungen, indem Sie ihnen Rollen zuweisen.

Benutzer- und Identitätsressourcen werden in der Regel nicht im Voraus erstellt. Sie werden von OpenShift nach einer erfolgreichen interaktiven Anmeldung mit OAuth meist automatisch erstellt.

Authentifizieren von API-Anforderungen

Autorisierung und Authentifizierung sind die beiden Sicherheitsebenen, die für die Interaktion eines Benutzers mit dem Cluster zuständig sind. Wenn ein Benutzer eine Anfrage an die API stellt, ordnet die API den Benutzer zur Anfrage zu. Die Authentifizierungsebene authentifiziert den Benutzer. Nach erfolgreicher Authentifizierung entscheidet die Autorisierungsebene, ob die API-Anforderung akzeptiert oder abgelehnt wird. Die Autorisierungsebene verwendet Role-Based Access Control-Richtlinien (RBAC), um die Berechtigungen der Benutzer zu ermitteln.

Die OpenShift-API verfügt über zwei Methoden zur Authentifizierung von Anforderungen:

- OAuth-Zugriffstoken
- X.509-Client-Zertifikate

Wenn die Anforderung kein Zugriffstoken oder Zertifikat enthält, weist die Authentifizierungsebene ihr den virtuellen Benutzer `system:anonymous` und die virtuelle Gruppe `system:unauthenticated` zu.

Einführung in den Authentifizierungsoperator

Die OpenShift Container Platform verfügt über einen Authentifizierungsoperator, der einen OAuth-Server ausführt. Der OAuth-Server stellt Benutzern OAuth-Zugriffstoken zur Verfügung, wenn sie versuchen, sich bei der API zu authentifizieren. Ein Identitätsanbieter muss konfiguriert und für den OAuth-Server verfügbar sein. Der OAuth-Server verwendet einen Identitätsanbieter, um die Identität des Requestors zu validieren. Der Server ordnet den Benutzer zur Identität zu und erstellt das OAuth-Zugriffstoken für den Benutzer. OpenShift erstellt nach einer erfolgreichen Anmeldung automatisch Identitäts- und Benutzerressourcen.

Einführung in Identitätsanbieter

Der OpenShift OAuth-Server kann verschiedene Identitätsanbieter verwenden. Die folgenden Listen enthalten die gebräuchlichsten:

HTPasswd

Validiert Benutzernamen und Passwörter anhand eines Secrets, in dem die mit dem `htpasswd` generierten Anmeldedaten gespeichert werden.

Keystone

Aktiviert die gemeinsame Authentifizierung mit einem OpenStack Keystone V3-Server.

LDAP

Konfiguriert den LDAP-Identitätsanbieter, um Benutzernamen und Passwörter anhand eines LDAPv3-Servers mithilfe der einfachen BIND-Authentifizierung zu validieren.

GitHub oder GitHub Enterprise

Konfiguriert einen GitHub-Identitätsanbieter, um Benutzernamen und Passwörter für GitHub oder den GitHub Enterprises OAuth-Authentifizierungsserver zu validieren.

OpenID Connect

Integration in einen OpenID Connect-Identitätsanbieter mit einem Autorisierungscode-Flow.

Die benutzerdefinierte OAuth-Ressource muss mit dem gewünschten Identitätsanbieter aktualisiert werden. Sie können mehrere Identitätsanbieter derselben oder verschiedener Typen für dieselbe benutzerdefinierte OAuth-Ressource definieren.

Authentifizierung als Cluster-Administrator

Bevor Sie einen Identitätsanbieter konfigurieren und Benutzer verwalten können, müssen Sie als Cluster-Administrator auf Ihren OpenShift-Cluster zugreifen. Ein frisch installierter OpenShift-Cluster bietet zwei Möglichkeiten zur Authentifizierung von API-Anforderungen mit Cluster-Administratorberechtigungen:

- Authentifizieren Sie sich als der virtuelle Benutzer `kubeadmin`. Bei erfolgreicher Authentifizierung wird ein OAuth-Zugriffstoken gewährt.
- Verwenden Sie die Datei `kubeconfig`, in der ein X.509-Client-Zertifikat ohne Ablaufdatum eingebettet ist.

Um zusätzliche Benutzer zu erstellen und ihnen unterschiedliche Zugriffsstufen zu gewähren, müssen Sie einen Identitätsanbieter konfigurieren und ihren Benutzern Rollen zuweisen.

Authentifizieren mit dem X.509-Zertifikat

Während der Installation erstellt das OpenShift-Installationsprogramm eine spezifische `kubeconfig`-Datei im Verzeichnis `auth`. Die Datei `kubeconfig` enthält spezifische Details und Parameter, die von der CLI verwendet werden, um einen Client mit dem richtigen API-Server zu verbinden (einschließlich X.509-Zertifikat).

Die Installationsprotokolle enthalten den Speicherort der Datei `kubeconfig`:

```
INFO Run 'export KUBECONFIG=root/auth/kubeconfig' to manage the cluster with 'oc'.
```



Anmerkung

In der Kursumgebung befindet sich die Datei `kubeconfig` auf dem Rechner utility unter `/home/lab/ocp4/auth/kubeconfig`.

Um die `kubeconfig`-Datei zum Authentifizieren von `oc`-Befehlen zu verwenden, müssen Sie die Datei auf Ihre Workstation kopieren und den absoluten oder relativen Pfad auf die Umgebungsvariable `KUBECONFIG` festlegen. Anschließend können Sie alle `oc`-Befehle ausführen, für die Cluster-Administratorberechtigungen erforderlich sind, ohne sich bei OpenShift anzumelden.

```
[user@host ~]$ export KUBECONFIG=/home/user/auth/kubeconfig  
[user@host ~]$ oc get nodes
```

Alternativ können Sie die Option `--kubeconfig` des Befehls `oc` verwenden.

```
[user@host ~]$ oc --kubeconfig /home/user/auth/kubeconfig get nodes
```

Authentifizieren mit dem virtuellen Benutzer

Nach Abschluss der Installation erstellt OpenShift den virtuellen Benutzer `kubeadmin`. Das `kubeadmin`-Secret im `kube-system`-Namespace enthält das gehashte Passwort für den Benutzer `kubeadmin`. Der Benutzer `kubeadmin` verfügt über Cluster-Administratorberechtigungen.

Kapitel 3 | Konfigurieren von Autorisierung und Authentifizierung

Das OpenShift-Installationsprogramm generiert dynamisch ein eindeutiges `kubeadmin`-Passwort für den Cluster. Die Installationsprotokolle enthalten die `kubeadmin`-Anmeldedaten für die Anmeldung beim Cluster. Die Cluster-Installationsprotokolle enthalten zudem die Anmeldung, das Passwort und die URL für den Konsolenzugriff.

```
...output omitted...
INFO The cluster is ready when 'oc login -u kubeadmin -p shdU_trbi_6ucX_edbu_aqop'
...output omitted...
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.ocp4.example.com
INFO Login to the console with user: kubeadmin, password: shdU_trbi_6ucX_edbu_aqop
```



Anmerkung

In der Kursumgebung wird das Passwort für den Benutzer `kubeadmin` auf dem Rechner utility unter `/home/lab/ocp4/auth/kubeconfig` gespeichert.

Löschen des virtuellen Benutzers

Nachdem Sie einen Identitätsanbieter definiert, einen neuen Benutzer erstellt und diesem Benutzer die Rolle `cluster-admin` zugewiesen haben, können Sie die Benutzeranmeldedaten für `kubeadmin` entfernen, um die Cluster-Sicherheit zu erhöhen.

```
[user@host ~]$ oc delete secret kubeadmin -n kube-system
```



Warnung

Wenn Sie das `kubeadmin`-Secret löschen, bevor Sie einen anderen Benutzer mit Cluster-Admin-Berechtigungen konfiguriert haben, können Sie Ihren Cluster nur noch mit der Datei `kubeconfig` verwalten. Wenn Sie keine Kopie dieser Datei an einem sicheren Ort haben, gibt es keine Möglichkeit, den Administratorzugriff auf Ihren Cluster wiederherzustellen. Die einzige Alternative ist die Zerstörung und Neuinstallation Ihres Clusters.



Warnung

Löschen Sie während dieses Kurses den Benutzer `kubeadmin` **zu keinem Zeitpunkt**. Der Benutzer `kubeadmin` ist für die Übungsarchitektur des Kurses unerlässlich. Wenn Sie den Benutzer `kubeadmin` löschen, wird die Lab-Umgebung beschädigt. Daher müssen Sie eine neue Lab-Umgebung erstellen.

Konfigurieren des HTPasswd-Identitätsanbieters

Der HTPasswd-Identitätsanbieter validiert Benutzer gegen ein Secret, das Benutzernamen und Passwörter enthält, die mit dem Befehl `htpasswd` aus dem Apache HTTP Server-Projekt generiert wurden. Nur ein Cluster-Administrator darf die Daten im HTPasswd-Secret ändern. Reguläre Benutzer können ihre eigenen Passwörter nicht ändern.

Das Verwalten von Benutzern mit dem HTPasswd-Identitätsanbieter kann für eine Proof-of-Concept-Umgebung mit einer kleinen Gruppe von Benutzern ausreichend sein. In den meisten

Produktionsumgebungen ist jedoch ein leistungsfähigerer Identitätsanbieter erforderlich, der in das Identitätsmanagementsystem des Unternehmens integriert wird.

Konfigurieren der benutzerdefinierten OAuth-Ressource

Um den HTPasswd-Identitätsanbieter verwenden zu können, muss die benutzerdefinierte OAuth-Ressource bearbeitet und dem Array `.spec.identityProviders` ein Eintrag hinzugefügt werden:

```
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
  name: cluster
spec:
  identityProviders:
    - name: my_htpasswd_provider ❶
      mappingMethod: claim ❷
      type: HTPasswd
      htpasswd:
        fileData:
          name: htpasswd-secret ❸
```

- ❶ Diesem Anbieternamen werden Anbieter-Benutzernamen vorangestellt, um einen Identitätsnamen zu bilden.
- ❷ Steuert, wie Zuordnungen zwischen Anbieteridentitäten und Benutzerobjekten erstellt werden.
- ❸ Ein vorhandenes Secret, das die mit dem Befehl `htpasswd` generierten Daten enthält.

Aktualisieren der benutzerdefinierten OAuth-Ressource

Um die benutzerdefinierte OAuth-Ressource zu aktualisieren, verwenden Sie den Befehl `oc get`, um die vorhandene OAuth-Cluster-Ressource in eine Datei mit YAML-Format zu exportieren.

```
[user@host ~]$ oc get oauth cluster -o yaml > oauth.yaml
```

Öffnen Sie anschließend die entstandene Datei in einem Texteditor, und nehmen Sie die erforderlichen Änderungen an den Einstellungen des eingebetteten Identitätsanbieters vor.

Nachdem Sie Änderungen vorgenommen und die Datei gespeichert haben, müssen Sie die neue benutzerdefinierte Ressource mit dem Befehl `oc replace` anwenden.

```
[user@host ~]$ oc replace -f oauth.yaml
```

Verwalten von Benutzern mit dem HTPasswd-Identitätsanbieter

Die Verwaltung von Benutzeranmeldedaten mit dem HTPasswd-Identitätsanbieter erfordert das Erstellen einer temporären `htpasswd`-Datei, die Durchführung von Änderungen an der Datei und die Anwendung dieser Änderungen auf das Secret.

Erstellen einer HTPasswd-Datei

Das Paket `httpd-utils` enthält das Dienstprogramm `htpasswd`. Das Paket `httpd-utils` muss auf Ihrem System installiert und verfügbar sein.

Erstellen Sie die `htpasswd`-Datei:

```
[user@host ~]$ htpasswd -c -B -b /tmp/htpasswd student redhat123
```



Wichtig

Verwenden Sie die Option `-c` nur, wenn Sie eine neue Datei erstellen. Die Option `-c` ersetzt den gesamten Dateiinhalt, wenn die Datei bereits vorhanden ist.

Fügen Sie Anmelddaten hinzu bzw. aktualisieren Sie diese.

```
[user@host ~]$ htpasswd -b /tmp/htpasswd student redhat1234
```

Löschen Sie die Anmelddaten.

```
[user@host ~]$ htpasswd -D /tmp/htpasswd student
```

Erstellen des HTPasswd-Secrets

Um den HTPasswd-Anbieter verwenden zu können, müssen Sie ein Secret erstellen, das die `htpasswd`-Dateidaten enthält. Im folgenden Beispiel hat das Secret den Namen `htpasswd-secret`.

```
[user@host ~]$ oc create secret generic htpasswd-secret \
>   --from-file htpasswd=/tmp/htpasswd -n openshift-config
```



Wichtig

Für ein Secret, das vom HTPasswd-Identitätsanbieter verwendet wird, müssen Sie das Präfix `htpasswd=` hinzufügen, bevor Sie den Pfad zur Datei angeben.

Extrahieren von Secret-Daten

Beim Hinzufügen oder Entfernen von Benutzern dürfen Administratoren nicht davon ausgehen, dass die lokale `htpasswd`-Datei gültig ist. Darüber hinaus befindet sich der Administrator möglicherweise nicht einmal auf einem System mit der `htpasswd`-Datei. In einem realen Szenario würde der Administrator den Befehl `oc extract` verwenden.

Standardmäßig speichert der Befehl `oc extract` jeden Schlüssel in einer Konfigurations-Map oder einem Secret als separate Datei. Alternativ können alle Daten dann an eine Datei umgeleitet oder als Standardausgabe angezeigt werden. Um Daten aus dem `htpasswd-secret`-Secret in das `/tmp/-Verzeichnis` zu extrahieren, verwenden Sie den folgenden Befehl. Die Option `--confirm` ersetzt die Datei, wenn sie bereits vorhanden ist.

```
[user@host ~]$ oc extract secret/htpasswd-secret -n openshift-config \
> --to /tmp/ --confirm /tmp/htpasswd
```

Aktualisieren des HTPasswd Secret

Das Secret muss nach dem Hinzufügen, Ändern oder Löschen von Benutzern aktualisiert werden. Verwenden Sie den Befehl `oc set data secret`, um ein Secret zu aktualisieren. Falls die Datei nicht den Namen `htpasswd` hat, müssen Sie `htpasswd=` angeben, um den `htpasswd`-Schlüssel im Secret zu aktualisieren.

Der folgende Befehl aktualisiert das `htpasswd-secret`-Secret im Namespace `openshift-config` unter Verwendung des Inhalts der Datei `/tmp/htpasswd`.

```
[user@host ~]$ oc set data secret/htpasswd-secret \
> --from-file htpasswd=/tmp/htpasswd -n openshift-config
```

Nach der Aktualisierung des Secrets stellt der Operator OAuth die Pods im `openshift-Authentication`-Namespace erneut bereit. Überwachen Sie die Neubereitstellung der neuen OAuth-Pods, indem Sie Folgendes ausführen:

```
[user@host ~]$ watch oc get pods -n openshift-authentication
```

Ergänzungen, Änderungen oder Löschungen des Secrets können getestet werden, nachdem die neuen Pods bereitgestellt wurden.

Löschen von Benutzern und Identitäten

Wenn ein Szenario auftritt, in dem Sie einen Benutzer löschen müssen, genügt es nicht, den Benutzer aus dem Identitätsanbieter zu löschen. Die Benutzer- und Identitätsressourcen müssen ebenfalls gelöscht werden.

Sie müssen das Passwort aus dem `htpasswd`-Secret entfernen, den Benutzer aus der lokalen `htpasswd`-Datei entfernen und dann das Secret aktualisieren.

Führen Sie den folgenden Befehl aus, um den Benutzer aus `htpasswd` zu löschen:

```
[user@host ~]$ htpasswd -D /tmp/htpasswd manager
```

Aktualisieren Sie das Secret, um das Benutzerpasswort vollständig zu entfernen.

```
[user@host ~]$ oc set data secret/htpasswd-secret \
> --from-file htpasswd=/tmp/htpasswd -n openshift-config
```

Entfernen Sie die Benutzerressource mit dem folgenden Befehl:

```
[user@host ~]$ oc delete user manager
user.user.openshift.io "manager" deleted
```

Identitätsressourcen enthalten den Namen des Identitätsanbieters. Um die Identitätsressource für den Benutzer `manager` zu löschen, suchen Sie nach der Ressource und löschen Sie sie.

```
[user@host ~]$ oc get identities | grep manager
my_htpasswd_provider:manager    my_htpasswd_provider    manager      manager   ...
[user@host ~]$ oc delete identity my_htpasswd_provider:manager
identity.user.openshift.io "my_htpasswd_provider:manager" deleted
```

Zuweisen von Administratorberechtigungen

Die Cluster-weite `cluster-admin`-Rolle gewährt Benutzern und Gruppen Berechtigungen für die Cluster-Administration. Benutzer mit dieser Rolle können jede Aktion für alle Ressourcen im Cluster ausführen. Im folgenden Beispiel wird die Rolle `cluster-admin` dem Benutzer `student` zugewiesen.

```
[user@host ~]$ oc adm policy add-cluster-role-to-user cluster-admin student
```



Anmerkung

Weitere Informationen zu Identitätsanbietern finden Sie im Kapitel „Understanding identity provider configuration“ in der Dokumentation zu „Red Hat OpenShift Container Platform 4.6 Authentication and authorization“ unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/authentication_and_authorization/index#understanding-identity-provider.

► Angeleitete Übung

Konfigurieren der Identitätsanbieter

In dieser Übung konfigurieren Sie den HTPasswd-Identitätsanbieter und erstellen Benutzer für Cluster-Administratoren.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen von Benutzern und Passwörtern für die HTPasswd-Authentifizierung.
- Konfigurieren des HTPasswd-Identitätsanbieters für die HTPasswd-Authentifizierung.
- Zuweisen von Cluster-Administratorrechten für Benutzer.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

```
[student@workstation ~]$ lab auth-provider start
```

Mit dem Befehl wird sichergestellt, dass die Cluster-API erreichbar ist, das Paket `httpd-utils` installiert ist und die Authentifizierungseinstellungen auf die Standardeinstellungen für die Installation konfiguriert sind.

Anweisungen

- 1. Fügen Sie einen Eintrag für zwei htpasswd-Benutzer, `admin` und `developer`, hinzu. Weisen Sie dem `admin` das Passwort `redhat` und dem `developer` das Passwort `developer` zu.
- 1.1. Rufen Sie die Kursumgebungs-Konfigurationsdatei auf, auf die unter `/usr/local/etc/ocp4.config` zugegriffen werden kann.

```
[student@workstation ~]$ source /usr/local/etc/ocp4.config
```



Warnung

Löschen Sie während dieses Kurses den Benutzer `kubeadmin` **zu keinen**

Zeitpunkt. Der Benutzer `kubeadmin` ist für die Übungsarchitektur des Kurses unerlässlich. Wenn Sie den Benutzer `kubeadmin` löschen, wird die Lab-Umgebung beschädigt. Daher müssen Sie eine neue Lab-Umgebung erstellen.

- 1.2. Erstellen Sie eine HTPasswd-Authentifizierungsdatei mit dem Namen `htpasswd` im Verzeichnis `~/D0280/labs/auth-provider/`. Fügen Sie den Benutzer `admin` mit dem Passwort `redhat` hinzu. Sie können einen beliebigen Namen für die Datei verwenden. In dieser Übung wird die Datei `~/D0280/labs/auth-provider/htpasswd` verwendet.

Kapitel 3 | Konfigurieren von Autorisierung und Authentifizierung

Verwenden Sie den Befehl `htpasswd`, um die HTPasswd-Authentifizierungsdatei mit den Benutzernamen und verschlüsselten Passwörtern zu füllen. Die Option `-B` verwendet die bcrypt-Verschlüsselung. Der Befehl `htpasswd` verwendet standardmäßig die MD5-Verschlüsselung, wenn Sie keine Verschlüsselungsoption festlegen.

```
[student@workstation ~]$ htpasswd -c -B -b ~/D0280/labs/auth-provider/htpasswd \
> admin redhat
Adding password for user admin
```

- Fügen Sie den Benutzer `developer` mit dem Passwort `developer` zur Datei `~/D0280/labs/auth-provider/htpasswd` hinzu.

```
[student@workstation ~]$ htpasswd -B -b ~/D0280/labs/auth-provider/htpasswd \
> developer developer
Adding password for user developer
```

- Überprüfen Sie die Inhalte von `~/D0280/labs/auth-provider/htpasswd`, und vergewissern Sie sich, dass sie zwei Einträge mit gehaschten Passwörtern enthält: eine für den Benutzer `admin` und eine weitere für den Benutzer `developer`.

```
[student@workstation ~]$ cat ~/D0280/labs/auth-provider/htpasswd
admin:$2y$05$QPuzHdl06IDkJsST.tdkZuSmgjUHV1XeYU4FjxhQrFqKL7hs2ZUL6
developer:$apr1$0Nzmc1rh$yGtne1k.JX6L5s5wNa2ye.
```

- 2. Melden Sie sich bei OpenShift an, und erstellen Sie ein Secret, das die Benutzerdatei HTPasswd enthält.

- Melden Sie sich als Benutzer `kubeadmin` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u kubeadmin -p ${RHT_OCP4_KUBEADM_PASSWORD} \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- Erstellen Sie aus der Datei `/home/student/D0280/labs/auth-provider/htpasswd` ein Secret. Um den HTPasswd-Identitätsanbieter verwenden zu können, müssen Sie ein Secret mit einem Schlüssel namens `htpasswd` definieren, der die HTPasswd-Benutzerdatei `/home/student/D0280/labs/auth-provider/htpasswd` enthält.



Wichtig

Für ein Secret, das vom HTPasswd-Identitätsanbieter verwendet wird, müssen Sie das Präfix `htpasswd=` hinzufügen, bevor Sie den Pfad zur Datei angeben.

```
[student@workstation ~]$ oc create secret generic localusers \
> --from-file htpasswd=/home/student/D0280/labs/auth-provider/htpasswd \
> -n openshift-config
secret/localusers created
```

- 2.3. Weisen Sie dem Benutzer `admin` die Rolle `cluster-admin` zu.



Anmerkung

Die Ausgabe weist darauf hin, dass der Benutzer `admin` nicht gefunden wurde und kann problemlos ignoriert werden.

```
[student@workstation ~]$ oc adm policy add-cluster-role-to-user \
>   cluster-admin admin
...output omitted...
clusterrole.rbac.authorization.k8s.io/cluster-admin added: "admin"
```

- 3. Aktualisieren Sie den HTPasswd-Identitätsanbieter für den Cluster so, dass sich Ihre Benutzer authentifizieren können. Konfigurieren Sie die benutzerdefinierte Ressourcendatei, und aktualisieren Sie den Cluster.

- 3.1. Exportieren Sie die vorhandene OAuth-Ressource in eine Datei mit dem Namen `oauth.yaml` im Verzeichnis `~/D0280/labs/auth-provider`.

```
[student@workstation ~]$ oc get oauth cluster \
>   -o yaml > ~/D0280/labs/auth-provider/oauth.yaml
```



Anmerkung

Sie finden eine `oauth.yaml`-Datei mit der vollständigen benutzerdefinierten Ressourcendatei unter `~/D0280/solutions/auth-provider`.

- 3.2. Bearbeiten Sie die Datei `~/D0280/labs/auth-provider/oauth.yaml` mit Ihrem bevorzugten Texteditor. Sie können die Namen der Strukturen `identityProviders` und `fileData` festlegen. Verwenden Sie für diese Übung die Werte `myusers` und `localusers`.

Die fertige benutzerdefinierte Ressource sollte mit Folgendem übereinstimmen. Beachten Sie, dass sich `htpasswd`, `mappingMethod`, `name` und `type` auf derselben Einrückungsebene befinden.

```
apiVersion: config.openshift.io/v1
kind: OAuth
...output omitted...
spec:
  identityProviders:
    - htpasswd:
        fileData:
          name: localusers
        mappingMethod: claim
        name: myusers
        type: HTPasswd
```

- 3.3. Wenden Sie die im vorherigen Schritt definierte benutzerdefinierte Ressource an.

```
[student@workstation ~]$ oc replace -f ~/DO280/labs/auth-provider/oauth.yaml  
oauth.config.openshift.io/cluster replaced
```



Anmerkung

Pods im Namespace `openshift-authentication` werden erneut bereitgestellt, wenn der Befehl `oc replace` erfolgreich ist. Wenn das zuvor erstellte Secret korrekt erstellt wurde, können Sie sich mit dem HTPasswd-Identitätsanbieter anmelden.

- ▶ 4. Melden Sie sich als `admin` und als `developer` an, um die HTPasswd-Benutzerkonfiguration zu verifizieren.
 - 4.1. Melden Sie sich als Benutzer `admin` beim Cluster an, um zu überprüfen, ob die HTPasswd-Authentifizierung ordnungsgemäß konfiguriert ist. Der Authentifizierungsoperator benötigt einige Zeit, um die Konfigurationsänderungen aus dem vorherigen Schritt zu laden.



Anmerkung

Wenn die Authentifizierung fehlschlägt, warten Sie einige Augenblicke, und versuchen Sie es erneut.

```
[student@workstation ~]$ oc login -u admin -p redhat  
Login successful.  
...output omitted...
```

- 4.2. Verwenden Sie den Befehl `oc get nodes`, um sicherzustellen, dass der Benutzer `admin` über die Rolle `cluster-admin` verfügt.

```
[student@workstation ~]$ oc get nodes  
NAME      STATUS    ROLES          AGE     VERSION  
master01   Ready     master,worker  2d2h    v1.19.0+d856161  
master02   Ready     master,worker  2d2h    v1.19.0+d856161  
master03   Ready     master,worker  2d2h    v1.19.0+d856161
```

- 4.3. Melden Sie sich als Benutzer `developer` beim Cluster an, um zu überprüfen, ob die HTPasswd-Authentifizierung ordnungsgemäß konfiguriert ist.

```
[student@workstation ~]$ oc login -u developer -p developer  
Login successful.  
...output omitted...
```

- 4.4. Verwenden Sie den Befehl `oc get nodes`, um sicherzustellen, dass die Benutzer `developer` und `admin` nicht dieselbe Zugriffsebene besitzen.

```
[student@workstation ~]$ oc get nodes  
Error from server (Forbidden): nodes is forbidden: User "developer" cannot list  
resource "nodes" in API group "" at the cluster scope
```

- 4.5. Melden Sie sich als Benutzer `admin` an, und listen Sie die aktuellen Benutzer auf.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
[student@workstation ~]$ oc get users
NAME          UID           FULL NAME  IDENTITIES
admin         31f6ccd2-6c58-47ee-978d-5e5e3c30d617
developer     d4e77b0d-9740-4f05-9af5-ecfc08a85101
```

- 4.6. Zeigen Sie die Liste der aktuellen Identitäten an.

```
[student@workstation ~]$ oc get identity
NAME          IDP NAME    IDP USER NAME  USER NAME  USER UID
myusers:admin myusers      admin        admin      31f6ccd2-6c58-47...
myusers:developer myusers    developer    developer  d4e77b0d-9740-4f...
```

- 5. Erstellen Sie als Benutzer `admin` einen neuen HTPasswd-Benutzer namens `manager` und mit dem Passwort `redhat`.

- 5.1. Extrahieren Sie die Dateidaten aus dem Secret in die Datei `~/D0280/labs/auth-provider/htpasswd`.

```
[student@workstation ~]$ oc extract secret/localusers -n openshift-config \
>   --to ~/D0280/labs/auth-provider/ --confirm
/home/student/D0280/labs/auth-provider/htpasswd
```

- 5.2. Fügen Sie der Datei `~/D0280/labs/auth-provider/htpasswd` einen Eintrag für den zusätzlichen Benutzer `manager` mit dem Passwort `redhat` hinzu.

```
[student@workstation ~]$ htpasswd -b ~/D0280/labs/auth-provider/htpasswd \
>   manager redhat
Adding password for user manager
```

- 5.3. Überprüfen Sie den Inhalt der Datei `~/D0280/labs/auth-provider/htpasswd`, und stellen Sie sicher, dass sie drei Einträge mit gehaschten Passwörtern enthält: jeweils eines für den `admin`-, `developer`- und `manager`-Benutzer.

```
[student@workstation ~]$ cat ~/D0280/labs/auth-provider/htpasswd
admin:$2y$05$QPuzHdl06IDkJssT.tdkZuSmgjUHV1XeYU4FjxhQrFqKL7hs2ZUl6
developer:$apr1$0Nzmc1rh$yGtne1k.JX6L5s5wNa2ye.
manager:$apr1$CJ/tpha6a$sLhjPkIIAy755ZArTT5EH/
```

- 5.4. Das Secret muss nach dem Hinzufügen von Benutzern aktualisiert werden. Verwenden Sie den Befehl `oc set data secret`, um das Secret zu aktualisieren. Wenn ein Fehler auftritt, führen Sie den Befehl nach einigen Augenblicken erneut aus, da der oauth-Operator möglicherweise noch neu geladen wird.

```
[student@workstation ~]$ oc set data secret/localusers \
>   --from-file htpasswd=/home/student/D0280/labs/auth-provider/htpasswd \
>   -n openshift-config
secret/localusers data updated
```

- 5.5. Warten Sie einige Augenblicke, bis der Authentifizierungsoperator neu geladen wurde, und melden Sie sich dann als Benutzer **manager** beim Cluster an.



Anmerkung

Wenn die Authentifizierung fehlschlägt, warten Sie einige Augenblicke, und versuchen Sie es erneut.

```
[student@workstation ~]$ oc login -u manager -p redhat
Login successful.
...output omitted...
```

- ▶ 6. Erstellen Sie ein neues Projekt mit dem Namen **auth-provider**, und überprüfen Sie anschließend, ob der Benutzer **developer** auf das Projekt zugreifen kann.

- 6.1. Erstellen Sie als Benutzer **manager** ein neues **auth-provider**-Projekt.

```
[student@workstation ~]$ oc new-project auth-provider
Now using project "auth-provider" on server https://api.ocp4.example.com:6443".
...output omitted...
```

- 6.2. Melden Sie sich als Benutzer **developer** an, und versuchen Sie dann, das Projekt **auth-provider** zu löschen.

```
[student@workstation ~]$ oc login -u developer -p developer
Login successful.
...output omitted...
[student@workstation ~]$ oc delete project auth-provider
Error from server (Forbidden): projects.project.openshift.io "auth-provider"
is forbidden: User "developer" cannot delete resource "projects"
in API group "project.openshift.io" in the namespace "auth-provider"
```

- ▶ 7. Ändern Sie das Passwort des Benutzers **manager**.

- 7.1. Melden Sie sich als **admin** an und extrahieren Sie die Dateidaten aus dem Secret in die Datei **~/D0280/labs/auth-provider/htpasswd**.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
[student@workstation ~]$ oc extract secret/localusers -n openshift-config \
>   --to ~/D0280/labs/auth-provider/ --confirm
/home/student/D0280/labs/auth-provider/htpasswd
```

Kapitel 3 | Konfigurieren von Autorisierung und Authentifizierung

- 7.2. Generieren Sie ein zufälliges Benutzerpasswort, und weisen Sie es der Variablen `MANAGER_PASSWD` zu.

```
[student@workstation ~]$ MANAGER_PASSWD="$(openssl rand -hex 15)"
```

- 7.3. Aktualisieren Sie den Benutzer `manager`, um das in der Variablen `MANAGER_PASSWD` gespeicherte Passwort zu verwenden.

```
[student@workstation ~]$ htpasswd -b ~/D0280/labs/auth-provider/htpasswd \
>     manager ${MANAGER_PASSWD}
Updating password for user manager
```

- 7.4. Aktualisieren Sie das Secret.

```
[student@workstation ~]$ oc set data secret/localusers \
>     --from-file htpasswd=/home/student/D0280/labs/auth-provider/htpasswd \
>     -n openshift-config
secret/localusers data updated
```

**Anmerkung**

Wenn die Authentifizierung fehlschlägt, warten Sie einige Augenblicke, und versuchen Sie es erneut.

- 7.5. Melden Sie sich als Benutzer `manager` an, um das aktualisierte Passwort zu verifizieren.

```
[student@workstation ~]$ oc login -u manager -p ${MANAGER_PASSWD}
Login successful.
...output omitted...
```

► 8. Entfernen Sie den Benutzer `manager`.

- 8.1. Melden Sie sich als `admin` an und extrahieren Sie die Dateidaten aus dem Secret in die Datei `~/D0280/labs/auth-provider/htpasswd`.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
[student@workstation ~]$ oc extract secret/localusers -n openshift-config \
>     --to ~/D0280/labs/auth-provider/ --confirm
/home/student/D0280/labs/auth-provider/htpasswd
```

- 8.2. Löschen Sie den Benutzer `manager` aus der Datei `~/D0280/labs/auth-provider/htpasswd`.

```
[student@workstation ~]$ htpasswd -D ~/D0280/labs/auth-provider/htpasswd manager
Deleting password for user manager
```

- 8.3. Aktualisieren Sie das Secret.

Kapitel 3 | Konfigurieren von Autorisierung und Authentifizierung

```
[student@workstation ~]$ oc set data secret/localusers \
>   --from-file htpasswd=/home/student/D0280/labs/auth-provider/htpasswd \
>   -n openshift-config
secret/localusers data updated
```

- 8.4. Löschen Sie die Identitätsressource für den Benutzer manager.

```
[student@workstation ~]$ oc delete identity "myusers:manager"
identity.user.openshift.io "myusers:manager" deleted
```

- 8.5. Löschen Sie die Benutzerressource für den Benutzer manager.

```
[student@workstation ~]$ oc delete user manager
user.user.openshift.io manager deleted
```

- 8.6. Sie können sich jetzt nicht mehr mit dem Benutzer manager anmelden.

```
[student@workstation ~]$ oc login -u manager -p ${MANAGER_PASSWD}
Login failed (401 Unauthorized)
Verify you have provided correct credentials.
```

- 8.7. Führen Sie die aktuellen Benutzer auf, um sicherzustellen, dass der Benutzer manager gelöscht wurde.

```
[student@workstation ~]$ oc get users
NAME          UID                                     FULL NAME  IDENTITIES
admin        31f6cccd2-6c58-47ee-978d-5e5e3c30d617
developer    d4e77b0d-9740-4f05-9af5-ecfc08a85101           myusers:admin
                                         myusers:developer
```

- 8.8. Zeigen Sie die Liste der aktuellen Identitäten an, um zu überprüfen, ob die manager-Identität gelöscht wurde.

```
[student@workstation ~]$ oc get identity
NAME          IDP NAME  IDP USER NAME  USER NAME
myusers:admin  myusers   admin        admin      ...
myusers:developer  myusers   developer   developer  ...
```

- 8.9. Extrahieren Sie das Secret, und überprüfen Sie, ob nur die Benutzer admin und developer angezeigt werden. Durch die Verwendung von --to - wird das Secret an STDOUT gesendet und nicht in einer Datei gespeichert.

```
[student@workstation ~]$ oc extract secret/localusers -n openshift-config --to -
# htpasswd
admin:$2y$05$TizWp/2ct4Edn08gmeMBI09IXujpLqkKAJ0Nldxc/V2XYYMBf6wBy
developer:$apr1$8Bc6txgb$bwHke4cGRGk9C8tQLg.hi1
```

- 9. Entfernen Sie den Identitätsanbieter, und bereinigen Sie alle Benutzer.

- 9.1. Melden Sie sich als Benutzer kubeadmin an.

```
[student@workstation ~]$ oc login -u kubeadmin -p ${RHT_OCP4_KUBEADM_PASSWD}
Login successful.
...output omitted...
```

9.2. Löschen Sie das Projekt auth-provider.

```
[student@workstation ~]$ oc delete project auth-provider
project.project.openshift.io "auth-provider" deleted
```

9.3. Bearbeiten Sie die Ressource direkt, um den Identitätsanbieter aus OAuth zu entfernen:

```
[student@workstation ~]$ oc edit oauth
```

Löschen Sie alle Zeilen unter spec:, und hängen Sie dann {} nach spec: an.
Behalten Sie alle anderen Informationen in der Datei bei. Ihre spec:-Zeile sollte mit der Folgenden übereinstimmen:

```
...output omitted...
spec: {}
```

Speichern Sie Ihre Änderungen, und überprüfen Sie dann, ob der Befehl oc edit Ihre Änderungen übernommen hat:

```
oauth.config.openshift.io/cluster edited
```

9.4. Löschen Sie das localusers-Secret aus dem Namespace openshift-config.

```
[student@workstation ~]$ oc delete secret localusers -n openshift-config
secret "localusers" deleted
```

9.5. Löschen Sie alle Benutzerressourcen.

```
[student@workstation ~]$ oc delete user --all
user.user.openshift.io "admin" deleted
user.user.openshift.io "developer" deleted
```

9.6. Löschen Sie alle Identitätsressourcen.

```
[student@workstation ~]$ oc delete identity --all
identity.user.openshift.io "myusers:admin" deleted
identity.user.openshift.io "myusers:developer" deleted
```

Beenden

Führen Sie auf dem Rechner workstation den Befehl lab aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab auth-provider finish
```

Hiermit ist die angeleitete Übung beendet.

Definieren und Anwenden von Berechtigungen mit RBAC

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, rollenbasierte Zugriffskontrollen zu definieren und Berechtigungen für Benutzer anzuwenden.

Rollenbasierte Zugriffskontrolle (RBAC)

Role-based access control (RBAC) bzw. die rollenbasierte Zugriffskontrolle ist eine Technik zur Verwaltung des Zugriffs auf Ressourcen in einem Computersystem. In Red Hat OpenShift bestimmt die RBAC, ob ein Benutzer bestimmte Aktionen innerhalb des Clusters oder Projekts durchführen kann. Es gibt zwei Arten von Rollen, die je nach Verantwortungsbereich des Benutzers verwendet werden können: cluster und local.



Anmerkung

Die Autorisierung ist neben der Authentifizierung ein eigener Schritt.

Autorisierungsprozess

Der Autorisierungsprozess wird von Regeln, Rollen und Bindungen verwaltet.

RBAC-Objekt	Beschreibung
Regel	Zulässige Aktionen für Objekte oder Objektgruppen.
Rolle	Regelsätze. Benutzer und Gruppen können mehreren Rollen zugeordnet werden.
Binding	Zuweisung von Benutzern oder Gruppen zu einer Rolle.

RBAC-Geltungsbereich

Die Red Hat OpenShift Container Platform (RHOCOP) definiert je nach Umfang und Verantwortlichkeit der Benutzer zwei Gruppen von Rollen und Bindungen: Cluster-Rollen und lokale Rollen.

Rollenebene	Beschreibung
Cluster-Rolle	Benutzer oder Gruppen mit dieser Rolle können den OpenShift-Cluster verwalten.
Lokale Rolle	Benutzer oder Gruppen mit dieser Rollenebene können nur Elemente auf Projektebene verwalten.



Anmerkung

Die Cluster-Rollenbindungen haben Vorrang vor lokalen Rollenbindungen.

Verwalten der RBAC über die CLI

Cluster-Administratoren können den Befehl `oc adm policy` verwenden, um Cluster-Rollen und Namespace-Rollen hinzuzufügen und zu entfernen.

Um einem Benutzer eine Cluster-Rolle zuzuweisen, verwenden Sie den Sub-Befehl `add-cluster-role-to-user`:

```
[user@host ~]$ oc adm policy add-cluster-role-to-user cluster-role username
```

Beispiel: Um einen regulären Benutzer zum Cluster-Administrator zu ernennen, führen Sie den folgenden Befehl aus:

```
[user@host ~]$ oc adm policy add-cluster-role-to-user cluster-admin username
```

Um eine Cluster-Rolle für einen Benutzer zu entfernen, verwenden Sie den Sub-Befehl `remove-cluster-role-from-user`:

```
[user@host ~]$ oc adm policy remove-cluster-role-from-user cluster-role username
```

Beispiel: Um einen Cluster-Administrator zu einem regulären Benutzer zu ernennen, führen Sie den folgenden Befehl aus:

```
[user@host ~]$ oc adm policy remove-cluster-role-from-user cluster-admin username
```

Regeln werden durch eine Aktion und eine Ressource definiert. Beispielsweise ist die Regel `create user` Teil der Rolle `cluster-admin`.

Sie können den Befehl `oc adm policy who-can` verwenden, um zu bestimmen, ob ein Benutzer eine Aktion auf einer Ressource ausführen kann. Beispiel:

```
[user@host ~]$ oc adm policy who-can delete user
```

Standardrollen

OpenShift wird mit einer Reihe von standardmäßigen Cluster-Rollen ausgeliefert, die lokal oder dem gesamten Cluster zugewiesen werden können. Sie können diese Rollen für eine detaillierte Zugriffskontrolle auf OpenShift-Ressourcen ändern, aber es sind zusätzliche Schritte erforderlich, die in diesem Kurs nicht behandelt werden.

Standardrollen	Beschreibung
admin	Benutzer mit dieser Rolle können alle Projektressourcen verwalten und anderen Benutzern Zugriff auf das Projekt erteilen.
basic-user	Benutzer mit dieser Rolle haben Lesezugriff auf das Projekt.

Standardrollen	Beschreibung
cluster-admin	Benutzer mit dieser Rolle haben Superuser-Zugriff auf die Cluster-Ressourcen. Diese Benutzer können alle Aktionen im Cluster durchführen und haben die volle Kontrolle über alle Projekte.
cluster-status	Benutzer mit dieser Rolle können Cluster-Statusinformationen abrufen.
edit	Benutzer mit dieser Rolle können allgemeine Anwendungsressourcen wie Services und Bereitstellungen im Projekt erstellen, ändern oder aus dem Projekt entfernen. Sie können keine Verwaltungsressourcen wie Einschränzungsbereiche und Quoten bearbeiten oder Zugriffsberechtigungen auf das Projekt verwalten.
self-provisioner	Benutzer mit dieser Rolle können neue Projekte erstellen. Dies ist eine Cluster-Rolle, keine Projektrolle.
view	Benutzer mit dieser Rolle können Projektressourcen anzeigen, jedoch keine Projekt Ressourcen ändern.

Die admin-Rolle gewährt Benutzern – zusätzlich zur Möglichkeit, neue Anwendungen zu erstellen – Zugriff auf Projektressourcen wie Quoten und Einschränzungsbereiche. Benutzer mit der edit-Rolle haben ausreichende Zugriffsrechte, um in einem Projekt als Entwickler zu arbeiten. Die Rolle unterliegt jedoch den vom Projektadministrator konfigurierten Beschränkungen.

Projektadministratoren können den Befehl `oc policy` verwenden, um Namespace-Rollen hinzuzufügen und zu entfernen.

Mit dem Sub-Befehl `add-role-to-user` können sie einem Benutzer eine bestimmte Rolle hinzufügen. Beispiel:

```
[user@host ~]$ oc policy add-role-to-user role-name username -n project
```

So fügen Sie den Benutzer dev im Projekt wordpress zur Rolle basic-user hinzu:

```
[user@host ~]$ oc policy add-role-to-user basic-user dev -n wordpress
```

Benutzertypen

Die Interaktion mit OpenShift Container Platform wird einem Benutzer zugeordnet. An OpenShift Container Platform user object represents a user who can be granted permissions in the system by adding roles to that user or to a user's group via rolebindings.

Reguläre Benutzer

Die meisten interaktiven Benutzer von OpenShift Container Platform sind regelmäßige Benutzer, repräsentiert durch das Objekt User. Dieser Benutzertyp repräsentiert eine Person, welcher der Zugriff auf die Plattform gestattet wurde. Beispiele für reguläre Benutzer sind user1 und user2.

Systembenutzer

Viele Systembenutzer werden automatisch erstellt, wenn die Infrastruktur definiert wird. Dies dient hauptsächlich dem Zweck, eine sichere Interaktion zwischen der Infrastruktur und der API zu ermöglichen. Zu den Systembenutzern gehören ein Cluster-Administrator

Kapitel 3 | Konfigurieren von Autorisierung und Authentifizierung

(mit Vollzugriff), ein Benutzer pro Knoten, Benutzer für durch Router und Registries usw. Zudem gibt es einen anonymen Systembenutzer, der standardmäßig für nicht authentifizierte Anfragen verwendet wird. Beispiele für Systembenutzer sind: `system:admin`, `system:openshift-registry` und `system:node:node1.example.com`.

Servicekonten

Dies sind spezielle Systembenutzer, die Projekten zugeordnet sind. Einige Servicekonto-Benutzer werden beim ersten Erstellen des Projekts automatisch erstellt. Projektadministratoren können weitere erstellen, um den Zugriff auf die Inhalte der einzelnen Projekte zu definieren. Servicekonten werden häufig verwendet, um Pods oder Bereitstellungen zusätzliche Berechtigungen zu erteilen. Servicekonten werden durch das `ServiceAccount`-Objekt repräsentiert. Beispiele für Servicekontenbenutzer sind `system:serviceaccount:default:deployer` und `system:serviceaccount:foo:builder`.

Jeder Benutzer muss sich authentifizieren, bevor er auf OpenShift Container Platform zugreifen kann. API-Anforderungen ohne Authentifizierung bzw. mit ungültiger Authentifizierung werden als Anfragen durch den anonymen Systembenutzer authentifiziert. Nach erfolgreicher Authentifizierung legt die Richtlinie fest, zu welchen Aktionen der Benutzer befugt ist.



Literaturhinweise

Weitere Informationen zu Kubernetes-Namespace finden Sie in der **Kubernetes-Dokumentation**

<https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/>

Weitere Informationen über RBAC finden Sie im Kapitel *Using RBAC to define and apply permissions* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Authentication and authorization* unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/authentication_and_authorization/index#using-rbac

Weitere Informationen über Gruppen finden Sie im Kapitel *Understanding authentication* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Authentication and authorization* unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/authentication_and_authorization/index#understanding-authentication

► Angeleitete Übung

Definieren und Anwenden von Berechtigungen mit RBAC

In dieser Übung definieren Sie rollenbasierte Zugriffskontrollen und wenden Berechtigungen für Benutzer an.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Entfernen der Berechtigungen für die Projekterstellung von Benutzern, die keine OpenShift-Cluster-Administratoren sind.
- Erstellen von OpenShift-Gruppen und Hinzufügen von Mitgliedern zu diesen Gruppen.
- Erstellen eines Projekts, und Zuweisen von Projektadministratorberechtigungen zum Projekt.
- Zuweisen von Lese- und Schreibberechtigungen zu verschiedenen Benutzergruppen als Projektadministrator.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt einige HTPasswd-Benutzer für die Übung.

```
[student@workstation ~]$ lab auth-rbac start
```

Anweisungen

- 1. Melden Sie sich beim OpenShift-Cluster an, und ermitteln Sie, welche Cluster-Rollenbindungen die Cluster-Rolle `self-provisioner` zuweisen.

- 1.1. Melden Sie sich als Benutzer `admin` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat \
>   https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Listen Sie alle Cluster-Rollenbindungen auf, die auf die Cluster-Rolle `self-provisioner` verweisen.

```
[student@workstation ~]$ oc get clusterrolebinding -o wide \
>   | grep -E 'NAME|self-provisioners'
NAME                  ROLE          ...
self-provisioners    ...  ClusterRole/self-provisioner  ...
```

- 2. Entfernen Sie die Berechtigung zum Erstellen neuer Projekte von allen Benutzern, die keine Cluster-Administratoren sind, indem Sie die Cluster-Rolle `self-provisioner` aus der virtuellen Gruppe `system:authenticated:oauth` löschen.
- 2.1. Vergewissern Sie sich, dass die Cluster-Rollenbindung von `self-provisioners` aus dem vorherigen Schritt die Cluster-Rolle `self-provisioner` zur Gruppe `system:authenticated:oauth` zuweist.

```
[student@workstation ~]$ oc describe clusterrolebindings self-provisioners
Name:          self-provisioners
Labels:        <none>
Annotations:  rbac.authorization.kubernetes.io/autoupdate: true
Role:
  Kind:  ClusterRole
  Name:  self-provisioner
Subjects:
  Kind  Name           Namespace
  ----  --             -----
  Group system:authenticated:oauth
```

- 2.2. Entfernen Sie die Cluster-Rolle `self-provisioner` aus der virtuellen Gruppe `system:authenticated:oauth`, was die Rollenbindung `self-provisioners` löscht. Sie können die Warnung, dass Ihre Änderungen verloren gehen, bedenkenlos ignorieren.

```
[student@workstation ~]$ oc adm policy remove-cluster-role-from-group \
>   self-provisioner system:authenticated:oauth
Warning: Your changes may get lost whenever a master is restarted,
unless you prevent reconciliation of this rolebinding using the
following command: oc annotate clusterrolebinding.rbac self-provisioner
'rbac.authorization.kubernetes.io/autoupdate+++=+++false' --overwrite
clusterrole.rbac.authorization.k8s.io/self-provisioner removed:
"system:authenticated:oauth"
```

- 2.3. Überprüfen Sie, ob die Rolle aus der Gruppe entfernt wurde. Die Cluster-Rollenbindung `self-provisioners` sollte nicht vorhanden sein.

```
[student@workstation ~]$ oc describe clusterrolebindings self-provisioners
Error from server (NotFound): clusterrolebindings.rbac.authorization.k8s.io "self-
provisioners" not found
```

- 2.4. Legen Sie fest, ob andere Cluster-Rollenbindungen auf die Cluster-Rolle `self-provisioners` verweisen.

```
[student@workstation ~]$ oc get clusterrolebinding -o wide \
>   | grep -E 'NAME|self-provisioner'
NAME          ROLE      ...

```

- 2.5. Melden Sie sich mit dem Benutzer `leader` und dem Passwort `redhat` an und versuchen Sie, ein Projekt zu erstellen. Die Projekterstellung sollte fehlschlagen.

```
[student@workstation ~]$ oc login -u leader -p redhat
Login successful.
...output omitted...
[student@workstation ~]$ oc new-project test
Error from server (Forbidden): You may not request a new project via this API.
```

- 3. Erstellen Sie ein Projekt, und weisen Sie dem Benutzer `leader` Projektadministratorberechtigungen zu.
- 3.1. Melden Sie sich als der Benutzer `admin` an, und erstellen Sie das Projekt `auth-rbac`.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
[student@workstation ~]$ oc new-project auth-rbac
Now using project "auth-rbac" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

- 3.2. Erteilen Sie dem Benutzer `leader` Projektadministratorberechtigungen für das Projekt `auth-rbac`.

```
[student@workstation ~]$ oc policy add-role-to-user admin leader
clusterrole.rbac.authorization.k8s.io/admin added: "leader"
```

- 4. Erstellen Sie die Gruppen `dev-group` und `qa-group`, und fügen Sie die jeweiligen Mitglieder hinzu.

- 4.1. Erstellen Sie eine Gruppe mit dem Namen `dev-group`.

```
[student@workstation ~]$ oc adm groups new dev-group
group.user.openshift.io/dev-group created
```

- 4.2. Fügen Sie den Benutzer `developer` zu `dev-group` hinzu.

```
[student@workstation ~]$ oc adm groups add-users dev-group developer
group.user.openshift.io/dev-group added: "developer"
```

- 4.3. Erstellen Sie eine zweite Gruppe mit dem Namen `qa-group`.

```
[student@workstation ~]$ oc adm groups new qa-group
group.user.openshift.io/qa-group created
```

- 4.4. Fügen Sie den Benutzer `qa-engineer` zu `qa-group` hinzu.

Kapitel 3 | Konfigurieren von Autorisierung und Authentifizierung

```
[student@workstation ~]$ oc adm groups add-users qa-group qa-engineer
group.user.openshift.io/qa-group added: "qa-engineer"
```

- 4.5. Überprüfen Sie alle vorhandenen OpenShift-Gruppen, um sicherzustellen, dass sie die richtigen Mitglieder haben.

```
[student@workstation ~]$ oc get groups
NAME      USERS
dev-group developer
qa-group   qa-engineer
```

- 5. Weisen Sie als Benutzer **leader** im Projekt **auth-rbac** der Gruppe **dev-group** Schreibrechte und der Gruppe **qa-group** Leserechte zu.

- 5.1. Melden Sie sich als Benutzer **leader** an.

```
[student@workstation ~]$ oc login -u leader -p redhat
Login successful.
...output omitted...
Using project "auth-rbac".
```

- 5.2. Fügen Sie im Projekt **auth-rbac** der Gruppe **dev-group** Schreibrechte hinzu.

```
[student@workstation ~]$ oc policy add-role-to-group edit dev-group
clusterrole.rbac.authorization.k8s.io/edit added: "dev-group"
```

- 5.3. Fügen Sie im Projekt **auth-rbac** der Gruppe **qa-group** Leserechte hinzu.

```
[student@workstation ~]$ oc policy add-role-to-group view qa-group
clusterrole.rbac.authorization.k8s.io/view added: "qa-group"
```

- 5.4. Überprüfen Sie alle Rollenbindungen im Projekt **auth-rbac**, um sicherzustellen, dass die Rollen zu den richtigen Gruppen und Benutzern zugewiesen werden. In der folgenden Ausgabe sind die von OpenShift zugewiesenen Standard-Rollenbindungen für Servicekonten nicht enthalten.

```
[student@workstation ~]$ oc get rolebindings -o wide
NAME      ROLE          AGE     USERS      GROUPS      ...
admin     ClusterRole/admin  58s    admin       ...
admin-0   ClusterRole/admin  51s    leader      ...
edit      ClusterRole/edit   12s    ...         dev-group
...output omitted...
view      ClusterRole/view   8s    ...         qa-group
```

- 6. Stellen Sie als Benutzer **developer** einen Apache-HTTP-Server bereit, um zu beweisen, dass der Benutzer **developer** über Schreibberechtigungen im Projekt verfügt. Versuchen Sie außerdem, dem Benutzer **qa-engineer** Schreibrechte zu gewähren, um zu beweisen, dass der Benutzer **developer** über keine Projektadministratorberechtigungen verfügt.

- 6.1. Melden Sie sich als Benutzer **developer** an.

```
[student@workstation ~]$ oc login -u developer -p developer
Login successful.
...output omitted...
Using project "auth-rbac".
```

- 6.2. Stellen Sie mithilfe des Standard-Image-Streams von OpenShift einen Apache-HTTP-Server bereit.

```
[student@workstation ~]$ oc new-app --name httpd httpd:2.4
...output omitted...
--> Creating resources ...
  imagestreamtag.image.openshift.io "httpd:2.4" created
  deployment.apps "httpd" created
  service "httpd" created
--> Success
...output omitted...
```

- 6.3. Versuchen Sie, dem Benutzer qa-engineer Schreibrechte zu gewähren. Dies sollte fehlschlagen.

```
[student@workstation ~]$ oc policy add-role-to-user edit qa-engineer
Error from server (Forbidden): rolebindings.rbac.authorization.k8s.io is
forbidden: User "developer" cannot list resource "rolebindings" in API group
"rbac.authorization.k8s.io" in the namespace "auth-rbac"
```

- 7. Überprüfen Sie, ob der Benutzer qa-engineer nur über Leseberechtigungen für die httpd-Anwendung verfügt.

- 7.1. Melden Sie sich als Benutzer qa-engineer an.

```
[student@workstation ~]$ oc login -u qa-engineer -p redhat
Login successful.
...output omitted...
Using project "auth-rbac".
```

- 7.2. Versuchen Sie, die Anwendung httpd zu skalieren. Dies sollte fehlschlagen.

```
[student@workstation ~]$ oc scale deployment httpd --replicas 3
Error from server (Forbidden): deployments.apps "httpd" is forbidden: User "qa-
engineer" cannot patch resource "deployments/scale" in API group "apps" in the
namespace "auth-rbac"
```

- 8. Stellen Sie die Berechtigungen für die Projekterstellung für alle Benutzer wieder her.

- 8.1. Melden Sie sich als Benutzer admin an.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
```

- 8.2. Stellen Sie die Berechtigungen für die Projekterstellung für alle Benutzer wieder her, indem Sie die vom OpenShift-Installationsprogramm erstellte Cluster-Rollenbindung `self-provisioners` neu erstellen. Sie können die Warnung, dass die Gruppe nicht gefunden wurde, bedenkenlos ignorieren.

```
[student@workstation ~]$ oc adm policy add-cluster-role-to-group \
>   --rolebinding-name self-provisioners \
>   self-provisioner system:authenticated:oauth
Warning: Group 'system:authenticated:oauth' not found
clusterrole.rbac.authorization.k8s.io/self-provisioner added:
"system:authenticated:oauth"
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab auth-rbac finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Überprüfen der Integrität eines Clusters

Leistungscheckliste

In dieser praktischen Übung konfigurieren Sie den HTPasswd-Identitätsanbieter, erstellen Gruppen und weisen Rollen zu Benutzern und Gruppen zu.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen von Benutzern und Passwörtern für die HTPasswd-Authentifizierung.
- Konfigurieren des HTPasswd-Identitätsanbieters für die HTPasswd-Authentifizierung.
- Zuweisen von Cluster-Administratorrechten für Benutzer.
- Entfernen der Möglichkeit, Projekte auf Cluster-Ebene zu erstellen.
- Erstellen von Gruppen und Hinzufügen von Mitgliedern zu Gruppen.
- Verwalten von Benutzerberechtigungen in Projekten durch die Gewährung von Berechtigungen für Gruppen.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

```
[student@workstation ~]$ lab auth-review start
```

Mit dem Befehl wird sichergestellt, dass die Cluster-API erreichbar ist, das Paket `httpd-util` installiert ist und die Authentifizierungseinstellungen auf die Standardeinstellungen für die Installation konfiguriert sind.

Anweisungen

1. Aktualisieren Sie die vorhandene HTPasswd-Authentifizierungsdatei `~/D0280/labs/auth-review/tmp_users`, um den Benutzer `analyst` zu entfernen. Stellen Sie sicher, dass die Benutzer `tester` und `leader` in der Datei das Passwort `L@bR3v!ew` verwenden. Fügen Sie der Datei zwei neue Einträge für die Benutzer `admin` und `developer` hinzu. Verwenden Sie `L@bR3v!ew` als Passwort für die neuen Benutzer.
2. Melden Sie sich bei Ihrem OpenShift-Cluster als Benutzer `kubeadmin` an und verwenden Sie die in der Datei `/usr/local/etc/ocp4.config` definierte Variable `RHT_OCP4_KUBEADM_PASSWD` als Passwort. Konfigurieren Sie Ihren Cluster so, dass er den HTPasswd-Identitätsanbieter mit den Benutzernamen und Passwörtern aus der Datei `~/D0280/labs/auth-review/tmp_users` verwendet.
3. Konfigurieren Sie den Benutzer `admin` als Cluster-Administrator. Melden Sie sich als `admin` und als `developer` an, um die HTPasswd-Benutzerkonfiguration und die Cluster-Berechtigungen zu verifizieren.

Kapitel 3 | Konfigurieren von Autorisierung und Authentifizierung

4. Melden Sie sich als Benutzer `admin` an, und entfernen Sie die Möglichkeit, Cluster-weite Projekte zu erstellen.
5. Erstellen Sie eine Gruppe mit dem Namen `managers`, und fügen Sie den Benutzer `leader` zur Gruppe hinzu. Erteilen Sie der Gruppe `managers` die Berechtigung zum Erstellen von Projekten. Erstellen Sie als Benutzer `leader` das Projekt `auth-review`.
6. Erstellen Sie eine Gruppe mit dem Namen `developers`, und gewähren Sie Bearbeitungsberechtigungen für das Projekt `auth-review`. Fügen Sie der Gruppe den Benutzer `developer` hinzu.
7. Erstellen Sie eine Gruppe mit dem Namen `qa`, und gewähren Sie Anzeigeberechtigungen für das Projekt `auth-review`. Fügen Sie den Benutzer `tester` zur Gruppe hinzu.

Bewertung

Führen Sie auf der **Workstation** den Befehl `lab` aus, um Ihre Arbeit zu bewerten. Beheben Sie sämtliche gemeldeten Fehler, und führen Sie das Skript so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab auth-review grade
```

Beenden

Führen Sie auf dem Rechner **workstation** den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab auth-review finish
```

Hiermit ist die praktische Übung beendet.

► Lösung

Überprüfen der Integrität eines Clusters

Leistungscheckliste

In dieser praktischen Übung konfigurieren Sie den HTPasswd-Identitätsanbieter, erstellen Gruppen und weisen Rollen zu Benutzern und Gruppen zu.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen von Benutzern und Passwörtern für die HTPasswd-Authentifizierung.
- Konfigurieren des HTPasswd-Identitätsanbieters für die HTPasswd-Authentifizierung.
- Zuweisen von Cluster-Administratorrechten für Benutzer.
- Entfernen der Möglichkeit, Projekte auf Cluster-Ebene zu erstellen.
- Erstellen von Gruppen und Hinzufügen von Mitgliedern zu Gruppen.
- Verwalten von Benutzerberechtigungen in Projekten durch die Gewährung von Berechtigungen für Gruppen.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

```
[student@workstation ~]$ lab auth-review start
```

Mit dem Befehl wird sichergestellt, dass die Cluster-API erreichbar ist, das Paket `httpd-util` installiert ist und die Authentifizierungseinstellungen auf die Standardeinstellungen für die Installation konfiguriert sind.

Anweisungen

1. Aktualisieren Sie die vorhandene HTPasswd-Authentifizierungsdatei `~/D0280/labs/auth-review/tmp_users`, um den Benutzer `analyst` zu entfernen. Stellen Sie sicher, dass die Benutzer `tester` und `leader` in der Datei das Passwort `L@bR3v!ew` verwenden. Fügen Sie der Datei zwei neue Einträge für die Benutzer `admin` und `developer` hinzu. Verwenden Sie `L@bR3v!ew` als Passwort für die neuen Benutzer.
 - 1.1. Entfernen Sie den Benutzer `analyst` aus der HTPasswd-Authentifizierungsdatei `~/D0280/labs/auth-review/tmp_users`.

```
[student@workstation ~]$ htpasswd -D ~/D0280/labs/auth-review/tmp_users analyst
Deleting password for user analyst
```

Kapitel 3 | Konfigurieren von Autorisierung und Authentifizierung

- 1.2. Aktualisieren Sie die Einträge für die Benutzer `tester` und `leader`, so dass diese das Passwort `L@bR3v!ew` verwenden. Fügen Sie Einträge für die Benutzer `admin` und `developer` mit dem Passwort `L@bR3v!ew` hinzu.

```
[student@workstation ~]$ for NAME in tester leader admin developer
>   do
>     htpasswd -b ~/D0280/labs/auth-review/tmp_users ${NAME} 'L@bR3v!ew'
>   done
Updating password for user tester
Updating password for user leader
Adding password for user admin
Adding password for user developer
```

- 1.3. Überprüfen Sie den Inhalt der Datei `~/D0280/labs/auth-review/tmp_users`. Sie enthält keine Zeile für den Benutzer `analyst`. Sie enthält jedoch zwei neue Einträge mit gehaschten Passwörtern für die Benutzer `admin` und `developer`.

```
[student@workstation ~]$ cat ~/D0280/labs/auth-review/tmp_users
tester:$apr1$0eqhKgbU$DWd0CB4IumhasaRuEr6hp0
leader:$apr1$.EB5IXlu$FDV.Av16njl0CMzgolScr/
admin:$apr1$ItcCncDS$xFQCUjQGTsXAup00KQfmw0
developer:$apr1$D8F1Hren$izDhAwq5DRjUHPv0i7FHn.
```

2. Melden Sie sich bei Ihrem OpenShift-Cluster als Benutzer `kubeadmin` an und verwenden Sie die in der Datei `/usr/local/etc/ocp4.config` definierte Variable `RHT_OCP4_KUBEADM_PASSWD` als Passwort. Konfigurieren Sie Ihren Cluster so, dass er den HTPasswd-Identitätsanbieter mit den Benutzernamen und Passwörtern aus der Datei `~/D0280/labs/auth-review/tmp_users` verwendet.
- 2.1. Rufen Sie die Kursumgebungs-Konfigurationsdatei auf, auf die unter `/usr/local/etc/ocp4.config` zugegriffen werden kann.

```
[student@workstation ~]$ source /usr/local/etc/ocp4.config
```

- 2.2. Melden Sie sich als Benutzer `kubeadmin` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u kubeadmin -p ${RHT_OCP4_KUBEADM_PASSWD} \
>   https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 2.3. Erstellen Sie mit der Datei `~/D0280/labs/auth-review/tmp_users` ein Secret namens `auth-review`.

```
[student@workstation ~]$ oc create secret generic auth-review \
>   --from-file htpasswd=/home/student/D0280/labs/auth-review/tmp_users \
>   -n openshift-config
secret/auth-review created
```

- 2.4. Exportieren Sie die vorhandene OAuth-Ressource nach `~/D0280/labs/auth-review/oauth.yaml`.

```
[student@workstation ~]$ oc get oauth cluster \
> -o yaml > ~/D0280/labs/auth-review/oauth.yaml
```

- 2.5. Bearbeiten Sie die Datei ~/D0280/labs/auth-review/oauth.yaml, und ersetzen Sie die Zeile spec: {} durch die folgenden fett gedruckten Zeilen. Beachten Sie, dass sich htpasswd, mappingMethod, name und type auf derselben Einrückungsebene befinden.

```
apiVersion: config.openshift.io/v1
kind: OAuth
...output omitted...
spec:
  identityProviders:
    - htpasswd:
        fileData:
          name: auth-review
      mappingMethod: claim
      name: htpasswd
      type: HTPasswd
```



Anmerkung

Zur Vereinfachung enthält die Datei ~/D0280/solutions/auth-review/oauth.yaml eine Minimalversion der OAuth-Konfiguration mit den angegebenen Anpassungen.

- 2.6. Wenden Sie die im vorherigen Schritt definierte benutzerdefinierte Ressource an.

```
[student@workstation ~]$ oc replace -f ~/D0280/labs/auth-review/oauth.yaml
oauth.config.openshift.io/cluster replaced
```

- 2.7. Bei einer erfolgreichen Aktualisierung der oauth/cluster-Ressource werden die oauth-openshift-Pods im openshift-authentication-Namespace neu erstellt.

```
[student@workstation ~]$ watch oc get pods -n openshift-authentication
```

Warten Sie, bis die neuen oauth-openshift-Pods bereit sind und ausgeführt werden und die vorherigen Pods beendet wurden.

NAME	READY	STATUS	RESTARTS	AGE
oauth-openshift-6755d8795-h8bgv	1/1	Running	0	34s
oauth-openshift-6755d8795-rk4m6	1/1	Running	0	38s

Drücken Sie Strg+C, um den watch-Befehl zu beenden.

3. Konfigurieren Sie den Benutzer admin als Cluster-Administrator. Melden Sie sich als admin und als developer an, um die HTPasswd-Benutzerkonfiguration und die Cluster-Berechtigungen zu verifizieren.

Kapitel 3 | Konfigurieren von Autorisierung und Authentifizierung

- 3.1. Weisen Sie dem Benutzer `admin` die Rolle `cluster-admin` zu.

```
[student@workstation ~]$ oc adm policy add-cluster-role-to-user \
>   cluster-admin admin
clusterrole.rbac.authorization.k8s.io/cluster-admin added: "admin"
```



Anmerkung

Die Ausgabe weist darauf hin, dass der Benutzer `admin` nicht gefunden wurde und kann problemlos ignoriert werden.

- 3.2. Melden Sie sich als Benutzer `admin` beim Cluster an, um zu überprüfen, ob die HTPasswd-Authentifizierung korrekt konfiguriert wurde.

```
[student@workstation ~]$ oc login -u admin -p 'L@bR3v!ew'
Login successful.
...output omitted...
```

- 3.3. Verwenden Sie den Befehl `oc get nodes`, um sicherzustellen, dass der Benutzer `admin` über die Rolle `cluster-admin` verfügt. Die Namen der Knoten aus Ihrem Cluster sind möglicherweise anders.

```
[student@workstation ~]$ oc get nodes
NAME      STATUS    ROLES      AGE     VERSION
master01   Ready     master,worker  46d    v1.19.0+d856161
master02   Ready     master,worker  46d    v1.19.0+d856161
master03   Ready     master,worker  46d    v1.19.0+d856161
```

- 3.4. Melden Sie sich als Benutzer `developer` beim Cluster an, um zu überprüfen, ob die HTPasswd-Authentifizierung ordnungsgemäß konfiguriert ist.

```
[student@workstation ~]$ oc login -u developer -p 'L@bR3v!ew'
Login successful.
...output omitted...
```

- 3.5. Verwenden Sie den Befehl `oc get nodes`, um sicherzustellen, dass der Benutzer `developer` keine Cluster-Administratorberechtigungen hat.

```
[student@workstation ~]$ oc get nodes
Error from server (Forbidden): nodes is forbidden: User "developer" cannot list
resource "nodes" in API group "" at the cluster scope
```

4. Melden Sie sich als Benutzer `admin` an, und entfernen Sie die Möglichkeit, Cluster-weite Projekte zu erstellen.

- 4.1. Melden Sie sich als Benutzer `admin` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u admin -p 'L@bR3v!ew'
Login successful.
...output omitted...
```

- 4.2. Entfernen Sie die Cluster-Rolle `self-provisioner` aus der virtuellen Gruppe `system:authenticated:oauth`.

```
[student@workstation ~]$ oc adm policy remove-cluster-role-from-group \
>   self-provisioner system:authenticated:oauth
clusterrole.rbac.authorization.k8s.io/self-provisioner removed:
"system:authenticated:oauth"
```



Anmerkung

Sie können die Warnung, dass Ihre Änderungen verloren gehen, bedenkenlos ignorieren.

5. Erstellen Sie eine Gruppe mit dem Namen `managers`, und fügen Sie den Benutzer `leader` zur Gruppe hinzu. Erteilen Sie der Gruppe `managers` die Berechtigung zum Erstellen von Projekten. Erstellen Sie als Benutzer `leader` das Projekt `auth-review`.

- 5.1. Erstellen Sie eine Gruppe mit dem Namen `managers`.

```
[student@workstation ~]$ oc adm groups new managers
group.user.openshift.io/managers created
```

- 5.2. Fügen Sie den Benutzer `leader` zur Gruppe `managers` hinzu.

```
[student@workstation ~]$ oc adm groups add-users managers leader
group.user.openshift.io/managers added: "leader"
```

- 5.3. Weisen Sie die Cluster-Rolle `self-provisioner` zur Gruppe `managers` zu.

```
[student@workstation ~]$ oc adm policy add-cluster-role-to-group \
>   self-provisioner managers
clusterrole.rbac.authorization.k8s.io/self-provisioner added: "managers"
```

- 5.4. Erstellen Sie als Benutzer `leader` das Projekt `auth-review`.

```
[student@workstation ~]$ oc login -u leader -p 'L@bR3v!ew'
Login successful.
...output omitted...
```

Der Benutzer, der ein Projekt erstellt, erhält automatisch die Rolle `admin` für das Projekt.

```
[student@workstation ~]$ oc new-project auth-review
Now using project "auth-review" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

6. Erstellen Sie eine Gruppe mit dem Namen `developers`, und gewähren Sie Bearbeitungsberechtigungen für das Projekt `auth-review`. Fügen Sie der Gruppe den Benutzer `developer` hinzu.

- 6.1. Melden Sie sich als Benutzer `admin` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u admin -p 'L@bR3v!ew'  
Login successful.  
...output omitted...
```

6.2. Erstellen Sie eine Gruppe mit dem Namen **developers**.

```
[student@workstation ~]$ oc adm groups new developers  
group.user.openshift.io/developers created
```

6.3. Fügen Sie den Benutzer **developer** zur Gruppe **developers** hinzu.

```
[student@workstation ~]$ oc adm groups add-users developers developer  
group.user.openshift.io/developers added: "developer"
```

6.4. Erteilen Sie der Gruppe **developers** Bearbeitungsberechtigungen für das Projekt **auth-review**.

```
[student@workstation ~]$ oc policy add-role-to-group edit developers  
clusterrole.rbac.authorization.k8s.io/edit added: "developers"
```

7. Erstellen Sie eine Gruppe mit dem Namen **qa**, und gewähren Sie Anzeigeberechtigungen für das Projekt **auth-review**. Fügen Sie den Benutzer **tester** zur Gruppe hinzu.

7.1. Erstellen Sie eine Gruppe mit dem Namen **qa**.

```
[student@workstation ~]$ oc adm groups new qa  
group.user.openshift.io/qa created
```

7.2. Fügen Sie den Benutzer **tester** zur Gruppe **qa** hinzu.

```
[student@workstation ~]$ oc adm groups add-users qa tester  
group.user.openshift.io/qa added: "tester"
```

7.3. Erteilen Sie der Gruppe **qa** Anzeigeberechtigungen für das Projekt **auth-review**.

```
[student@workstation ~]$ oc policy add-role-to-group view qa  
clusterrole.rbac.authorization.k8s.io/view added: "qa"
```

Bewertung

Führen Sie auf der Workstation den Befehl **lab** aus, um Ihre Arbeit zu bewerten. Beheben Sie sämtliche gemeldeten Fehler, und führen Sie das Skript so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab auth-review grade
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab auth-review finish
```

Hiermit ist die praktische Übung beendet.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Ein frisch installierter OpenShift-Cluster bietet zwei Authentifizierungsmethoden, die administrativen Zugriff gewähren: die Datei `kubeconfig` und den virtuellen Benutzer `kubeadmin`.
- Der HTPasswd-Identitätsanbieter authentifiziert Benutzer anhand der in einem Secret gespeicherten Anmeldedaten. Der Name des Secret und andere Einstellungen für den Identitätsanbieter werden in der benutzerdefinierten OAuth-Ressource gespeichert.
- Um Benutzeranmeldedaten mit dem HTPasswd-Identitätsanbieter zu verwalten, müssen Sie die Daten aus dem Secret extrahieren, sie mit dem Befehl `HTPasswd` ändern und anschließend wieder auf das Secret anwenden.
- Das Erstellen von OpenShift-Benutzern erfordert gültige Anmeldedaten, die von einem Identitätsanbieter verwaltet werden, sowie Benutzer- und Identitätsressourcen.
- Um OpenShift-Benutzer zu löschen, müssen Sie Ihre Anmeldedaten vom Identitätsanbieter sowie deren Benutzer- und Identitätsressourcen löschen.
- OpenShift verwendet die rollenbasierte Zugriffskontrolle (RBAC), um Benutzeraktionen zu steuern. Eine Rolle ist eine Sammlung von Regeln, welche die Interaktion mit OpenShift-Ressourcen steuern. Standardrollen sind für Cluster-Administratoren, Entwickler und Auditoren vorhanden.
- Um die Benutzerinteraktion zu steuern, weisen Sie einem Benutzer eine oder mehr Rollen zu. Eine Rollenbindung enthält alle Zuordnungen einer Rolle zu Benutzern und Gruppen.
- Weisen Sie dem Benutzer die Rolle `cluster-admin` zu, um einem Benutzer Cluster-Administratorberechtigungen zu erteilen.

Kapitel 4

Konfigurieren der Anwendungssicherheit

Ziel

Beschränken der Berechtigungen von Anwendungen mithilfe von Sicherheitskontextbeschränkungen und Schützen der Zugangsdaten mit Secrets

Ziele

- Erstellen und Anwenden von Geheimnissen zum Verwalten von vertraulichen Informationen und Teilen von Geheimnissen zwischen Anwendungen
- Erstellen von Servicekonten und Anwenden von Berechtigungen sowie Verwalten von Sicherheitskontextbeschränkungen

Abschnitte

- Verwalten von vertraulichen Informationen mit Secrets (und angeleitete Übung)
- Kontrollieren von Anwendungsberechtigungen mit Sicherheitskontextbeschränkungen (und angeleitete Übung)

Praktische Übung

Konfigurieren der Anwendungssicherheit

Verwalten von vertraulichen Informationen mit Secrets

Ziele

Am Ende dieses Abschnitts sollten Sie in der Lage sein, Geheimnisse zu erstellen und anzuwenden, um vertrauliche Informationen zu verwalten, und Geheimnisse zwischen Anwendungen zu teilen.

Secrets – Übersicht

Moderne Anwendungen sind so konzipiert, dass sie Code, Konfiguration und Daten lose miteinander verbinden. Konfigurationsdateien und -daten sind als Teil der Software nicht hartcodiert. Stattdessen lädt die Software die Konfiguration und die Daten aus einer externen Quelle. Dadurch wird die Bereitstellung einer Anwendung in unterschiedlichen Umgebungen ermöglicht, ohne dass der Anwendungsquellcode geändert werden muss.

Häufig benötigen Anwendungen Zugriff auf vertrauliche Informationen. Beispiel: Eine Back-End-Webanwendung benötigt Zugriff auf Datenbank-Anmelddaten, um Datenbankabfragen auszuführen. Kubernetes und OpenShift verwenden Secret-Ressourcen, um vertrauliche Informationen zu speichern, z. B.:

- Passwörter.
- Vertrauliche Konfigurationsdateien
- Passwörter für eine externe Ressource, z. B. ein SSH-Schlüssel oder OAuth-Token

Ein Secret kann beliebige Datentypen speichern. Die Daten in einem Secret sind Base64-codiert und werden nicht im Klartext gespeichert. Secret-Daten sind nicht verschlüsselt. Sie können das Secret vom Base64-Format decodieren, um auf die ursprünglichen Daten zuzugreifen.

Obwohl Secrets jede Art von Daten speichern können, unterstützen Kubernetes und OpenShift verschiedene Arten von Secrets. Es gibt verschiedene Arten von Secret-Ressourcen, einschließlich Dienstkonto-Token, SSH-Schlüssel und TLS-Zertifikate. Wenn Sie Informationen in einem bestimmten Secret-Ressourcentyp speichern, überprüft Kubernetes, ob die Daten dem Secret-Typ entsprechen.



Anmerkung

Sie können die Etcd-Datenbank verschlüsseln, auch wenn dies nicht die Standardeinstellung ist. Bei Aktivierung verschlüsselt Etcd die folgenden Ressourcen: Secrets, Konfigurations-Maps, Routen, OAuth-Zugriffstoken und OAuth-Autorisierungstoken. Die Aktivierung der Etcd-Verschlüsselung wird in diesem Kurs nicht behandelt.

Funktionen von Secrets

Zu den wichtigsten Funktionen von Secrets gehören:

- Secret-Daten können innerhalb eines Namespace freigegeben werden.

- Secret-Daten können unabhängig von ihrer Definition referenziert werden. Administratoren können eine Secret-Ressource erstellen und verwalten, und andere Teammitglieder verweisen in ihren Bereitstellungskonfigurationen auf das Secret.
- Secret-Daten werden in Pods injiziert, wenn OpenShift einen Pod erstellt. Sie können ein Secret als Umgebungsvariable oder als gemountete Datei im Pod verfügbar machen.
- Wenn sich der Wert eines Secrets während der Ausführung des Pods ändert, werden die Secret-Daten im Pod nicht aktualisiert. Nachdem ein Secret-Wert geändert wurde, müssen Sie neue Pods erstellen, um die neuen Secret-Daten einzufügen.
- Alle Secret-Daten, die OpenShift in einen Pod injiziert, sind kurzlebig. Wenn OpenShift vertrauliche Daten für einen Pod als Umgebungsvariablen bereitstellt, werden diese Variablen zerstört, wenn der Pod zerstört wird.

Secret-Daten-Volumes werden mittels temporärer Dateispeicherung gesichert. Wenn ein Secret als Datei im Pod gemountet wird, wird die Datei ebenfalls zerstört, wenn der Pod zerstört wird. Beendete Pods enthalten keine Secret-Daten.

Anwendungsfälle für Secrets

Zwei Hauptanwendungsfälle für Secrets sind das Speichern von Anmelddaten und die Sicherung der Kommunikation zwischen Services.

Anmelddaten

Speichern Sie vertrauliche Informationen, wie Passwörter und Benutzernamen, in einem Secret.

Wenn eine Anwendung vertrauliche Informationen aus einer Datei liest, wird das Secret als Daten-Volume an den Pod gemountet. Die Anwendung kann das Secret als normale Datei für den Zugriff auf die vertrauliche Informationen lesen. Einige Datenbanken lesen beispielsweise die Anmelddaten aus einer Datei, um Benutzer zu authentifizieren.

Einige Anwendungen verwenden Umgebungsvariablen zum Lesen von Konfigurations- und vertraulichen Daten. Sie können Secret-Variablen mit Pod-Umgebungsvariablen in einer Bereitstellungskonfiguration verknüpfen.

Transport Layer Security (TLS) und Schlüsselpaare

Verwenden Sie ein TLS-Zertifikat und einen Schlüssel, um die Kommunikation mit einem Pod zu sichern. Ein TLS-Geheimnis speichert das Zertifikat als `tls.crt` und den Zertifikatschlüssel als `tls.key`. Entwickler können das Secret als Volume mounten und eine Passthrough-Weiterleitung zur Anwendung erstellen.

Erstellen von Geheimnissen

Wenn ein Pod Zugriff auf vertrauliche Informationen benötigt, erstellen Sie vor der Bereitstellung des Pods ein Secret für die Informationen. Verwenden Sie einen der folgenden Befehle, um ein Geheimnis zu erstellen:

- Erstellen Sie ein generisches Secret mit Schlüssel-Wert-Paaren aus literalen Werten, die Sie in der Befehlszeile eingeben:

```
[user@host ~]$ oc create secret generic secret_name \
>   --from-literal key1=secret1 \
>   --from-literal key2=secret2
```

Kapitel 4 | Konfigurieren der Anwendungssicherheit

- Erstellen Sie ein generisches Secret mit den in der Befehlszeile angegebenen Schlüsselnamen und den Werten aus den Dateien:

```
[user@host ~]$ oc create secret generic ssh-keys \
>   --from-file id_rsa=/path-to/id_rsa \
>   --from-file id_rsa.pub=/path-to/id_rsa.pub
```

- Erstellen Sie ein TLS-Geheimnis mit einem Zertifikat und dem zugehörigen Schlüssel:

```
[user@host ~]$ oc create secret tls secret-tls \
>   --cert /path-to-certificate --key /path-to-key
```

Bereitstellen von Secrets für Pods

Um einem Pod ein Secret bereitzustellen, erstellen Sie zunächst das Secret. Weisen Sie jedem Teil der vertraulichen Daten einen Schlüssel zu. Nach der Erstellung enthält das Secret Schlüssel-Wert-Paare. Der folgende Befehl erstellt ein generisches Secret mit dem Namen `demo-secret` mit zwei Schlüsseln: `user` mit dem Wert `demo-user` und `root_password` mit dem Wert `zT1KTgk`.

```
[user@host ~]$ oc create secret generic demo-secret \
>   --from-literal user=demo-user
>   --from-literal root_password=zT1KTgk
```

Secrets als Pod-Umgebungsvariablen

Stellen Sie sich eine Datenbankanwendung vor, die das Passwort des Datenbankadministrators aus der Umgebungsvariablen `MYSQL_ROOT_PASSWORD` liest. Modifizieren Sie den Abschnitt mit den Umgebungsvariablen der Bereitstellungskonfiguration so, dass Werte aus dem Secret verwendet werden:

```
env:
  - name: MYSQL_ROOT_PASSWORD ❶
    valueFrom:
      secretKeyRef: ❷
        name: demo-secret ❸
        key: root_password ❹
```

- Der Name der Umgebungsvariablen im Pod, der Daten aus einem Secret enthält.
- Der Schlüssel `secretKeyRef` erwartet ein Secret. Verwenden Sie den Schlüssel `configMapKeyRef` für Konfigurations-Maps.
- Der Name des Secrets, das die gewünschten vertraulichen Informationen enthält.
- Der Name des Schlüssels, der die vertraulichen Informationen im Secret enthält.

Sie können auch den Befehl `oc set env` verwenden, um Umgebungsvariablen in der Anwendung aus Secrets oder Konfigurations-Maps festzulegen: In einigen Fällen können Sie die Namen der Schlüssel mit der Option `--prefix` an die Namen der Umgebungsvariablen angleichen. Im folgenden Beispiel legt der Schlüssel `user` die Umgebungsvariable `MYSQL_USER` fest. Der Schlüssel `root_password` legt die Umgebungsvariable `MYSQL_ROOT_PASSWORD` fest. Wenn ein

Secret-Schlüssel in Kleinbuchstaben geschrieben ist, wird die entsprechende Umgebungsvariable in Großbuchstaben konvertiert.

```
[user@host ~]$ oc set env deployment/demo --from secret/demo-secret \
>   --prefix MYSQL_
```

Secrets als Dateien in einem Pod

Ein Secret kann in einem Verzeichnis in einem Pod gemountet werden. Für jeden Schlüssel im Secret wird eine Datei mit dem Namen des Schlüssels erstellt. Die Dateien enthalten jeweils den decodierten Wert des Secrets. Der folgende Befehl zeigt, wie Geheimnisse in einem Pod gemountet werden:

```
[user@host ~]$ oc set volume deployment/demo \
>   --add --type secret \
>   --secret-name demo-secret \
>   --mount-path /app-secrets
```

- ➊ Ändern Sie die Volume-Konfiguration in der demo-Bereitstellungskonfiguration.
- ➋ Fügen Sie ein neues Volume aus einem Secret hinzu. Konfigurations-Maps können auch als Volumes gemountet werden.
- ➌ Verwenden Sie das Secret demo-secret.
- ➍ Stellen Sie die Secret-Daten im Verzeichnis /app-secrets im Pod zur Verfügung. Der Inhalt der Datei /app-secrets/user ist demo-user. Der Inhalt der Datei /app-secrets/root_password ist zT1KTgk.

Das Container-Image kann den Speicherort des Mount-Punkts und die erwarteten Dateinamen vorgeben. Beispielsweise kann ein Container-Image, auf dem NGINX ausgeführt wird, den Speicherort des SSL-Zertifikats und des SSL-Zertifikatschlüssels in der Konfigurationsdatei /etc/nginx/nginx.conf angeben. Wenn die erwarteten Dateien nicht gefunden werden, kann es sein, dass der Container nicht ausgeführt wird.



Wichtig

Wenn der Mount-Punkt bereits im Pod vorhanden ist, werden alle vorhandenen Dateien am Mount-Punkt durch das bereitgestellte Geheimnis verdeckt. Die vorhandenen Dateien sind nicht sichtbar und können nicht geöffnet werden.

Konfigurations-Maps: Übersicht

Ähnlich wie Secrets dienen Konfigurations-Maps dazu, Konfigurationsinformationen von Container-Images zu entkoppeln. Im Gegensatz zu Secrets müssen die in den Konfigurations-Maps enthaltenen Informationen nicht geschützt werden. Sie können die Daten in einer Konfigurations-Map verwenden, um Umgebungsvariablen im Container-Image festzulegen, oder Sie können die Konfigurations-Map auch als Volume im Container-Image mounten.

Container-Images müssen nicht neu erstellt werden, wenn ein Secret oder eine Konfigurations-Map geändert wird. Neue Pods verwenden die aktualisierten Secrets und Konfigurations-Maps. Pods mit den älteren Secrets und Konfigurations-Maps können gelöscht werden.

Kapitel 4 | Konfigurieren der Anwendungssicherheit

Die Syntax zum Erstellen einer Konfigurations-Map ist der Syntax zum Erstellen von Secrets sehr ähnlich. Sie können Schlüssel-Wert-Paare in der Befehlszeile eingeben oder den Inhalt einer Datei als Wert eines angegebenen Schlüssels verwenden. Der folgende Befehl zeigt, wie eine Konfigurations-Map erstellt wird:

```
[user@host ~]$ oc create configmap my-config \
>   --from-literal key1=config1 --from-literal key2=config2
```

Secrets und Konfigurations-Maps aktualisieren

Secrets und Konfigurations-Maps müssen gelegentlich aktualisiert werden. Verwenden Sie den Befehl `oc extract`, um sicherzustellen, dass Sie die neuesten Daten verwenden. Speichern Sie die Daten mit der Option `--to` in einem bestimmten Verzeichnis. Für jeden Schlüssel im Secret oder in der Konfigurations-Map wird eine Datei mit dem Namen des Schlüssels erstellt. Die Dateien enthalten jeweils den Wert des entsprechenden Schlüssels. Wenn Sie den Befehl `oc extract` mehrmals ausführen, verwenden Sie die Option `--confirm`, um vorhandene Dateien zu überschreiben.

```
[user@host ~]$ oc extract secret/htpasswd-ppk1q -n openshift-config \
>   --to /tmp/ --confirm
```

Führen Sie nach der Aktualisierung der lokal gespeicherten Dateien den Befehl `oc set data` aus, um das Secret bzw. die Konfigurations-Map zu aktualisieren. Geben Sie für jeden Schlüssel, der aktualisiert werden muss, den Namen eines Schlüssels und den zugehörigen Wert an. Wenn sich der Wert in einer Datei befindet, verwenden Sie die Option `--from-file`.

Im vorherigen Beispiel für `oc extract` enthielt das Secret `htpasswd-ppk1q` nur einen Schlüssel mit dem Namen `htpasswd`. Wenn Sie den Befehl `oc set data` verwenden, können Sie den Schlüsselnamen `htpasswd` explizit mit `--from-file htpasswd=/tmp/htpasswd` angeben. Wenn Sie den Schlüsselnamen nicht angeben, wird der Dateiname als Schlüsselname verwendet.

```
[user@host ~]$ oc set data secret/htpasswd-ppk1q -n openshift-config \
>   --from-file /tmp/htpasswd
```



Literaturhinweise

Weitere Informationen zu Geheimnissen finden Sie im Abschnitt *Understanding Secrets* des Kapitels *Working with Pods* in der Dokumentation zu „Red Hat OpenShift Container Platform 4.6 Nodes“ unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/nodes/index#nodes-pods-secrets-about_nodes-pods-secrets

Weitere Informationen zur Etcd-Verschlüsselung finden Sie im Kapitel *Encrypting Etcd data* in der Dokumentation zu „Red Hat OpenShift Container Platform 4.6 Security“ unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/security/index#encrypting-etcd

► Angeleitete Übung

Verwalten von vertraulichen Informationen mit Secrets

In dieser Übung lernen Sie, wie Sie Informationen mit Secrets verwalten können.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Verwalten von Secrets und deren Verwendung zum Initialisieren von Umgebungsvariablen in Anwendungen.
- Verwenden von Secrets für eine MySQL-Datenbankanwendung.
- Zuweisen von Secrets zu einer Anwendung, die eine Verbindung zu einer MySQL-Datenbank herstellt.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Cluster-API erreichbar ist, und lädt die für diese Übung benötigten Ressourcen herunter.

```
[student@workstation ~]$ lab authorization-secrets start
```

Anweisungen

- 1. Melden Sie sich bei dem OpenShift-Cluster an, und erstellen Sie das Projekt `authorization-secrets`.
- 1.1. Melden Sie sich als Benutzer `developer` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \
>   https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Erstellen Sie das Projekt `authorization-secrets`.

```
[student@workstation ~]$ oc new-project authorization-secrets
Now using project "authorization-secrets" on server
"https://api.ocp4.example.com:6443".
...output omitted...
```

- 2. Erstellen Sie ein Secret mit den Anmeldeinformationen und Verbindungsinformationen für den Zugriff auf eine MySQL-Datenbank.

```
[student@workstation ~]$ oc create secret generic mysql \
> --from-literal user=myuser --from-literal password=redhat123 \
> --from-literal database=test_secrets --from-literal hostname=mysql
secret/mysql created
```

- 3. Stellen Sie eine Datenbank bereit, und fügen Sie das Secret für die Benutzer- und Datenbankkonfiguration hinzu.
- 3.1. Versuchen Sie, einen temporären Datenbankserver bereitzustellen. Dies sollte fehlgeschlagen, da das MySQL-Image Umgebungsvariablen für die anfängliche Konfiguration benötigt. Die Werte für diese Variablen können nicht mit dem Befehl `oc new-app` aus einem Secret zugewiesen werden.

```
[student@workstation ~]$ oc new-app --name mysql \
> --docker-image registry.redhat.io/rhel8/mysql-80:1
...output omitted...
--> Creating resources ...
imagestream.image.openshift.io "mysql" created
deployment.apps "mysql" created
service "mysql" created
--> Success
...output omitted...
```

- 3.2. Führen Sie den Befehl `oc get pods` mit der Option `-w` aus, um den Status der Bereitstellung in Echtzeit abzurufen. Beachten Sie, dass sich der Datenbank-Pod in einem fehlgeschlagenen Status befindet. Drücken Sie Strg+C, um den Befehl zu verlassen.

NAME	READY	STATUS	RESTARTS	AGE
mysql-786bb947f9-qz2fm	0/1	Error	3	71s
mysql-786bb947f9-qz2fm	0/1	CrashLoopBackOff	3	75s
mysql-786bb947f9-qz2fm	0/1	Error	4	103s
mysql-786bb947f9-qz2fm	0/1	CrashLoopBackOff	4	113s



Anmerkung

Es kann eine Weile dauern, bis der Pod den Fehlerstatus erreicht.

- 3.3. Verwenden Sie das Secret `mysql`, um Umgebungsvariablen in der Bereitstellung `mysql` zu initialisieren. Die Bereitstellung benötigt die Umgebungsvariablen `MYSQL_USER`, `MYSQL_PASSWORD` und `MYSQL_DATABASE` für eine erfolgreiche Initialisierung. Das Secret enthält die Schlüssel `user`, `password` und `database`, die Sie der Bereitstellung als Umgebungsvariablen zuweisen können, indem Sie das Präfix `MYSQL_` hinzufügen.

```
[student@workstation ~]$ oc set env deployment/mysql --from secret/mysql \
> --prefix MYSQL_
deployment.apps/mysql updated
```

Kapitel 4 | Konfigurieren der Anwendungssicherheit

- 3.4. Um zu demonstrieren, wie ein Secret als Volume gemountet werden kann, mounten Sie das Secret `mysql` im Verzeichnis `/run/kubernetes/mysql` im Pod.

```
[student@workstation ~]$ oc set volume deployment/mysql --add --type secret \
>   --mount-path /run/secrets/mysql --secret-name mysql
info: Generated volume name: volume-nrh7r
deployment.apps/mysql volume updated
```

- 3.5. Ändern Sie die Bereitstellung mit dem Befehl `oc set env` oder `oc set volume`, um eine neue Anwendungsbereitstellung auszulösen. Überprüfen Sie, ob die `mysql`-Anwendung nach den Änderungen erfolgreich bereitgestellt wurde.

```
[student@workstation ~]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
mysql-7cd7499d66-gm2rh   1/1     Running   0          21s
```

Notieren Sie sich den Pod-Namen im Status `Running`. Sie benötigen ihn für die weiteren Schritten.

- 4. Überprüfen Sie, ob die Datenbank nun mit den vom `mysql`-Secret initialisierten Umgebungsvariablen authentifiziert wird.

- 4.1. Öffnen Sie eine Remote-Shell-Sitzung mit dem `mysql`-Pod im Status `Running`.

```
[student@workstation ~]$ oc rsh mysql-7cd7499d66-gm2rh
sh-4.4$
```

- 4.2. Starten Sie eine MySQL-Sitzung, um zu verifizieren, dass durch das `mysql`-Secret initialisierte Umgebungsvariablen zur Konfiguration der `mysql`-Anwendung verwendet wurden.

```
sh-4.4$ mysql -u myuser --password=redhat123 test_secrets -e 'show databases;'
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+
| Database      |
+-----+
| information_schema |
| test_secrets   |
+-----+
```

- 4.3. Listen Sie die Mount-Punkte im Pod auf, die das Muster `mysql` enthalten. Beachten Sie, dass der Mount-Punkt durch ein temporäres Dateisystem (`tmpfs`) gesichert wird. Dies gilt für alle Secrets, die als Volumes gemountet werden.

```
sh-4.4$ df -h | grep mysql
tmpfs      7.9G   16K   7.9G   1% /run/secrets/mysql
```

- 4.4. Untersuchen Sie die Dateien, die am Mount-Punkt `/run/secrets/mysql` gemountet sind. Jede Datei entspricht einem Schlüsselnamen im Secret, und die Dateien enthalten jeweils den Wert des entsprechenden Schlüssels.

```
sh-4.4$ for FILE in $(ls /run/secrets/mysql)
> do
> echo "${FILE}: $(cat /run/secrets/mysql/${FILE})"
> done
database: test_secrets
hostname: mysql
password: redhat123
user: myuser
```

- 4.5. Schließen Sie die Remote-Shell-Sitzung, um von Ihrem Rechner `workstation` aus fortzufahren.

```
sh-4.4$ exit
exit
[student@workstation ~]$
```

- 5. Erstellen Sie eine neue Anwendung, welche die MySQL-Datenbank verwendet.

- 5.1. Erstellen Sie eine neue Anwendung mit dem Image `redhattraining/famous-quotes` von Quay.io.

```
[student@workstation ~]$ oc new-app --name quotes \
>   --docker-image quay.io/redhattraining/famous-quotes:2.1
--> Found container image 7ff1a7b (7 months old) from quay.io for "quay.io/
redhattraining/famous-quotes:latest"
...output omitted...
--> Creating resources ...
imagestream.image.openshift.io "quotes" created
deployment.apps "quotes" created
service "quotes" created
--> Success
...output omitted...
```

- 5.2. Überprüfen Sie den Status des Anwendungs-Pods `quotes`: Der Pod zeigt einen Fehler an, da er keine Verbindung zur Datenbank herstellen kann. Es kann einige Zeit in Anspruch nehmen, bis dies in der Ausgabe angezeigt wird. Drücken Sie `Strg+C`, um den Befehl zu verlassen.

NAME	READY	STATUS	RESTARTS	AGE
quotes-6b658d57bc-vsz8q	0/1	CrashLoopBackOff	3	86s
quotes-6b658d57bc-vsz8q	0/1	Error	4	108s
quotes-6b658d57bc-vsz8q	0/1	CrashLoopBackOff	4	2m1s

- 6. Die `quotes`-Anwendung benötigt mehrere Umgebungsvariablen. Das Secret `mysql` kann Umgebungsvariablen für die Anwendung `quotes` initialisieren, indem das Präfix `QUOTES_` hinzugefügt wird.

- 6.1. Verwenden Sie das `mysql`-Geheimnis, um die folgenden Umgebungsvariablen zu initialisieren, die die `quotes`-Anwendung zum Verbinden mit der Datenbank benötigt: `QUOTES_USER`, `QUOTES_PASSWORD`, `QUOTES_DATABASE` und `QUOTES_HOSTNAME`,

Kapitel 4 | Konfigurieren der Anwendungssicherheit

die den Schlüsseln user, password, database und hostname des mysql-Geheimnisses entsprechen.

```
[student@workstation ~]$ oc set env deployment/quotes --from secret/mysql \
>   --prefix QUOTES_
deployment.apps/quotes updated
```

- 6.2. Warten Sie, bis die Anwendung quotes erfolgreich erneut bereitgestellt wurde. Die älteren Pods werden automatisch beendet.

```
[student@workstation ~]$ oc get pods -l deployment=quotes
NAME                  READY   STATUS    RESTARTS   AGE
quotes-77df54758b-mqdtf   1/1     Running   3          7m17s
```

**Anmerkung**

Es kann eine Weile dauern, bis der Pod die Bereitstellung beendet hat.

- ▶ 7. Überprüfen Sie, ob sich der Pod quotes erfolgreich mit der Datenbank verbindet und ob in der Anwendung eine Liste von Anführungszeichen angezeigt wird.
- 7.1. Überprüfen Sie die Pod-Protokolle mit dem Befehl `oc logs`. Die Protokolle zeigen eine erfolgreiche Datenbankverbindung an.

```
[student@workstation ~]$ oc logs quotes-77df54758b-mqdtf | head -n2
... Connecting to the database: myuser:redhat123@tcp(mysql:3306)/test_secrets
... Database connection OK
```

- 7.2. Stellen Sie den Service quotes bereit, um ihn von außerhalb des Clusters erreichbar zu machen.

```
[student@workstation ~]$ oc expose service quotes \
>   --hostname quotes.apps.ocp4.example.com
route.route.openshift.io/quotes exposed
```

- 7.3. Überprüfen Sie den Hostnamen der Anwendung.

```
[student@workstation ~]$ oc get route quotes
NAME      HOST/PORT           PATH  SERVICES  PORT  ...
quotes   quotes.apps.ocp4.example.com  quotes  8000-tcp ...
```

- 7.4. Überprüfen Sie, ob die Variablen in der Anwendung ordnungsgemäß festgelegt wurden, indem Sie auf den REST API-Einstiegspunkt `env` zugreifen.

```
[student@workstation ~]$ curl -s \
>   http://quotes.apps.ocp4.example.com/env | grep QUOTES_
<li>QUOTES_USER: myuser </li>
<li>QUOTES_PASSWORD: redhat123 </li>
<li>QUOTES_DATABASE: test_secrets</li>
<li>QUOTES_HOST: mysql</li>
```

Kapitel 4 | Konfigurieren der Anwendungssicherheit

- 7.5. Greifen Sie auf den REST API-Einstiegspunkt `status` der Anwendung zu, um die Datenbankverbindung zu testen.

```
[student@workstation ~]$ curl -s http://quotes.apps.ocp4.example.com/status
Database connection OK
```

- 7.6. Testen Sie die Anwendungsfunktionalität, indem Sie auf den REST API-Einstiegspunkt `random` zugreifen.

```
[student@workstation ~]$ curl -s http://quotes.apps.ocp4.example.com/random
8: Those who can imagine anything, can create the impossible.
- Alan Turing
```

- 8. Entfernen Sie das Projekt `authorization-secrets`.

```
[student@workstation ~]$ oc delete project authorization-secrets
project.project.openshift.io "authorization-secrets" deleted
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab authorization-secrets finish
```

Hiermit ist die angeleitete Übung beendet.

Kontrollieren von Anwendungsberechtigungen mit Sicherheitskontextbeschränkungen

Ziele

Am Ende dieses Abschnitts sollten Sie zu Folgendem in der Lage sein: Erstellen von Servicekonten und Anwenden von Berechtigungen sowie Verwalten von Sicherheitskontextbeschränkungen.

Sicherheitskontextbeschränkungen (Security Context Constraints, SCCs)

Red Hat OpenShift bietet einen Sicherheitsmechanismus mit der Bezeichnung *Security Context Constraints* (SCCs), mit dem in OpenShift der Zugriff auf Ressourcen, nicht jedoch auf Vorgänge, eingeschränkt wird.

SCCs beschränken in OpenShift den Zugriff von ausgeführten Pods auf die Host-Umgebung. SCCs-Steuerung:

- Ausführen privilegierter Container
- Anfordern zusätzlicher Funktionen für Container
- Verwenden von Host-Verzeichnissen als Volumes
- Änderungen am SELinux-Kontext von Containern
- Änderungen an der Benutzer-ID

Einige von der Community entwickelte Container benötigen unter Umständen gelockerte Sicherheitskontextbeschränkungen für den Zugriff auf Ressourcen, die standardmäßig geschützt sind (z. B. Dateisysteme, Sockets oder der Zugriff auf den SELinux-Kontext).

Sie können den folgenden Befehl als Cluster-Administrator ausführen, um die von OpenShift definierten SCCs anzuzeigen:

```
[user@host ~]$ oc get scc
```

OpenShift bietet acht Standard-SCCs:

- anyuid
- hostaccess
- hostmount-anyuid
- hostnetwork
- node-exporter
- nonroot
- privileged
- restricted

Weitere Informationen zu einer SCC erhalten Sie über den Befehl `oc describe`:

```
[user@host ~]$ oc describe scc anyuid
Name:          anyuid
Priority:      10
Access:
  Users:        <none>
  Groups:       system:cluster-admins
Settings:
...output omitted...
```

Die meisten von OpenShift erstellten Pods verwenden die Sicherheitskontextbeschränkung (SCC) namens **restricted**. Diese bietet einen eingeschränkten Zugriff auf Ressourcen außerhalb von OpenShift. Verwenden Sie den Befehl `oc describe`, um die Sicherheitskontextbeschränkung eines Pods anzuzeigen.

```
[user@host ~]$ oc describe pod console-5df4fcbb47-67c52 \
>   -n openshift-console | grep scc
          openshift.io/scc: restricted
```

Container-Images aus öffentlichen Container-Registries wie Docker Hub können möglicherweise nicht mit der **restricted** SCC ausgeführt werden. Ein Container-Image, das mit einer bestimmten Benutzer-ID ausgeführt werden muss, kann beispielsweise fehlschlagen, da die **restricted** SCC den Container mit einer zufälligen Benutzer-ID ausführt. Ein Container-Image, das an Port 80 oder Port 443 auf Verbindungen wartet, kann aus ähnlichen Gründen fehlschlagen. Die zufällige Benutzer-ID der **restricted** SCC kann keine Dienste starten, die privilegierte Netzwerk-Ports überwachen (Portnummern kleiner als 1024). Verwenden Sie den Unterbefehl `scc-subject-review`, um alle Sicherheitskontextbeschränkungen zurückzugeben, mit denen Sie die Einschränkungen eines Containers überwinden können:

```
[user@host ~]$ oc get pod podname -o yaml | \
>   oc adm policy scc-subject-review -f -
```

Bei der SCC `anyuid` wird die Strategie `run as user` als RunAsAny definiert; das bedeutet, der Pod kann als jede beliebige Benutzer-ID ausgeführt werden, die im Container verfügbar ist. Mit dieser Strategie können Container, die einen bestimmten Benutzer erfordern, mit einer spezifischen Benutzer-ID ausgeführt werden.

Um einen Container mit einer anderen Sicherheitskontextbeschränkung auszuführen, müssen Sie ein Servicekonto erstellen, das mit einem Pod verknüpft ist. Verwenden Sie den Befehl `oc create serviceaccount`, um das Servicekonto zu erstellen, und verwenden Sie die Option `-n`, um das Dienstkonto in einem anderen Namespace als dem aktuellen zu erstellen:

```
[user@host ~]$ oc create serviceaccount service-account
```

Verwenden Sie den Befehl `oc adm policy`, um das Servicekonto mit einer Sicherheitskontextbeschränkung zu verknüpfen. Verwenden Sie die Option `-z`, um ein Servicekonto zu identifizieren, und die Option `-n`, falls sich das Servicekonto in einem anderen Namespace als dem aktuellen befindet:

```
[user@host ~]$ oc adm policy add-scc-to-user SCC -z service-account
```



Wichtig

Nur Cluster-Administratoren können SCCs zu Servicekonten zuweisen oder aus Servicekonten entfernen. Die Ausführung von Pods mit weniger restriktiven SCCs kann die Sicherheit Ihres Clusters beeinträchtigen. Verwenden Sie diese Pods mit Vorsicht.

Mit dem Befehl `oc set serviceaccount deployment/deployment-name \> service-account-name`

Wenn der Befehl erfolgreich ausgeführt wird, werden die mit der Bereitstellung oder Bereitstellungskonfiguration verknüpften Pods erneut bereitgestellt.

Privilegierte Container

Einige Container müssen u. U. auf die Laufzeitumgebung des Hosts zugreifen. Beispiel: Die S2I-Builder-Container sind eine Klasse von privilegierten Containern, die Zugriff jenseits der Grenzen ihrer eigenen Container erfordern. Diese Container können ein Sicherheitsrisiko darstellen, da sie alle Ressourcen auf einem OpenShift-Knoten verwenden können. Mit SCCs können Sie Servicekonten mit privilegiertem Zugriff erstellen, um den Zugriff auf privilegierte Container zu ermöglichen.



Literaturhinweise

Weitere Informationen finden Sie im Kapitel *Managing Security Context Constraints* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Authentication and Authorization* unter
https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/authentication_and_authorization/index#managing-pod-security-policies

► Angeleitete Übung

Kontrollieren von Anwendungsberechtigungen mit Sicherheitskontextbeschränkungen

In dieser Übung stellen Sie Anwendungen bereit, für die Pods mit erweiterten Berechtigungen erforderlich sind.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen von Servicekonten und Zuweisen von Sicherheitskontextbeschränkungen (SCCs).
- Zuweisen eines Dienstkontos zu einer Bereitstellungskonfiguration.
- Ausführen von Anwendungen, die root-Berechtigungen benötigen.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt einige HTPasswd-Benutzer für die Übung.

```
[student@workstation ~]$ lab authorization-scc start
```

Anweisungen

- 1. Melden Sie sich bei dem OpenShift-Cluster an, und erstellen Sie das Projekt `authorization-scc`.

- 1.1. Melden Sie sich als Benutzer `developer` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Erstellen Sie das Projekt `authorization-scc`.

```
[student@workstation ~]$ oc new-project authorization-scc
Now using project "authorization-scc" on server
"https://api.ocp4.example.com:6443".
...output omitted...
```

- 2. Stellen Sie eine Anwendung mit dem Namen `gitlab` unter Verwendung des Container-Images bereit, das sich unter `quay.io/redhattraining/gitlab-ce:8.4.3-ce.0` befindet. Bei diesem Image handelt es sich um eine Kopie des unter `docker.io/gitlab/gitlab-ce:8.4.3-ce.0` verfügbaren Container-Images. Überprüfen Sie, ob der Pod fehlschlägt, da das Container-Image root-Berechtigungen benötigt.

- 2.1. Stellen Sie die Anwendung `gitlab` bereit.

```
[student@workstation ~]$ oc new-app --name gitlab \
>   --docker-image quay.io/redhattraining/gitlab-ce:8.4.3-ce.0
...output omitted...
--> Creating resources ...
  imagestream.image.openshift.io "gitlab" created
  deployment.apps "gitlab" created
  service "gitlab" created
--> Success
...output omitted...
```

- 2.2. Ermitteln Sie, ob die Anwendung erfolgreich bereitgestellt wurde. Es sollte ein Fehler auftreten, da dieses Image root-Berechtigungen für die ordnungsgemäße Bereitstellung benötigt.

```
[student@workstation ~]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
gitlab-7d67db7875-gcsjl   0/1     Error      1          60s
```



Anmerkung

Es kann eine Weile dauern, bis das Image den Status `Error` erreicht. Möglicherweise wird auch der Status `CrashLoopBackOff` angezeigt, während die Integrität des Pods überprüft wird.

- 2.3. Überprüfen Sie die Anwendungsprotokolle, um zu sehen, ob der Fehler durch unzureichende Berechtigungen verursacht wird.

```
[student@workstation ~]$ oc logs pod/gitlab-7d67db7875-gcsjl
...output omitted...
=====
Recipe Compile Error in /opt/gitlab/embedded/cookbooks/cache/cookbooks/gitlab/
recipes/default.rb
=====

Chef::Exceptions::InsufficientPermissions
-----
directory[/etc/gitlab] (gitlab::default line 26) had an error:
  Chef::Exceptions::InsufficientPermissions: Cannot create directory[/etc/gitlab]
  at /etc/gitlab due to insufficient permissions
...output omitted...
```

- 2.4. Melden Sie sich als Benutzer `admin` an.

Kapitel 4 | Konfigurieren der Anwendungssicherheit

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
```

- 2.5. Überprüfen Sie, ob Sie das Berechtigungsproblem mit einer anderen SCC beheben können.

```
[student@workstation ~]$ oc get pod/gitlab-7d67db7875-gcsjl -o yaml \
>   | oc adm policy scc-subject-review -f -
RESOURCE                      ALLOWED BY
Pod/gitlab-7d67db7875-gcsjl  anyuid
```

- 3. Erstellen Sie ein neues Servicekonto und weisen Sie ihm die SCC anyuid zu.

- 3.1. Erstellen Sie ein Servicekonto mit dem Namen `gitlab-sa`.

```
[student@workstation ~]$ oc create sa gitlab-sa
serviceaccount/gitlab-sa created
```

- 3.2. Weisen Sie dem Servicekonto `gitlab-sa` die SCC `anyuid` hinzu.

```
[student@workstation ~]$ oc adm policy add-scc-to-user anyuid -z gitlab-sa
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:anyuid added: "gitlab-sa"
```

- 4. Ändern Sie die Anwendung `gitlab` so, dass das neu erstellte Servicekonto verwendet wird. Überprüfen Sie, ob die Anwendung erfolgreich bereitgestellt wurde.

- 4.1. Melden Sie sich als Benutzer `developer` an.

```
[student@workstation ~]$ oc login -u developer -p developer
Login successful.
...output omitted...
```

- 4.2. Weisen Sie der Bereitstellung `gitlab` das Servicekonto `gitlab-sa` hinzu.

```
[student@workstation ~]$ oc set serviceaccount deployment/gitlab gitlab-sa
deployment.apps/gitlab serviceaccount updated
```

- 4.3. Überprüfen Sie, ob `gitlab` erfolgreich bereitgestellt wurde. Möglicherweise müssen Sie den Befehl `oc get pods` mehrmals ausführen, bis ein aktiver Pod angezeigt wird.

```
[student@workstation ~]$ oc get pods
NAME          READY   STATUS    RESTARTS   AGE
gitlab-86d6d65-zm2fd  1/1     Running   0          55s
```

- 5. Überprüfen Sie, ob die Anwendung `gitlab` ordnungsgemäß funktioniert.

Kapitel 4 | Konfigurieren der Anwendungssicherheit

- 5.1. Stellen Sie die Anwendung `gitlab` zur Verfügung: Da der `gitlab`-Service auf den Ports 22, 80 und 443 auf Verbindungen wartet, müssen Sie die Option `--port` verwenden.

```
[student@workstation ~]$ oc expose service/gitlab --port 80 \
>   --hostname gitlab.apps.ocp4.example.com
route.route.openshift.io/gitlab exposed
```

- 5.2. Ermitteln Sie die bereitgestellte Route.

```
[student@workstation ~]$ oc get routes
NAME      HOST/PORT          PATH  SERVICES  PORT  ...
gitlab    gitlab.apps.ocp4.example.com  gitlab     80     ...
```

- 5.3. Überprüfen Sie, ob die `gitlab`-Anwendung HTTP-Abfragen beantwortet.

```
[student@workstation ~]$ curl -s \
>   http://gitlab.apps.ocp4.example.com/users/sign_in | grep '<title>'
<title>Sign in · GitLab</title>
```

- 6. Löschen Sie das Projekt `authorization-scc`.

```
[student@workstation ~]$ oc delete project authorization-scc
project.project.openshift.io "authorization-scc" deleted
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab authorization-scc finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Konfigurieren der Anwendungssicherheit

In dieser praktischen Übung erstellen Sie ein Secret, um vertrauliche Konfigurationsinformationen zu teilen, und passen eine Bereitstellung so an, dass sie mit weniger restriktiven Einstellungen ausgeführt wird.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Verwenden von Secrets, um vertrauliche Informationen für Bereitstellungen verfügbar zu machen.
- Verwenden von Sicherheitskontextbeschränkungen, um Anwendungen in einer weniger restriktiven Umgebung auszuführen.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl sorgt dafür, dass die Cluster-API erreichbar ist und dass Sie sich beim Cluster anmelden können.

```
[student@workstation ~]$ lab authorization-review start
```

Anweisungen

1. Erstellen Sie das Projekt `authorization-review` als Benutzer `developer`. Alle zusätzlichen Aufgaben in dieser Übung verwenden das Projekt `authorization-review`.
2. Erstellen Sie ein Secret namens `review-secret`, das Sie mit der MySQL-Datenbank und den WordPress-Anwendungen verwenden werden. Das Secret muss drei Schlüssel-Wert-Paare enthalten: `user=wpuser`, `password=redhat123` und `database=wordpress`.
3. Stellen Sie eine MySQL-Datenbankanwendung mit dem Namen `mysql` und dem Container-Image unter `registry.redhat.io/rhel8/mysql-80:1` bereit. Passen Sie die Bereitstellung `mysql` so an, dass sie ihre Umgebungsvariablen aus dem Secret `review-secret` abruft. Die Umgebungsvariablen müssen das Präfix `MYSQL_` verwenden.
4. Stellen Sie eine WordPress-Anwendung mit dem Namen `wordpress` und dem Container-Image unter `quay.io/redhattraining/wordpress:5.7-php7.4-apache` bereit. Fügen Sie beim Erstellen der Anwendung die Umgebungsvariablen `WORDPRESS_DB_HOST=mysql`, `WORDPRESS_DB_NAME=wordpress`, `WORDPRESS_TITLE=auth-review`, `WORDPRESS_USER=wpuser`, `WORDPRESS_PASSWORD=redhat123`, `WORDPRESS_EMAIL= student@redhat.com` und `WORDPRESS_URL=wordpress-review.apps.ocp4.example.com` hinzu. Ändern Sie die `wordpress`-Bereitstellung nach dem Bereitstellen so, dass sie das Secret `review-secret` als zusätzliche Umgebungsvariablen verwendet. Die zusätzlichen Umgebungsvariablen müssen das Präfix `WORDPRESS_DB_` verwenden.



Anmerkung

Der `wordpress`-Pod wird erst erfolgreich ausgeführt, wenn Sie die Bereitstellung so anpassen, dass sie eine weniger restriktive Sicherheitskontextbeschränkung verwendet.

5. Identifizieren Sie als `admin`-Benutzer eine weniger restriktive SCC, mit der die `wordpress`-Bereitstellung erfolgreich ausgeführt werden kann. Erstellen Sie ein Servicekonto mit dem Namen `wordpress-sa`, und weisen Sie ihm die Sicherheitskontextbeschränkung `anyuid` zu. Passen Sie die `wordpress`-Bereitstellung so an, dass sie das Servicekonto `wordpress-sa` verwendet.
6. Stellen Sie als Benutzer `developer` den Service `wordpress` für externe Anfragen unter dem Hostnamen `wordpress-review.apps.ocp4.example.com` zur Verfügung. Greifen Sie über einen Webbrowser auf die Route zu und überprüfen Sie, ob der Setup-Assistent in der WordPress-Anwendung angezeigt wird.

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.

```
[student@workstation ~]$ lab authorization-review grade
```

Beenden

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab authorization-review finish
```

Hiermit ist die praktische Übung beendet.

► Lösung

Konfigurieren der Anwendungssicherheit

In dieser praktischen Übung erstellen Sie ein Secret, um vertrauliche Konfigurationsinformationen zu teilen, und passen eine Bereitstellung so an, dass sie mit weniger restriktiven Einstellungen ausgeführt wird.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Verwenden von Secrets, um vertrauliche Informationen für Bereitstellungen verfügbar zu machen.
- Verwenden von Sicherheitskontextbeschränkungen, um Anwendungen in einer weniger restriktiven Umgebung auszuführen.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl sorgt dafür, dass die Cluster-API erreichbar ist und dass Sie sich beim Cluster anmelden können.

```
[student@workstation ~]$ lab authorization-review start
```

Anweisungen

1. Erstellen Sie das Projekt `authorization-review` als Benutzer `developer`. Alle zusätzlichen Aufgaben in dieser Übung verwenden das Projekt `authorization-review`.
 - 1.1. Melden Sie sich als Benutzer `developer` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \
>   https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Erstellen Sie das Projekt `authorization-review`.

```
[student@workstation ~]$ oc new-project authorization-review
Now using project "authorization-review" on server
"https://api.ocp4.example.com:6443".
...output omitted...
```

2. Erstellen Sie ein Secret namens `review-secret`, das Sie mit der MySQL-Datenbank und den WordPress-Anwendungen verwenden werden. Das Secret muss drei Schlüssel-Wert-Paare enthalten: `user=wpuser`, `password=redhat123` und `database=wordpress`.
 - 2.1. Erstellen Sie ein Secret mit dem Namen `review-secret`.

```
[student@workstation ~]$ oc create secret generic review-secret \
>   --from-literal user=wpuser --from-literal password=redhat123 \
>   --from-literal database=wordpress
secret/review-secret created
```

3. Stellen Sie eine MySQL-Datenbankanwendung mit dem Namen `mysql` und dem Container-Image unter `registry.redhat.io/rhel8/mysql-80:1` bereit. Passen Sie die Bereitstellung `mysql` so an, dass sie ihre Umgebungsvariablen aus dem Secret `review-secret` abruft. Die Umgebungsvariablen müssen das Präfix `MYSQL_` verwenden.

- 3.1. Erstellen Sie eine neue Anwendung, um einen `mysql`-Datenbankserver bereitzustellen.

```
[student@workstation ~]$ oc new-app --name mysql \
>   --docker-image registry.redhat.io/rhel8/mysql-80:1
...output omitted...
--> Creating resources ...
  imagestream.image.openshift.io "mysql" created
  deployment.apps "mysql" created
  service "mysql" created
--> Success
...output omitted...
```

- 3.2. Verwenden Sie das Secret `review-secret`, um Umgebungsvariablen in der `mysql`-Bereitstellungskonfiguration zu initialisieren. Mit der Option `--prefix` wird sichergestellt, dass alle Variablen, die aus dem Secret in den Pod injiziert wurden, mit `MYSQL_` beginnen.

```
[student@workstation ~]$ oc set env deployment/mysql --prefix MYSQL_ \
>   --from secret/review-secret
deployment.apps/mysql updated
```

- 3.3. Überprüfen Sie, ob der `mysql`-Pod erfolgreich bereitgestellt wurde.

```
[student@workstation ~]$ watch oc get pods
```

Drücken Sie STRG+C, um den Befehl `watch` zu beenden, nachdem der `mysql`-Pod mit `1/1` und mit `Running` angezeigt wird.

NAME	READY	STATUS	RESTARTS	AGE
mysql-f675b96f8-vspb9	1/1	Running	0	20s



Anmerkung

Nach dem Festlegen des Secrets kann es einige Minuten dauern, bis die Bereitstellung erfolgreich ausgeführt wird.

4. Stellen Sie eine WordPress-Anwendung mit dem Namen `wordpress` und dem Container-Image unter `quay.io/redhattraining/wordpress:5.7-php7.4-apache` bereit. Fügen Sie beim Erstellen der Anwendung die Umgebungsvariablen

`WORDPRESS_DB_HOST=mysql, WORDPRESS_DB_NAME=wordpress,`
`WORDPRESS_TITLE=auth-review, WORDPRESS_USER=wpuser,`
`WORDPRESS_PASSWORD=redhat123, WORDPRESS_EMAIL= student@redhat.com und`
`WORDPRESS_URL=wordpress-review.apps.ocp4.example.com` hinzu. Ändern Sie die
 wordpress-Bereitstellung nach dem Bereitstellen so, dass sie das Secret `review-secret`
 als zusätzliche Umgebungsvariablen verwendet. Die zusätzlichen Umgebungsvariablen
 müssen das Präfix `WORDPRESS_DB_` verwenden.



Anmerkung

Der `wordpress`-Pod wird erst erfolgreich ausgeführt, wenn Sie die Bereitstellung so anpassen, dass sie eine weniger restriktive Sicherheitskontextbeschränkung verwendet.

- 4.1. Stellen Sie eine `wordpress`-Anwendung bereit.

```
[student@workstation ~]$ oc new-app --name wordpress \
>   --docker-image quay.io/redhattraining/wordpress:5.7-php7.4-apache \
>   -e WORDPRESS_DB_HOST=mysql \
>   -e WORDPRESS_DB_NAME=wordpress \
>   -e WORDPRESS_TITLE=auth-review \
>   -e WORDPRESS_USER=wpuser \
>   -e WORDPRESS_PASSWORD=redhat123 \
>   -e WORDPRESS_EMAIL=student@redhat.com \
>   -e WORDPRESS_URL=wordpress-review.apps.ocp4.example.com
...output omitted...
-> Creating resources ...
  imagestream.image.openshift.io "wordpress" created
  deployment.apps "wordpress" created
  service "wordpress" created
--> Success
...output omitted...
```

- 4.2. Verwenden Sie das Secret `review-secret`, um Umgebungsvariablen in der `wordpress`-Bereitstellungskonfiguration zu initialisieren. Mit der Option `--prefix` wird sichergestellt, dass die Variablen, die aus dem Secret in den Pod injiziert wurden, mit `WORDPRESS_DB_` beginnen.

```
[student@workstation ~]$ oc set env deployment/wordpress \
>   --prefix WORDPRESS_DB_ --from secret/review-secret
deployment.apps/wordpress updated
```

- 4.3. Vergewissern Sie sich, dass der `wordpress`-Pod nicht erfolgreich erneut bereitgestellt wurde, selbst nachdem Sie Variablen aus dem Secret `review-secret` injiziert haben.

```
[student@workstation ~]$ watch oc get pods -l deployment=wordpress
```

Warten Sie bis zu einer Minute, und drücken Sie dann STRG+C, um den Befehl `watch` zu beenden. Der `wordpress`-Pod wird immer wieder neu gestartet. Bei jedem Neustart wechselt der Pod zunächst in den Status `Error` und dann zu `CrashLoopBackOff`.

Kapitel 4 | Konfigurieren der Anwendungssicherheit

```
Every 2.0s: oc get pods -l deployment=wordpress
...
NAME          READY   STATUS      RESTARTS   AGE
wordpress-68c49c9d4-wq46g   0/1     CrashLoopBackOff   5          4m30s
```

4.4. Überprüfen Sie die Pod-Protokolle auf Fehlermeldungen.

```
[student@workstation ~]$ oc logs wordpress-68c49c9d4-wq46g
...output omitted...
(13)Permission denied: AH00072: make_sock: could not bind to address [::]:80
(13)Permission denied: AH00072: make_sock: could not bind to address 0.0.0.0:80
no listening sockets available, shutting down
AH00015: Unable to open logs
```

In der Standardeinstellung verhindert OpenShift, dass Pods Dienste starten, die Ports unter 1024 überwachen.

5. Identifizieren Sie als `admin`-Benutzer eine weniger restriktive SCC, mit der die `wordpress`-Bereitstellung erfolgreich ausgeführt werden kann. Erstellen Sie ein Servicekonto mit dem Namen `wordpress-sa`, und weisen Sie ihm die Sicherheitskontextbeschränkung `anyuid` zu. Passen Sie die `wordpress`-Bereitstellung so an, dass sie das Servicekonto `wordpress-sa` verwendet.

5.1. Melden Sie sich als Benutzer `admin` an.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
```

5.2. Überprüfen Sie, ob Sie das Berechtigungsproblem mit einer anderen SCC beheben können.

```
[student@workstation ~]$ oc get pod/wordpress-68c49c9d4-wq46g -o yaml \
>   | oc adm policy scc-subject-review -f -
RESOURCE           ALLOWED BY
Pod/wordpress-68c49c9d4-wq46g   anyuid
```

**Wichtig**

Der Befehl `oc adm policy` kann nur vom `admin`-Benutzer ausgeführt werden.

5.3. Erstellen Sie ein Servicekonto mit dem Namen `wordpress-sa`.

```
[student@workstation ~]$ oc create serviceaccount wordpress-sa
serviceaccount/wordpress-sa created
```

5.4. Erteilen Sie dem Servicekonto `wordpress-sa` die SCC `anyuid`. Wenn der WordPress-Pod mit dem `root`-Benutzer ausgeführt wird, erlaubt OpenShift dem Pod, einen Service auf Port 80 zu starten.

Kapitel 4 | Konfigurieren der Anwendungssicherheit

```
[student@workstation ~]$ oc adm policy add-scc-to-user anyuid -z wordpress-sa
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:anyuid added:
"wordpress-sa"
```

- 5.5. Konfigurieren Sie die `wordpress`-Bereitstellung so, dass sie das Servicekonto `wordpress-sa` verwendet.

```
[student@workstation ~]$ oc set serviceaccount deployment/wordpress \
>   wordpress-sa
deployment.apps/wordpress serviceaccount updated
```

- 5.6. Überprüfen Sie, ob der `wordpress`-Pod nach dem Festlegen des Servicekontos erfolgreich bereitgestellt wird.

```
[student@workstation ~]$ watch oc get pods -l deployment=wordpress
```

Drücken Sie STRG+C, um den Befehl `watch` zu beenden, nachdem der `wordpress`-Pod mit 1/1 und mit `Running` angezeigt wird.

```
Every 2.0s: oc get pods -l deployment=wordpress
```

```
...
```

NAME	READY	STATUS	RESTARTS	AGE
wordpress-bcb5d97f6-mwljs	1/1	Running	0	21s

6. Stellen Sie als Benutzer `developer` den Service `wordpress` für externe Anfragen unter dem Hostnamen `wordpress-review.apps.ocp4.example.com` zur Verfügung. Greifen Sie über einen Webbrowser auf die Route zu und überprüfen Sie, ob der Setup-Assistent in der WordPress-Anwendung angezeigt wird.

- 6.1. Verwenden Sie den Befehl `oc expose`, um eine Route zur `wordpress`-Anwendung zu erstellen.

```
[student@workstation ~]$ oc expose service/wordpress \
>   --hostname wordpress-review.apps.ocp4.example.com
route.route.openshift.io/wordpress exposed
```

- 6.2. Überprüfen Sie mit einem Webbrowser den Zugriff auf die URL `http://wordpress-review.apps.ocp4.example.com`. Wenn Sie die Anwendung ordnungsgemäß bereitgestellt haben, wird ein Setup-Assistent in einem Browser angezeigt.

Alternativ können Sie den Befehl `curl` verwenden, um direkt auf die Installations-URL zuzugreifen.

```
[student@workstation ~]$ curl -s \
>   http://wordpress-review.apps.ocp4.example.com/wp-admin/install.php \
>   | grep Installation
<title>WordPress &rsaquo; Installation</title>
```

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.

```
[student@workstation ~]$ lab authorization-review grade
```

Beenden

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab authorization-review finish
```

Hiermit ist die praktische Übung beendet.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Mit Secret-Ressourcen können Sie vertrauliche Informationen von Anwendungs-Pods trennen. Sie stellen Secrets für einen Anwendungs-Pod entweder als Umgebungsvariablen oder als normale Dateien bereit.
- OpenShift verwendet Sicherheitskontext-Beschränkungen (SCCs), um zulässige Pod-Interaktionen mit Systemressourcen zu definieren. Standardmäßig funktionieren Pods im Kontext `restricted`, wodurch der Zugriff auf Knoten-Ressourcen eingeschränkt wird.

Kapitel 5

Konfigurieren des OpenShift-Netzwerks für Anwendungen

Ziel

Beheben von Fehlern in OpenShift Software-defined Networking (SDN) und Konfigurieren von Netzwerkrichtlinien

Ziele

- Beheben von Fehlern des SDN von OpenShift über die Befehlszeilenschnittstelle
- Zulassen und Absichern von Netzwerkverbindungen zu Anwendungen in einem OpenShift-Cluster
- Beschränken des Netzwerkdatenverkehrs zwischen Projekten und Pods

Abschnitte

- Beheben von OpenShift-SDN-Fehlern (und angeleitete Übung)
- Bereitstellen von Anwendungen für externe Zugriffe (und angeleitete Übung)
- Konfigurieren von Netzwerk-Richtlinien (und angeleitete Übung)

Praktische Übung

Konfigurieren des OpenShift-Netzwerks für Anwendungen

Beheben von Software-Defined Networking-Fehlern in OpenShift

Ziele

Nach der Bearbeitung dieses Abschnitts können Sie eine Fehlerbehebung des OpenShift-SDN über die Befehlszeilenschnittstelle durchführen.

Einführung in OpenShift Software Defined Networking

OpenShift implementiert ein Software Defined Network (SDN), um die Netzwerkinfrastruktur von Cluster und Benutzeranwendungen zu verwalten. SDN bezeichnet ein Netzwerkmodell, in dem Sie die Netzwerkservices durch Abstraktion mehrerer Netzwerkschichten verwalten. Es entkoppelt die Software für die Datenübertragung, auch *Control Plane* genannt, von den zugrunde liegenden Mechanismen für die Weiterleitung des Datenverkehrs (*Datenschicht*). Unter den vielen Features von SDN bieten offene Standards den Anbietern die Möglichkeit, ihre Lösungen, ein zentrales Management, ein dynamisches Routing und eine Mandanten-Isolierung vorzuschlagen.

In der OpenShift Container Platform erfüllt SDN die folgenden fünf Anforderungen:

- Programmgesteuerte Verwaltung des Netzwerkverkehrs und der Netzwerkressourcen, sodass die Organisationsteams entscheiden können, wie ihre Anwendungen bereitgestellt werden.
- Verwalten der Kommunikation zwischen Containern, die im selben Projekt ausgeführt werden
- Verwalten der Kommunikation zwischen Pods, unabhängig davon, ob sie zu demselben Projekt gehören oder in separaten Projekten ausgeführt werden.
- Verwalten der Netzwerkkommunikation von einem Pod zu einem Service.
- Verwalten der Netzwerkkommunikation von einem externen Netzwerk zu einem Dienst oder von Containern zu externen Netzwerken.

Die SDN-Implementierung erstellt ein abwärtskompatibles Modell, in dem die Pods in Bezug auf Portzuweisung, IP-Adress-Leasing und Reservierung den virtuellen Rechnern entsprechen.

Diskutieren des OpenShift-Netzwerkmodells

Die Container-Netzwerkschnittstelle (CNI) ist eine gängige Schnittstelle zwischen dem Netzwerkanbieter und der Container-Ausführung und wird in Form von Netzwerk-Plug-Ins implementiert. Die CNI enthält die Spezifikation für Plug-Ins zur Konfiguration von Netzwerkschnittstellen in Containern. Mit den gemäß dieser Spezifikation geschriebenen Plug-ins können verschiedene Netzwerkanbieter das OpenShift-Cluster-Netzwerk steuern.

SDN verwendet CNI-Plug-Ins, um Linux-Namespaces zu erstellen und die Nutzung von Ressourcen und Prozessen auf physischen und virtuellen Hosts zu partitionieren. Mit dieser Implementierung können Container in Pods Netzwerkressourcen wie Geräte, IP-Stacks, Firewall-Regeln und Routing-Tabellen freigeben. SDN weist jedem Pod eine eindeutige routingfähige IP-Adresse zu, damit alle anderen Services im selben Netzwerk auf den Pod zugreifen können.

Einige gängige CNI-Plugins, die in OpenShift verwendet werden:

- OpenShift-SDN

- OVN-Kubernetes
- Kuryr

In OpenShift 4.6 sind OpenShift SDN und OVN-Kubernetes die Standard-Netzwerkanbieter.

Der OpenShift SDN-Netzwerkanbieter verwendet Open vSwitch (OVS) für Verbindungen zwischen Pods auf demselben Knoten und Virtual Extensible LAN (VXLAN)-Tunneling, um Knoten miteinander zu verbinden. OVN-Kubernetes verwendet Open Virtual Network (OVN), um das Cluster-Netzwerk zu verwalten. OVN erweitert OVS mit virtuellen Netzwerk-Abstraktionen. Kuryr stellt Netzwerkfunktionen über Neutron und Octavia Red Hat OpenStack Platform Services bereit.

Migrieren von Legacy-Anwendungen

Das SDN-Design erleichtert die Containerisierung Ihrer Legacy-Anwendungen, da Sie die Art und Weise, wie die Anwendungskomponenten miteinander kommunizieren, nicht ändern müssen. Wenn Ihre Anwendung aus zahlreichen Diensten besteht, die über den TCP/UDP-Stack kommunizieren, funktioniert dieser Weg immer noch, wenn Container in einem Pod denselben Netzwerk-Stack verwenden.

Das folgende Diagramm zeigt, wie alle Pods mit einem gemeinsam genutzten Netzwerk verbunden sind:

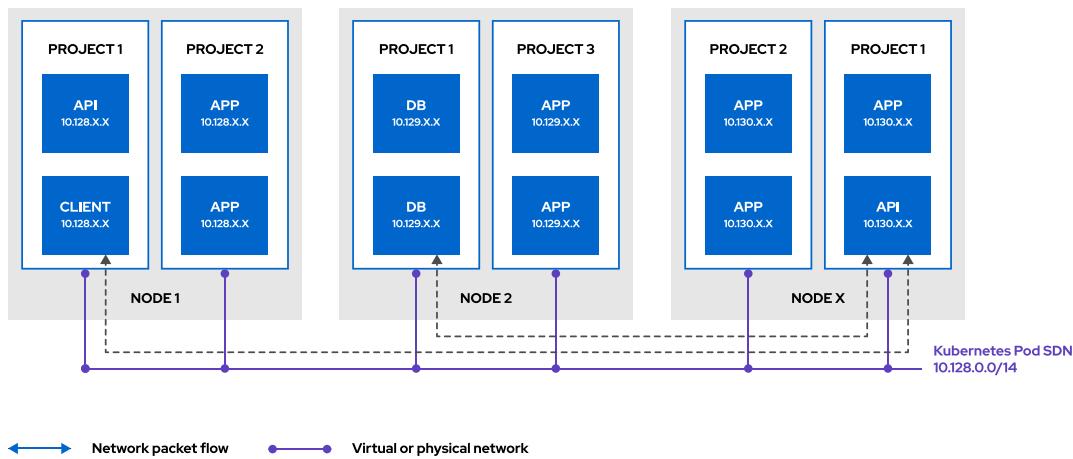


Abbildung 5.1: Kubernetes-Standardnetzwerke



Anmerkung

Der Cluster Network Operator von OpenShift verwaltet das Netzwerk; hierauf wird später noch eingegangen.

Verwenden von Services für den Zugriff auf Pods

Kubernetes bietet das Konzept eines Service, der eine essentielle Ressource in einer OpenShift-Anwendung ist. Services ermöglichen die logische Gruppierung von Pods unter einer gemeinsamen Zugriffsroute. Ein Service dient als Load-Balancer vor einem oder mehreren Pods und entkoppelt die Anwendungsspezifikationen (z. B. die Anzahl der Replikate) vom Zugriff auf die jeweilige Anwendung. Der Load Balancer verteilt Client-Anforderungen auf die Mitglieds-Pods und stellt eine stabile Schnittstelle für die Kommunikation mit Pods bereit, ohne einzelne Pod-IP-Adressen nachzuverfolgen.

Kapitel 5 | Konfigurieren des OpenShift-Netzwerks für Anwendungen

Die meisten gängigen Anwendungen werden nicht als einzelner Pod ausgeführt. Sie müssen horizontal skaliert werden; das bedeutet, dass eine Anwendung auf vielen Pods ausgeführt werden kann, um die wachsende Nachfrage der Nutzer zufriedenzustellen. In einem OpenShift-Cluster werden Pods ständig auf den Knoten im Cluster erstellt und zerstört, z. B. während der Bereitstellung einer neuen Anwendungsversion oder beim Entladen eines Knotens für die Wartung. Den Pods wird jedes Mal, wenn sie erstellt werden, eine andere IP-Adresse zugewiesen. Daher sind Pods nicht einfach adressierbar. Anstatt die IP-Adressen anderer Pods in einem Pod ermitteln zu müssen, können Sie mit Services eine einzige, eindeutige IP-Adresse für andere Pods bereitstellen, unabhängig davon, wo die Pods ausgeführt werden.

Services ermitteln anhand von Selektoren (Labels), welche Pods den Datenverkehr über den Service empfangen. Jeder Pod, dem mit diesen Selektoren übereinstimmt, wird als Endpunkt zur Serviceressource hinzugefügt. Das Pods ständig erstellt und vernichtet werden, aktualisiert der Service die Endpunkte automatisch. Die Verwendung von Selektoren sorgt für eine flexible Gestaltung der Architektur und des Routings Ihrer Anwendungen. Beispielsweise können Sie die Anwendung in Ebenen aufteilen und beschließen, für jede Ebene einen Service zu erstellen. Selektoren ermöglichen ein flexibles und hoch belastbares Design.

OpenShift verwendet zwei Subnetze: ein Subnetz für Pods und ein Subnetz für Services. Der Datenverkehr wird transparent an die Pods weitergeleitet. Ein Agent (je nach verwendetem Netzwerkmodus) verwaltet Routing-Regeln, um den Datenverkehr an die Pods weiterzuleiten, die mit den Selektoren übereinstimmen.

Das folgende Diagramm zeigt, wie drei API-Pods auf separaten Knoten ausgeführt werden. Der Service `service1` gleicht die Last zwischen diesen drei Pods aus.

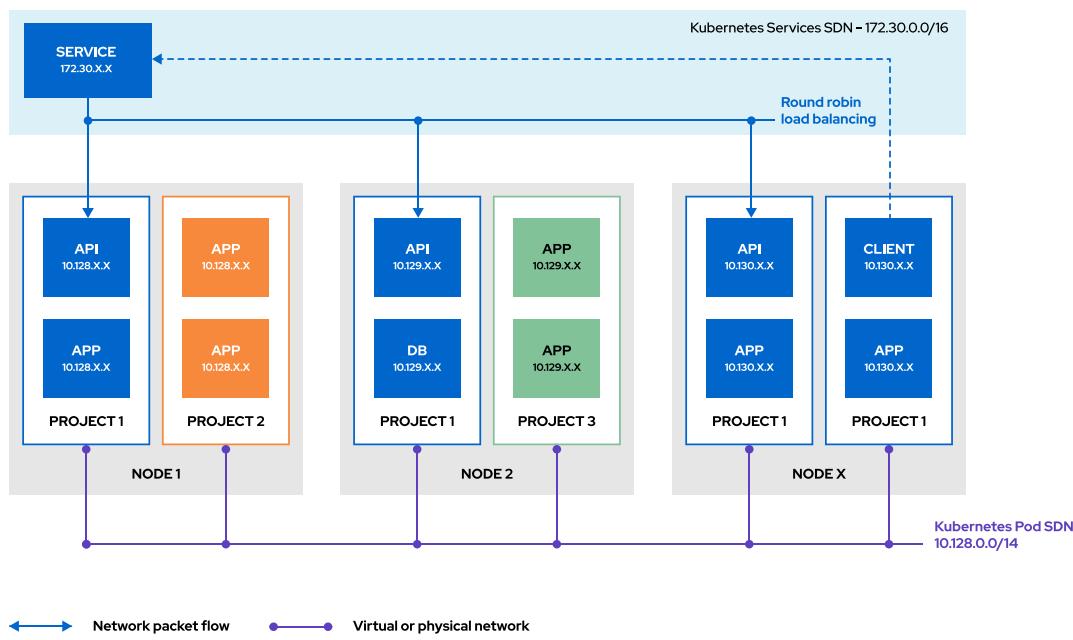


Abbildung 5.2: Verwenden von Services für den Zugriff auf Anwendungen

Die folgende YAML-Definition zeigt, wie Sie einen Service erstellen. Dadurch wird der Dienst `application-frontend` definiert, der eine virtuelle IP-Adresse erstellt, die den TCP-Port 443 verfügbar macht. Die Front-End-Anwendung überwacht den unprivilegierten Port 8843.

```
kind: Service
apiVersion: v1
metadata:
```

```

name: application-frontend 1
labels:
  app: frontend-svc 2
spec:
  ports: 3
    - name: HTTP
      protocol: TCP
      port: 443 4
      targetPort: 8443 5
  selector: 6
    app: shopping-cart
    name: frontend
  type: ClusterIP 7

```

- 1** Der Name des Service. Dieser Bezeichner ermöglicht es Ihnen, den Dienst nach seiner Erstellung zu verwalten.
- 2** Eine Bezeichnung, die Sie als Selektor verwenden können. Auf diese Weise können Sie Ihre Services logisch gruppieren.
- 3** Ein Array von Objekten, das die Netzwerkports beschreibt, die bereitgestellt werden.
Jeder Eintrag definiert den Namen für die Port-Zuordnung. Dieser Wert ist generisch und wird nur zu Identifikationszwecken verwendet.
- 4** Dies ist der Port, den der Dienst bereitstellt. Sie verwenden diesen Port, um eine Verbindung zu der Anwendung herzustellen, die vom Dienst bereitstellt wird.
- 5** Port, den die Anwendung überwacht. Der Service erstellt eine Weiterleitungsregel vom Service-Port zum Ziel-Port des Services.
- 6** Der Selektor legt fest, welche Pods im Service-Pool enthalten sind. Services verwenden diesen Selektor, um zu bestimmen, wohin der Datenverkehr weitergeleitet werden soll.
In diesem Beispiel richtet sich der Service an alle Pods, deren Labels `app: shopping-cart` und `name: frontend` entsprechen.
- 7** Hier wird der Service verfügbar gemacht. `ClusterIP` stellt den Service mit einer internen IP-Adresse für den Cluster bereit und ist der Standardwert. Andere Servicetypen werden an anderer Stelle in diesem Kurs beschrieben.

Besprechen des DNS-Operators

Der DNS-Operator stellt einen DNS-Server bereit, der von CoreDNS, einem Lightweight-DNS-Server, der in GoLang geschrieben wurde, verwaltet wird, und führt ihn aus. Der DNS-Operator ermöglicht eine DNS-Namensauflösung zwischen Pods, wodurch Services ihre Endpunkte ermitteln können.

Jedes Mal, wenn Sie eine neue Anwendung erstellen, konfiguriert OpenShift die Pods so, dass sie die CoreDNS-Service-IP für die DNS-Auflösung kontaktieren.

Führen Sie den folgenden Befehl aus, um die Konfiguration des DNS-Operators zu überprüfen:

```
[user@demo ~]$ oc describe dns.operator/default
Name:          default
...output omitted...
```

```
API Version: operator.openshift.io/v1
Kind:         DNS
...output omitted...
Spec:
Status:
  Cluster Domain: cluster.local
  Cluster IP:    172.30.0.10
...output omitted...
```

Der DNS-Operator ist für Folgendes verantwortlich:

- Erstellen eines standardmäßigen Cluster-DNS-Namens (`cluster.local`).
- Zuweisen von DNS-Namen zu den von Ihnen definierten Services (z. B. `db.backend.svc.cluster.local`).

Beispielsweise erreichen Sie von einem Pod mit dem Namen `example-6c4984d949-7m26r` den Pod `Hello` über den Service `Hello` im Projekt `Test` über den FQDN für den Service. Dies zeigt der folgende Befehl:

```
[user@demo ~]$ oc rsh example-6c4984d949-7m26r curl \
>   hello.test.svc.cluster.local:8080
```

Verwalten von DNS-Datensätzen für Services

Diese DNS-Implementierung ermöglicht es Pods, DNS-Namen für Ressourcen in einem Projekt oder Cluster nahtlos aufzulösen. Pods können ein vorhersagbares Benennungsschema für den Zugriff auf einen Service verwenden. Wenn Sie beispielsweise den Hostnamen `db.backend.svc.cluster.local` von einem Container abfragen, wird die IP-Adresse des Diensts zurückgegeben. In diesem Fall ist `db` der Name des Services, `backend` ist der Projektname, und `cluster.local` ist der Cluster-DNS-Name.

CoreDNS erstellt zwei Arten von Datensätzen für Services: A-Datensätze, die zu Services aufgelöst werden, und SRV-Datensätze, die mit dem folgenden Format übereinstimmen:

```
_port-name._port-protocol.svc-name.namespace.svc.cluster-domain.cluster-domain
```

Wenn Sie beispielsweise einen Service verwenden, der den TCP-Port 443 über den HTTPS-Dienst bereitstellt, wird der SRV-Datensatz wie folgt erstellt:

```
_443-tcp._tcp.https.frontend.svc.cluster.local
```



Anmerkung

Wenn Services nicht über eine Cluster-IP verfügen, weist der DNS-Operator Ihnen einen DNS-A-Datensatz zu, der in den IP-Satz der Pods hinter dem Service aufgelöst wird.

Entsprechend wird der neu erstellte SRV-Datensatz zu allen Pods aufgelöst, die den Service bereitstellen.

Einführung in den Cluster Network Operator

OpenShift Container Platform verwendet den Cluster Network Operator zur Verwaltung des SDN. Dazu gehören das zu verwendende Netzwerk-CIDR, der Netzwerkanbieter und die IP-Addresspools. Die Konfiguration des Cluster Network Operators erfolgt vor der Installation, obwohl Migrationen vom standardmäßigen OpenShift SDN-CNI-Netzwerkanbieter zum OVN-Kubernetes-Netzwerkanbieter unterstützt werden.

Führen Sie den folgenden Befehl `oc get` als Administrator aus, um die SDN-Konfiguration zu konsultieren, die von der benutzerdefinierten Ressourcendefinition `network.config.openshift.io` verwaltet wird:

```
[user@demo ~]$ oc get network/cluster -o yaml
apiVersion: config.openshift.io/v1
kind: Network
...output omitted...
spec:
  clusterNetwork:
    - cidr: 10.128.0.0/14 ①
      hostPrefix: 23 ②
    externalIP:
      policy: {}
    networkType: OpenshiftSDN ③
    serviceNetwork:
      - 172.30.0.0/16
...output omitted...
```

- ① Definiert die CIDR für alle Pods im Cluster. In diesem Beispiel hat das SDN eine Netzmaske von `255.252.0.0` und kann 262144 IP-Adressen zuweisen.
- ② Definiert das Host-Präfix. Der Wert 23 weist auf die Netzmaske `255.255.254.0` hin, die in 512 zuweisbare IPs übersetzt wird.
- ③ Zeigt den aktuellen SDN-Anbieter an. Sie haben die Wahl zwischen `OpenShiftSDN`, `OVNKubernetes` und `Kuryr`.



Anmerkung

Das Konfigurieren zusätzlicher Netzwerke geht über den Rahmen des Kurses hinaus. Weitere Informationen über die benutzerdefinierte Ressourcendefinition für das Kubernetes-Netzwerk finden Sie im Dokument *Kubernetes Network Custom Resource Definition De-facto Standard Version 1*, das im Abschnitt „References“ aufgeführt ist.

Einführung in Multus CNI

Multus ist ein Open Source-Projekt zur Unterstützung mehrerer Netzwerkkarten in OpenShift. Eine der Herausforderungen, die mit Multus gelöst werden, betrifft die Migration der Virtualisierung von Netzwerkfunktionen zu Containern. Multus dient als Broker und Arbiter für andere CNI-Plugins und verwaltet die Implementierung und den Lifecycle zusätzlicher Netzwerkgeräte in Containern. Multus unterstützt Plugins wie SR-IOV, vHost CNI, Flannel und Calico. Multus löst spezielle Randfälle, die häufig in Telekommunikationsdiensten, Edge-Computing und Virtualisierung auftreten, indem mehrere Netzwerkschnittstellen für Pods bereitgestellt werden.



Anmerkung

Die von Multus bereitgestellten zusätzlichen Netzwerkgeräte werden von allen Kubernetes- und OpenShift-Netzwerkfunktionen, beispielsweise Services, Ingress und Routen, ignoriert.



Literaturhinweise

Weitere Informationen finden Sie im Kapitel *Cluster Network Operator in OpenShift Container Platform* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 Networking unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/networking/index#cluster-network-operator

Cluster-Networking

<https://kubernetes.io/docs/concepts/cluster-administration/networking/>

Kubernetes Network Custom Resource Definition De-facto Standard Version 1

<https://github.com/k8snetworkplumbingwg/multi-net-spec/blob/master/v1.0/%5Bv1%5D%20Kubernetes%20Network%20Custom%20Resource%20Definition%20De-facto%20Standard.md>

CoreDNS: DNS and Service Discovery

<https://coredns.io/>

Multus-CNI

<https://github.com/intel/multus-cni>

► Angeleitete Übung

Beheben von Software-Defined Networking-Fehlern in OpenShift

In dieser Übung diagnostizieren und beheben Sie Verbindungsprobleme bei einer Kubernetes-Anwendungsbereitstellung.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Bereitstellen der Node.js-Anwendung To Do
- Erstellen einer Route zur Bereitstellung eines Anwendungsservice
- Beheben von Kommunikationsfehlern zwischen Pods in Ihrer Anwendung mit oc debug
- Aktualisieren eines OpenShift-Service

Bevor Sie Beginnen

Führen Sie auf dem Rechner workstation als Benutzer student den Befehl lab aus, um Ihr System für diese Übung vorzubereiten.

Mit dem Befehl wird sichergestellt, dass die Cluster-API erreichbar ist.

```
[student@workstation ~]$ lab network-sdn start
```

Anweisungen

Sie haben als OpenShift-Entwickler soeben die Migration der Node.js-Anwendung To Do zu OpenShift abgeschlossen. Die Anwendung besteht aus zwei Bereitstellungen, eine für die Datenbank und eine für das Front-End. Darüber hinaus enthält sie zwei Services für die Kommunikation zwischen Pods.

Obwohl die Anwendung anscheinend initialisiert wurde, können Sie nicht über einen Webbrowser darauf zugreifen. In dieser Übung führen Sie eine Fehlerbehebung bei Ihrer Anwendung durch und beheben das Problem.

- 1. Melden Sie sich bei dem OpenShift-Cluster an, und erstellen Sie das Projekt network-sdn.

- 1.1. Melden Sie sich als Benutzer developer bei dem Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \
>     https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Erstellen Sie das Projekt network-sdn.

```
[student@workstation ~]$ oc new-project network-sdn
Now using project "network-sdn" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

► 2. Stellen Sie die Datenbank bereit, und stellen Sie deren Daten wieder her.

In der Datei `/home/student/D0280/labs/network-sdn/todo-db.yaml` sind die folgenden Ressourcen definiert:

- Eine Bereitstellung, die einen Container auf Basis eines MySQL-Image erstellt
- Ein Service, der auf die Anwendung `mysql` verweist

2.1. Wechseln Sie zum Verzeichnis `network-sdn`, und listen Sie die Dateien auf. In einem späteren Schritt initialisieren Sie anhand von `db-data.sql` die Datenbank für die Anwendung.

```
[student@workstation ~]$ cd ~/D0280/labs/network-sdn
[student@workstation network-sdn]$ ls
db-data.sql  todo-db.yaml  todo-frontend.yaml
```

2.2. Führen Sie `oc create` mit dem Argument `-f` für `todo-db.yaml` aus, um den Datenbankserver-Pod bereitzustellen.

```
[student@workstation network-sdn]$ oc create -f todo-db.yaml
deployment.apps/mysql created
service/mysql created
```

2.3. Führen Sie den Befehl `oc status` aus, um die im Projekt vorhandenen Ressourcen zu überprüfen. Der Service `mysql` verweist auf den Datenbank-Pod.

```
[student@workstation network-sdn]$ oc status
In project network-sdn on server https://api.ocp4.example.com:6443

svc/mysql - 172.30.223.41:3306
  deployment/mysql deploys registry.redhat.io/rhel8/mysql-80:1
    deployment #1 running for 4 seconds - 0/1 pods
...output omitted...
```

2.4. Warten Sie einen Moment, um sicherzustellen, dass der Datenbank-Pod ausgeführt wird. Rufen Sie den Namen des Datenbank-Pods ab, um die Tabellen der Datenbank `items` wiederherzustellen.

```
[student@workstation network-sdn]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
mysql-94dc6645b-hjjqb   1/1     Running   0          33m
```

2.5. Führen Sie den Befehl `oc cp` aus, um das Datenbank-Image auf den Pod zu übertragen. Stellen Sie sicher, dass der Pod-Name durch den Namen ersetzt wird, den Sie im vorherigen Schritt abgerufen haben.

```
[student@workstation network-sdn]$ oc cp db-data.sql mysql-94dc6645b-hjjqb:/tmp/
```

- 2.6. Stellen Sie mit `oc rsh` eine Verbindung zum Pod her, und stellen Sie die Datenbank wieder her.

```
[student@workstation network-sdn]$ oc rsh mysql-94dc6645b-hjjqb bash  
bash-4.4$ mysql -u root items < /tmp/db-data.sql
```

- 2.7. Vergewissern Sie sich, dass die Tabelle `Item` in der Datenbank vorhanden ist.

```
bash-4.4$ mysql -u root items -e "show tables;"  
+-----+  
| Tables_in_items |  
+-----+  
| Item |  
+-----+
```

- 2.8. Beenden Sie den Container.

```
bash-4.4$ exit  
exit
```

- 3. Stellen Sie die Front-End-Anwendung bereit. In der Datei `/home/student/D0280/labs/network-sdn/todo-frontend.yaml` sind die folgenden Ressourcen definiert:

- Eine Bereitstellung, die die Node.js-Anwendung `Todo` erstellt
- Ein Service, der auf die Anwendung `frontend` verweist

- 3.1. Erstellen Sie die Front-End-Anwendung mit dem Befehl `oc create`.

```
[student@workstation network-sdn]$ oc create -f todo-frontend.yaml  
deployment.apps/frontend created  
service/frontend created
```

- 3.2. Warten Sie einen Moment, bis der Front-End-Container gestartet wurde, und führen Sie dann den Befehl `oc get pods` aus.

```
[student@workstation network-sdn]$ oc get pods  
NAME READY STATUS RESTARTS AGE  
frontend-57b8b445df-f56qh 1/1 Running 0 34s  
...output omitted...
```

- 4. Erstellen Sie eine Route für den Zugriff auf den Service `frontend`, und greifen Sie auf die Anwendung zu.

- 4.1. Sie müssen eine Route erstellen, um über ein externes Netzwerk auf die Anwendung zuzugreifen. Führen Sie den Befehl `oc expose` für den Service `frontend` aus, um diese Route zu erstellen. Überschreiben Sie mit der Option `--hostname` den standardmäßigen, von OpenShift erstellten FQDN.

```
[student@workstation network-sdn]$ oc expose service frontend \  
> --hostname todo.apps.ocp4.example.com  
route.route.openshift.io/frontend exposed
```

- 4.2. Listen Sie die Routen in dem Projekt auf.

```
[student@workstation network-sdn]$ oc get routes
NAME      HOST/PORT          PATH  SERVICES  PORT  ...
frontend  todo.apps.ocp4.example.com  frontend  8080  ...
```

Wie Sie im Beispiel sehen, erkennt OpenShift den von der Anwendung überwachten Port und erstellt eine Weiterleitungsregel von Port 80 zum Zielport 8080.

- 4.3. Öffnen Sie auf workstation Firefox, und wechseln Sie zu `http://todo.apps.ocp4.example.com/todo/`. Stellen Sie sicher, dass der abschließende Schrägstrich am Ende der URL hinzugefügt wird.

Wie in dem folgenden Screenshot dargestellt, ist die Anwendung nicht erreichbar.

Application is not available

The application is currently not serving requests at this endpoint. It may not have been started or is still starting.

ⓘ Possible reasons you are seeing this page:

- **The host doesn't exist.** Make sure the hostname was typed correctly and that a route matching this hostname exists.
- **The host exists, but doesn't have a matching path.** Check if the URL path was typed correctly and that the route was created using the desired path.
- **Route and path matches, but all pods are down.** Make sure that the resources exposed by this route (pods, services, deployment configs, etc) have at least one pod running.

- 4.4. Überprüfen Sie die Pod-Protokolle auf Fehler. In der Ausgabe sind keine Fehler angegeben.

```
[student@workstation network-sdn]$ oc logs frontend-57b8b445df-f56qh
App is ready at : 8080
```

- ▶ 5. Führen Sie `oc debug` aus, um eine Kopie eines vorhandenen Pods in der Bereitstellung `frontend` zu erstellen. Sie verwenden diesen Pod, um die Konnektivität zur Datenbank zu überprüfen.

- 5.1. Rufen Sie vor dem Erstellen eines Debug-Pods die IP-Adresse des Datenbankservice ab. In einem späteren Schritt verwenden Sie den Befehl `curl`, um auf den Datenbankendpunkt zuzugreifen.

Mit dem JSONPath-Ausdruck können Sie die IP-Adresse des Service abrufen.

```
[student@workstation network-sdn]$ oc get service/mysql \
>   -o jsonpath="{.spec.clusterIP}{'\n'}"
172.30.103.29
```

- 5.2. Führen Sie den Befehl `oc debug` für die Bereitstellung `frontend` aus, auf dem der Webanwendungs-Pod ausgeführt wird.

```
[student@workstation network-sdn]$ oc debug -t deployment/frontend
Starting pod/frontend-debug ...
Pod IP: 10.131.0.144
If you don't see a command prompt, try pressing enter.
sh-4.4$
```

- 5.3. Sie können die Konnektivität zwischen der Bereitstellung `frontend` und der Datenbank beispielsweise mit dem Befehl `curl` testen, der eine Vielzahl von Protokollen unterstützt.
Verwenden Sie `curl`, um sich über Port 3306 mit der Datenbank zu verbinden, dem MySQL-Standardport. Ersetzen Sie dabei die IP-Adresse durch die IP-Adresse, die Sie zuvor für den `mysql`-Service abgerufen haben. Drücken Sie anschließend `Strg+C`, um die Sitzung zu beenden, und geben Sie `exit` ein, um den Debug-Pod zu verlassen.

```
sh-4.4$ curl -v telnet://172.30.103.29:3306
* About to connect() to 172.30.103.29 port 3306 (#0)
*   Trying 172.30.103.29...
* Connected to 172.30.103.29 (172.30.103.29) port 3306 (#0)
J
8.0.21
* RCVD IAC 2
* RCVD IAC 199
^C
sh-4.4$ exit
exit

Removing debug pod ...
```

In der Ausgabe ist angegeben, dass die Datenbank ausgeführt wird und von der Bereitstellung `frontend` darauf zugegriffen werden kann.

- 6. In den folgenden Schritten überprüfen Sie, ob die Netzwerkkonnektivität im Cluster funktioniert, indem Sie vom Datenbankcontainer eine Verbindung zum Front-End-Container herstellen.
Rufen Sie einige Informationen über den Pod `frontend` ab und diagnostizieren Sie mit dem Befehl `oc debug` das Problem in der Bereitstellung `mysql`.
- 6.1. Rufen Sie die IP-Adresse des Service `frontend` ab, bevor Sie einen Debug-Pod erstellen.

```
[student@workstation network-sdn]$ oc get service/frontend \
>   -o jsonpath=".spec.clusterIP}{'\n'}"
172.30.23.147
```

- 6.2. Führen Sie den Befehl `oc debug` aus, um einen Container für die Fehlerbehebung auf Grundlage der Bereitstellung `mysql` zu erstellen. Sie müssen das Container-Image überschreiben, da das MySQL Server-Image den Befehl `curl` nicht bereitstellt.

```
[student@workstation network-sdn]$ oc debug -t deployment/mysql \
>   --image registry.access.redhat.com/ubi8/ubi:8.4
Starting pod/mysql-debug ...
Pod IP: 10.131.0.146
If you don't see a command prompt, try pressing enter.
sh-4.4$
```

- 6.3. Stellen Sie mit `curl` über den Port 8080 eine Verbindung zur Anwendung `frontend` her. Ersetzen Sie dabei die IP-Adresse durch die IP-Adresse, die Sie zuvor für den `frontend`-Service abgerufen haben.

In der folgenden Ausgabe sehen Sie, dass in `curl` eine Zeitüberschreitung auftritt. Dies deutet entweder darauf hin, dass die Anwendung nicht ausgeführt wird oder dass der Service nicht auf sie zugreifen kann.

```
sh-4.4$ curl -m 10 -v http://172.30.23.147:8080
* Rebuilt URL to: http://172.30.23.147:8080/
*   Trying 172.30.23.147...
* TCP_NODELAY set
* Connection timed out after 10000 milliseconds
* Closing connection 0
curl: (28) Connection timed out after 10000 milliseconds
```

- 6.4. Beenden Sie den Debug-Pod.

```
sh-4.4$ exit
exit

Removing debug pod ...
```

- 7. In den folgenden Schritten stellen Sie über die private IP-Adresse eine Verbindung zum Pod `frontend` her. Auf diese Weise können Sie herausfinden, ob das Problem mit dem Service zusammenhängt.

- 7.1. Rufen Sie die IP-Adresse des Pods `frontend` ab.

```
[student@workstation network-sdn]$ oc get pods -o wide -l name=frontend
NAME                  READY   STATUS    RESTARTS   AGE     IP           ...
frontend-57b8b445df-f56qh   1/1     Running   0          39m   10.128.2.61   ...
```

- 7.2. Erstellen Sie aus der Bereitstellung `mysql` einen Debug-Pod.

```
[student@workstation network-sdn]$ oc debug -t deployment/mysql \
>   --image registry.access.redhat.com/ubi8/ubi:8.4
Starting pod/mysql-debug ...
Pod IP: 10.131.1.27
If you don't see a command prompt, try pressing enter.
sh-4.4$
```

- 7.3. Führen Sie `curl` im Verbose-Modus für den Pod `frontend` auf Port 8080 aus. Ersetzen Sie dabei die IP-Adresse durch die IP-Adresse, die Sie zuvor für den `frontend`-Pod abgerufen haben.

```
sh-4.4$ curl -v http://10.128.2.61:8080/todo/
* Trying 10.128.2.61...
* TCP_NODELAY set
* Connected to 10.128.2.61 (10.128.2.61) port 8080 (#0)
> GET /todo/ HTTP/1.1
> Host: 10.128.2.61:8080
> User-Agent: curl/7.61.1
> Accept: */*
>
< HTTP/1.1 200 OK
...output omitted...
```

Der Befehl `curl` kann über die private IP-Adresse des Pods auf die Anwendung zugreifen.

- 7.4. Beenden Sie den Debug-Pod.

```
sh-4.2$ exit
exit

Removing debug pod ...
```

► 8. Überprüfen Sie die Konfiguration des Service `frontend`.

- 8.1. Listen Sie die Services im Projekt auf, und vergewissern Sie sich, dass der Service `frontend` vorhanden ist.

```
[student@workstation network-sdn]$ oc get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
frontend   ClusterIP  172.30.23.147  <none>          8080/TCP    93m
mysql     ClusterIP  172.30.103.29   <none>          3306/TCP    93m
```

- 8.2. Überprüfen Sie die Konfiguration und den Status des Service `frontend`. Beachten Sie den Wert des Service-Selektors, der angibt, an welchen Pod der Service Pakete weiterleiten soll.

```
[student@workstation network-sdn]$ oc describe svc/frontend
Name:           frontend
Namespace:      network-sdn
Labels:         app=todonodejs
                name=frontend
Annotations:    <none>
Selector:       name=api
Type:           ClusterIP
IP:             172.30.23.147
Port:           <unset>  8080/TCP
TargetPort:     8080/TCP
Endpoints:      <none>
Session Affinity: None
Events:         <none>
```

Diese Ausgabe gibt auch an, dass der Service keinen Endpunkt aufweist, sodass der eingehende Datenverkehr nicht an die Anwendung weitergeleitet werden kann.

- 8.3. Rufen Sie die Labels der Bereitstellung `frontend` ab. In der Ausgabe sehen Sie, dass Pods mit dem Label `name` und dem Wert `frontender` erstellt werden, während der Service aus dem vorherigen Schritt den Wert `api` verwendet.

```
[student@workstation network-sdn]$ oc describe deployment/frontend | \
>   grep Labels -A1
Labels:           app=todonodejs
                  name=frontend
--
Labels:  app=todonodejs
        name=frontend
```

- 9. Aktualisieren Sie den Service `frontend`, und greifen Sie auf die Anwendung zu.

- 9.1. Führen Sie den Befehl `oc edit` aus, um den Service `frontend` zu bearbeiten. Aktualisieren Sie den Selektor so, dass er mit dem richtigen Label übereinstimmt.

```
[student@workstation network-sdn]$ oc edit svc/frontend
```

Suchen Sie den Abschnitt, in dem der Selektor definiert wird, und aktualisieren Sie dann im Selektor das Label `name: frontend`. Wenn Sie die Änderungen vorgenommen haben, beenden Sie den Editor.

```
...output omitted...
selector:
  name: frontend
...output omitted...
```

Speichern Sie Ihre Änderungen, und verifizieren Sie, dass die Änderungen durch den Befehl `oc edit` angewendet wurden.

```
service/frontend edited
```

- 9.2. Überprüfen Sie die Servicekonfiguration, um sicherzustellen, dass der Service über einen Endpunkt verfügt.

```
[student@workstation network-sdn]$ oc describe svc/frontend
Name:           frontend
Namespace:      network-sdn
Labels:         app=todonodejs
                name=frontend
Annotations:   <none>
Selector:      name=frontend
Type:          ClusterIP
IP:            172.30.169.113
Port:          <unset>  8080/TCP
TargetPort:    8080/TCP
Endpoints:    10.128.2.61:8080
Session Affinity: None
Events:        <none>
```

- 9.3. Öffnen Sie Firefox auf dem Rechner `workstation` und wechseln Sie zur Anwendung „To Do“ unter <http://todo.apps.ocp4.example.com/todo/>.

Die Anwendung „To Do“ sollte angezeigt werden.

To Do List Application

To Do List

Id	Description	Done	
1	Pick up newsp...	false	X
2	Buy groceries	true	X

First Previous **1** Next Last

Add Task

Description:

Completed:

Clear Save

- 10. Wechseln Sie zum Benutzerverzeichnis, und löschen Sie das Projekt network - sdn.

```
[student@workstation network-sdn]$ cd  
[student@workstation ~]$ oc delete project network-sdn  
project.project.openshift.io "network-sdn" deleted
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab network-sdn finish
```

Hiermit ist die angeleitete Übung beendet.

Anwendungen für den externen Zugriff bereitstellen

Ziele

Nach diesem Abschnitt sollten Sie Netzwerkverbindungen zu Anwendungen in einem OpenShift-Cluster erlauben und absichern können.

Zugreifen auf Anwendungen aus externen Netzwerken

OpenShift Container Platform bietet viele Möglichkeiten, Ihre Anwendungen für externe Netzwerke bereitzustellen. Sie können HTTP- und HTTPS-Datenverkehr, TCP-Anwendungen sowie Nicht-TCP-Datenverkehr bereitstellen. Einige dieser Methoden sind *Servicetypen*, z. B. NodePort oder Load Balancer, während andere Methoden eigene API-Ressourcen verwenden, z. B. Ingress und Route.

Mit OpenShift-Routen können Sie Ihre Anwendungen für externe Netzwerke bereitzustellen.

Mit Routen können Sie auf Ihre Anwendung mit einem eindeutigen Hostnamen zugreifen, der öffentlich zugänglich ist. Routen basieren auf einem Router-Plugin, um den Datenverkehr von der öffentlichen IP-Adresse zu Pods umzuleiten.

Das folgende Diagramm zeigt, wie eine Anwendung, die als Pods in Ihrem Cluster ausgeführt wird, durch eine Route bereitgestellt wird:

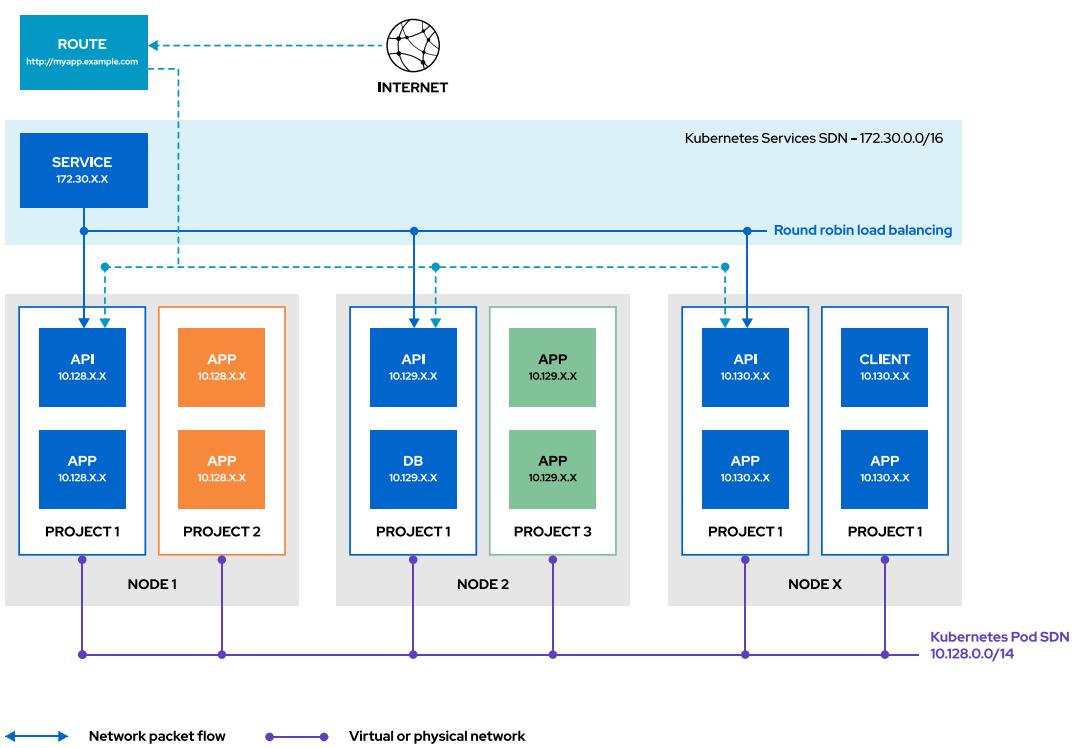


Abbildung 5.5: Bereitstellen von Anwendungen anhand von Routen



Anmerkung

Aus Leistungsgründen senden Router Anfragen entsprechend der Servicekonfiguration direkt an Pods.

Die gepunktete Zeile zeigt diese Implementierung an. Der Router greift also über das Services-Netzwerk auf die Pods zu.

Beschreiben von Methoden zur Verwaltung des eingehenden Datenverkehrs

Der Ingress-Controller ist die gängigste Methode zur Verwaltung des eingehenden Datenverkehrs. OpenShift implementiert den Ingress-Controller mit einem gemeinsam genutzten Router-Service, der als Pod im Cluster ausgeführt wird. Dieser Pod kann wie jeder andere normale Pod skaliert und repliziert werden. Dieser Routerservice basiert auf der Open Source-Software HAProxy.

Routen und eingehende Daten sind die wichtigsten Ressourcen für die Verarbeitung des eingehenden Datenverkehrs:

Route

Routen leiten den eingehenden Datenverkehr an Services im Cluster weiter. Routen wurden vor Ingress-Objekten in Kubernetes erstellt und bieten mehr Funktionen. Routen bieten erweiterte Funktionen, die von Kubernetes-Ingress-Controllern unter Umständen nicht über eine Standardschnittstelle unterstützt werden, z. B. erneute TLS-Verschlüsselung, TLS-Passthrough und Split-Datenverkehr für Blue-Green-Bereitstellungen.

Ingress

Bei einem Ingress handelt es sich um eine Kubernetes-Ressource, die einige der gleichen Funktionen wie Routen (eine OpenShift-Ressource) bereitstellt. Ingress-Objekte akzeptieren externe Anforderungen und leiten sie entsprechend der Route weiter. Dabei sind nur bestimmte Datenverkehrstypen zulässig: HTTP, HTTPS, Server Name Identification (SNI) und TLS mit SNI. In OpenShift werden Routen erstellt, um die vom Ingress-Objekt vorgegebenen Bedingungen zu erfüllen.

Es gibt Alternativen zu Ingress und Routen, die jedoch nur für spezielle Anwendungsfälle geeignet sind. Die folgenden Servicetypen bieten externen Zugriff auf Services:

Externer Load Balancer

Diese Ressource weist OpenShift an, einen Load Balancer in einer Cloud-Umgebung zu starten. Ein Load Balancer weist OpenShift an, mit dem Cloud-Anbieter zu interagieren, in dem der Cluster ausgeführt wird, um einen Load Balancer bereitzustellen.

Externe Service-IP

Diese Methode weist OpenShift an, NAT-Regeln so einzurichten, dass der Datenverkehr von einer der Cluster-IP-Adressen an den Container umgeleitet wird.

NodePort

Mit dieser Methode stellt OpenShift einen Service auf einem statischen Port auf der IP-Adresse des Knotens bereit. Sie müssen sicherstellen, dass die externen IP-Adressen ordnungsgemäß an die Knoten weitergeleitet werden.

Erstellen von Routen

Die einfachste und bevorzugte Methode zum Erstellen einer Route (gesichert oder ungesichert) besteht darin, den Befehl `oc expose service service` zu verwenden, wobei `service` einem

Service entspricht. Mit der Option `--hostname` können Sie einen benutzerdefinierten Hostnamen für die Route angeben.

```
[user@host ~]$ oc expose service api-frontend \
>   --hostname api.apps.acme.com
```

Wenn Sie den Hostnamen weglassen, generiert OpenShift einen Hostnamen für Sie mit der folgenden Struktur: <Routename>-<Projektname>. <Standard-Domain> Beispiel: Wenn Sie eine `frontend`-Route in einem `api`-Projekt in einem Cluster erstellen, der `apps.example.com` als Platzhalter-Domain verwendet, ist der Hostname der Route:

```
frontend.api.apps.example.com.
```

Wichtig

Dem DNS-Server, auf dem die Platzhalter-Domain gehostet wird, sind keine Routen-Hostnamen bekannt; er löst nur die Namen in die konfigurierten IPs auf. Nur der OpenShift-Router kennt die Routing-Hostnamen und behandelt jeden Namen wie einen virtuellen HTTP-Host.

Ungültige Hostnamen von Platzhalter-Domains – d. h. Hostnamen, die keiner Route entsprechen – werden vom OpenShift-Router blockiert und erzeugen die Fehlermeldung „HTTP 404“.

Berücksichtigen Sie beim Erstellen von Routen die folgenden Einstellungen:

- Den Namen des Service. Die Route verwendet den Service, um die Pods zu bestimmen, an die der Datenverkehr weitergeleitet werden soll.
- Einen Hostname für die Route. Eine Route ist immer eine Subdomain Ihrer Cluster-Platzhalter-Domain. Wenn Sie z. B. die Platzhalter-Domain `apps.dev-cluster.acme.com` verwenden und einen Front-End-Service über eine Route bereitstellen müssen, dann erhält sie folgenden Namen:

```
frontend.apps.dev-cluster.acme.com
```



Anmerkung

Sie können auch einen Hostnamen für die Route von OpenShift automatisch generieren lassen.

- Einen optionalen Pfad für pfadbasierte Routen.
- Einen Zielport, den die Anwendung überwacht. Der Zielport entspricht in der Regel dem Port, den Sie im Schlüssel `targetPort` eines Service definieren.
- Eine Verschlüsselungsstrategie, je nachdem, ob Sie eine gesicherte oder eine ungesicherte Route benötigen.

Die folgende Auflistung zeigt eine Minimaldefinition für eine Route:

```
kind: Route
apiVersion: route.openshift.io/v1
metadata:
  name: a-simple-route 1
  labels: 2
    app: API
    name: api-frontend
spec:
  host: api.apps.acme.com 3
  to:
    kind: Service
    name: api-frontend 4
    port: 5
      targetPort: 8443
```

- 1** Der Name der Route. Dieser Name muss eindeutig sein.
- 2** Eine Reihe von Labels, die Sie als Selektoren verwenden können.
- 3** Der Hostname der Route. Dieser Hostname muss eine Subdomain Ihrer Platzhalter-Domain sein, da OpenShift die Platzhalter-Domain an die Router weiterleitet.
- 4** Der Service, an den der Datenverkehr umgeleitet werden soll. Obwohl Sie einen Servicenamen verwenden, nutzt die Route diese Informationen nur, um die Liste der Pods zu ermitteln, die den Datenverkehr empfangen.
- 5** Der Anwendungs-Port. Da Routen Services umgehen, muss diese Angabe mit dem Anwendungs-Port und nicht mit dem Service-Port übereinstimmen.

Sichern von Routen

Routen können gesichert oder ungesichert sein. Gesicherte Routen bietet die Möglichkeit, mehrere Arten der TLS-Terminierung für die Bereitstellung von Zertifikaten an den Client zu verwenden. Ungesicherte Routen sind am einfachsten zu konfigurieren, da sie keine Schlüssel oder Zertifikate erfordern. Bei gesicherten Routen wird der Datenverkehr zu und von den Pods jedoch verschlüsselt.

Bei einer gesicherten Route wird die TLS-Terminierung der Route angegeben. Die verfügbaren Terminierungsarten sind in der folgenden Liste aufgeführt:

Gesicherte OpenShift-Routen

Edge

Bei der Edge-Terminierung erfolgt die TLS-Terminierung am Router, bevor der Datenverkehr an die Pods geleitet wird. Der Router stellt die TLS-Zertifikate bereit, daher müssen Sie sie in der Route konfigurieren. Andernfalls weist OpenShift dem Router ein eigenes Zertifikat für die TLS-Terminierung zu. Da die TLS-Terminierung am Router erfolgt, werden die Verbindungen vom Router zu den Endpunkten über das interne Netzwerk nicht verschlüsselt.

Passthrough

Bei der Passthrough-Terminierung wird der verschlüsselte Datenverkehr direkt an den Ziel-Pod gesendet, ohne dass der Router die TLS-Terminierung bereitstellt. In diesem Modus ist die Anwendung für die Bereitstellung von Zertifikaten für den Datenverkehr zuständig. Passthrough ist derzeit die einzige Methode, die die gegenseitige Authentifizierung zwischen der Anwendung und einem Client unterstützt, der auf sie zugreift.

Wiederverschlüsselung

Die Wiederverschlüsselung ist eine Variation der Edge-Terminierung, bei der der Router TLS mit einem Zertifikat terminiert und dann die Verbindung zum Endpunkt erneut verschlüsselt, der u. U. ein anderes Zertifikat verwendet. Aus diesem Grund wird der vollständige Verbindungspfad verschlüsselt, selbst über das interne Netzwerk. Der Router ermittelt anhand von Health Checks die Authentizität des Hosts.

Sichern von Anwendungen mit Edge-Routen

Vor dem Erstellen einer gesicherten Route müssen Sie ein TLS-Zertifikat erzeugen. Der folgende Befehl zeigt, wie eine sichere Edge-Route mit einem TLS-Zertifikat erstellt wird:

```
[user@host ~]$ oc create route edge \
>   --service api-frontend --hostname api.apps.acme.com \
>   --key api.key --cert api.crt
```

Für die Option `--key` ist der Private Key des Zertifikats und für die Option `--cert` das Zertifikat erforderlich, das mit diesem Schlüssel signiert wurde.

Wenn Sie eine Route im Edge-Modus verwenden, wird der Datenverkehr zwischen dem Client und dem Router verschlüsselt, aber der Datenverkehr zwischen dem Router und der Anwendung nicht:

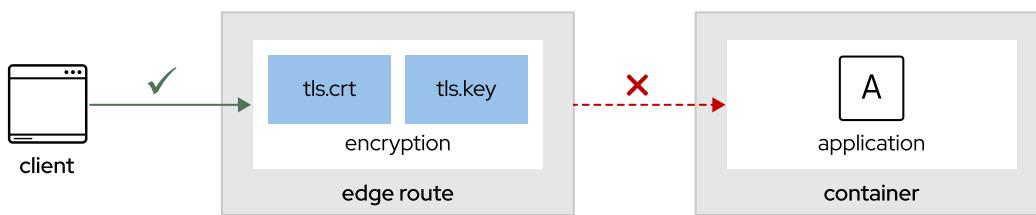


Abbildung 5.6: Sichern von Anwendungen mit Edge-Routen



Anmerkung

Mit Netzwerkrichtlinien können Sie den internen Datenverkehr zwischen Ihren Anwendungen oder zwischen Projekten sichern. Weitere Informationen dazu finden Sie im Dokument *Network Policy Objects in Action* im Abschnitt „References“.

Sichern von Anwendungen mit Passthrough-Routen

Im vorherigen Beispiel wird veranschaulicht, wie eine Edge-Route erstellt wird, d. h. eine OpenShift-Route, die am Edge ein Zertifikat präsentiert. Passthrough-Routen sind eine sichere Alternative, da die Anwendung ihr TLS-Zertifikat bereitstellt. Daher wird der Datenverkehr zwischen dem Client und der Anwendung verschlüsselt.

Um eine Passthrough-Route zu erstellen, benötigen Sie ein Zertifikat und eine Möglichkeit für Ihre Anwendung, darauf zuzugreifen. Der beste Weg, dies zu erreichen, ist die Verwendung von OpenShift-TLS-Secrets. Secrets werden dem Container über einen Mount-Punkt zur Verfügung gestellt.

Das folgende Diagramm zeigt, wie Sie eine `secret`-Ressource in Ihrem Container mounten können. Die Anwendung kann dann auf Ihr Zertifikat zugreifen. In diesem Modus gibt es keine Verschlüsselung zwischen dem Client und dem Router.

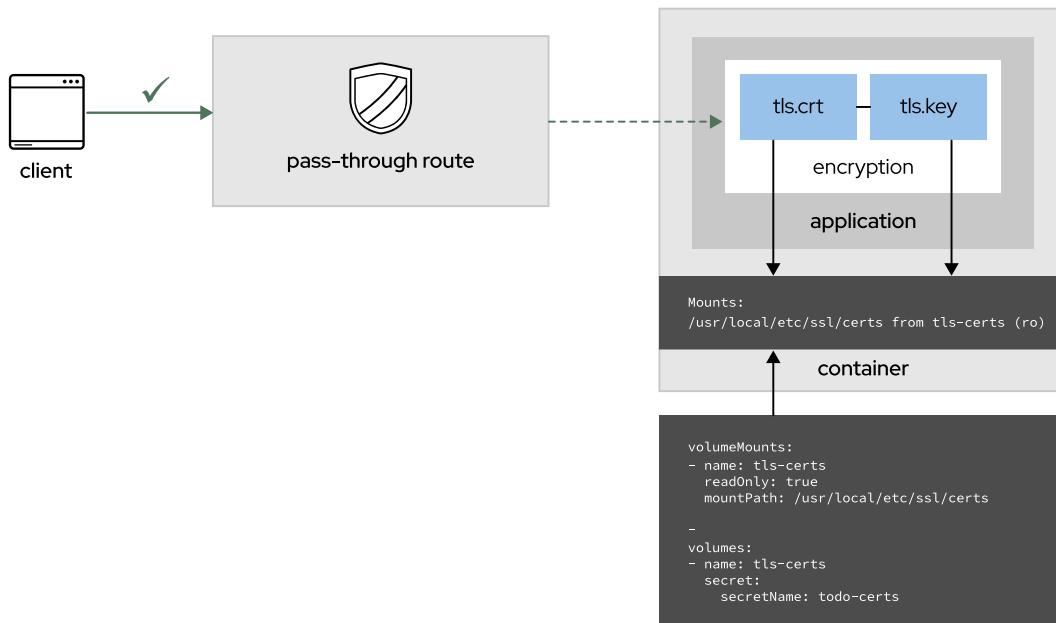


Abbildung 5.7: Sichern von Anwendungen mit Passthrough-Routen



Literaturhinweise

Weitere Informationen zum Verwalten von Routen finden Sie im Kapitel *Configuring Routes* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Networking* unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/networking/index#configuring-routes

Weitere Informationen zum Konfigurieren von Ingress-Cluster-Datenverkehr finden Sie im Kapitel *Configuring ingress cluster traffic* in der Dokumentation zu „Red Hat OpenShift Container Platform 4.6 Networking“ unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/networking/index#configuring-ingress-cluster-traffic

Routen

https://docs.openshift.com/online/pro/dev_guide/routes.html

Kubernetes Ingress vs OpenShift Route – OpenShift Blog

<https://blog.openshift.com/kubernetes-ingress-vs-openshift-route/>

Network Policy Objects in Action – OpenShift Blog

<https://blog.openshift.com/network-policy-objects-action/>

► Angeleitete Übung

Anwendungen für den externen Zugriff bereitstellen

In dieser Übung stellen Sie eine mit TLS-Zertifikaten gesicherte Anwendung bereit.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Bereitstellen einer Anwendung und Erstellen einer unverschlüsselten Route
- Erstellen einer OpenShift-Edge-Route mit Verschlüsselung
- Aktualisieren einer OpenShift-Bereitstellung zur Unterstützung einer neuen Version der Anwendung
- Erstellen eines OpenShift-TLS-Secret und Mounten Ihrer Anwendung
- Verifizieren, dass die Kommunikation mit der Anwendung verschlüsselt ist

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl stellt sicher, dass die Cluster-API erreichbar ist und das OpenShift-Projekt `network-ingress` erstellt wird. Außerdem erteilt dieser Befehl dem Benutzer `developer` Zugriff auf das Projekt.

```
[student@workstation ~]$ lab network-ingress start
```

Anweisungen

Als Anwendungsentwickler können Sie nun Ihre Anwendung in OpenShift bereitstellen. In dieser Übung stellen Sie zwei Versionen der Anwendung bereit: Eine, die über unverschlüsselten Datenverkehr (HTTP), und eine, die über gesicherten Datenverkehr bereitgestellt wird.

Das Container-Image unter <https://quay.io/redhattraining/todo-angular> verfügt über zwei Tags: `v1.1`, die ungesicherte Version der Anwendung, und `v1.2`, die gesicherte Version. Ihr Unternehmen verwendet eine eigene Zertifizierungsstelle (Certificate Authority, CA), die Zertifikate für die Domains `*.apps.ocp4.example.com` und `*.ocp4.example.com` signiert.

Auf das CA-Zertifikat kann unter `~/D0280/labs/network-ingress/certs/training-CA.pem` zugegriffen werden. Die Datei `passphrase.txt` enthält ein eindeutiges Passwort zum Schutz des CA-Schlüssels. Der Ordner `certs` enthält auch den CA-Schlüssel.

- 1. Melden Sie sich bei dem OpenShift-Cluster an, und erstellen Sie das Projekt `network-ingress`.
 - 1.1. Melden Sie sich als Benutzer `developer` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \
>   https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

1.2. Erstellen Sie das Projekt `network-ingress`.

```
[student@workstation ~]$ oc new-project network-ingress
Now using project "network-ingress" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

- 2. Die OpenShift-Bereitstellungsdatei für die Anwendung befindet sich unter `~/D0280/labs/network-ingress/todo-app-v1.yaml`. Die Bereitstellung verweist auf `quay.io/redhattraining/todo-angular:v1.1`. Hierbei handelt es sich um die anfängliche und unverschlüsselte Version der Anwendung. In der Datei ist der Service `todo-http` definiert, der auf den Anwendungs-Pod verweist.

Erstellen Sie die Anwendung, und stellen Sie den Service bereit.

- 2.1. Stellen Sie die Anwendung mit dem Befehl `oc create` im OpenShift-Projekt `network-ingress` bereit.

```
[student@workstation ~]$ oc create -f \
>   ~/D0280/labs/network-ingress/todo-app-v1.yaml
deployment.apps/todo-http created
service/todo-http created
```

- 2.2. Warten Sie ein paar Minuten, bis die Anwendung gestartet ist, und überprüfen Sie dann die Ressourcen im Projekt.

```
[student@workstation ~]$ oc status
In project network-ingress on server https://api.ocp4.example.com:6443

svc/todo-http - 172.30.247.75:80 -> 8080
  deployment/todo-http deploys quay.io/redhattraining/todo-angular:v1.1
    deployment #1 running for 16 seconds - 1 pod
...output omitted...
```

- 2.3. Erstellen Sie mit dem Befehl `oc expose` eine Route für den Zugriff auf die Anwendung. Geben Sie der Route den Hostnamen `todo-http.apps.ocp4.example.com`.

```
[student@workstation ~]$ oc expose svc todo-http \
>   --hostname todo-http.apps.ocp4.example.com
route.route.openshift.io/todo-http exposed
```

- 2.4. Rufen Sie den Namen der Route ab, und kopieren Sie ihn in die Zwischenablage.

```
[student@workstation ~]$ oc get routes
NAME      HOST/PORT          PATH  SERVICES  PORT  ...
todo-http  todo-http.apps.ocp4.example.com        todo-http  8080  ...
```

- 2.5. Öffnen Sie Firefox auf dem Rechner `workstation`, und öffnen Sie `http://todo-http.apps.ocp4.example.com`.
Vergewissern Sie sich, dass die Anwendung angezeigt wird.
- 2.6. Öffnen Sie eine neue Terminal-Registerkarte, und führen Sie `tcpdump` mit den folgenden Optionen aus, um den Datenverkehr an Port 80 abzufangen:
- `-i eth0` fängt den Datenverkehr an der Hauptschnittstelle ab.
 - `-A` entfernt die Header und gibt die Pakete im ASCII-Format aus.
 - `-n` deaktiviert die DNS-Auflösung.
 - `port 80` ist der Port der Anwendung.

Optional können Sie mit dem Befehl `grep` nach JavaScript-Ressourcen filtern.

Beginnen Sie damit, den Namen der Hauptschnittstelle abzurufen. Die IP-Adresse dieser Schnittstelle lautet `172.25.250.9`.

```
[student@workstation ~]$ ip a | grep 172.25.250.9
inet 172.25.250.9/24 brd 172.25.250.255 scope global noprefixroute eth0
[student@workstation ~]$ sudo tcpdump -i eth0 -A \
>     -n port 80 | grep js
```



Anmerkung

Der vollständige Befehl ist unter `~/D0280/labs/network-ingress/tcpdump-command.txt` verfügbar.

- 2.7. Aktualisieren Sie auf Firefox die Seite, und beachten Sie die Aktivität im Terminal.
Drücken Sie `Strg+C`, um das Abfangen zu stoppen.

```
...output omitted...
    toBe('Pretty text with some links: http://angularjs.org/',
us@somewhere.org, ' +
    toBe('Pretty text with some links: http://angularjs.org/',
mailto:us@somewhere.org, ' +
        toBe('http://angularjs.org/');
...output omitted...
/*jshint validthis: true */
/*jshint validthis: true */
...output omitted...
```

- 3. Erstellen Sie eine gesicherte Edge-Route. Edge-Zertifikate verschlüsseln den Datenverkehr zwischen dem Client und dem Router, der Datenverkehr zwischen dem Router und dem Service bleibt jedoch unverschlüsselt. OpenShift generiert ein eigenes Zertifikat, das es mit seiner CA signiert.

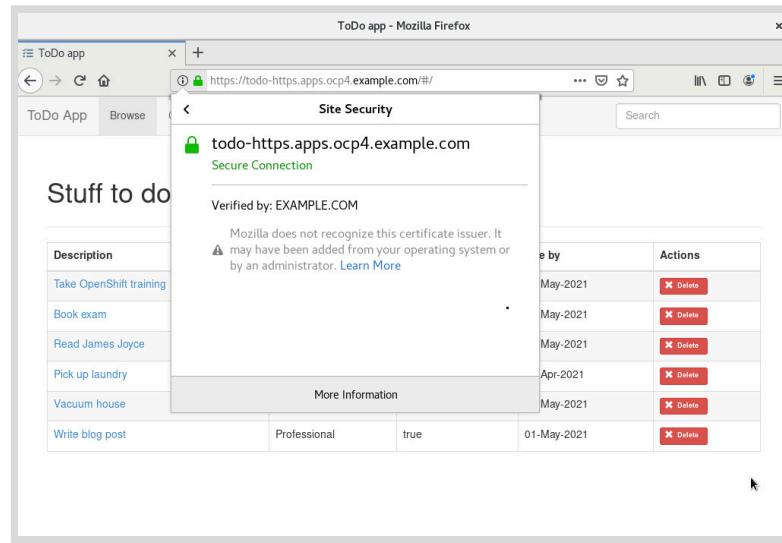
In späteren Schritten extrahieren Sie die CA, um sicherzustellen, dass das Routenzertifikat signiert wird.

- 3.1. Wechseln Sie zu `~/D0280/labs/network-ingress`, und führen Sie den Befehl `oc create route` aus, um die neue Route zu definieren.

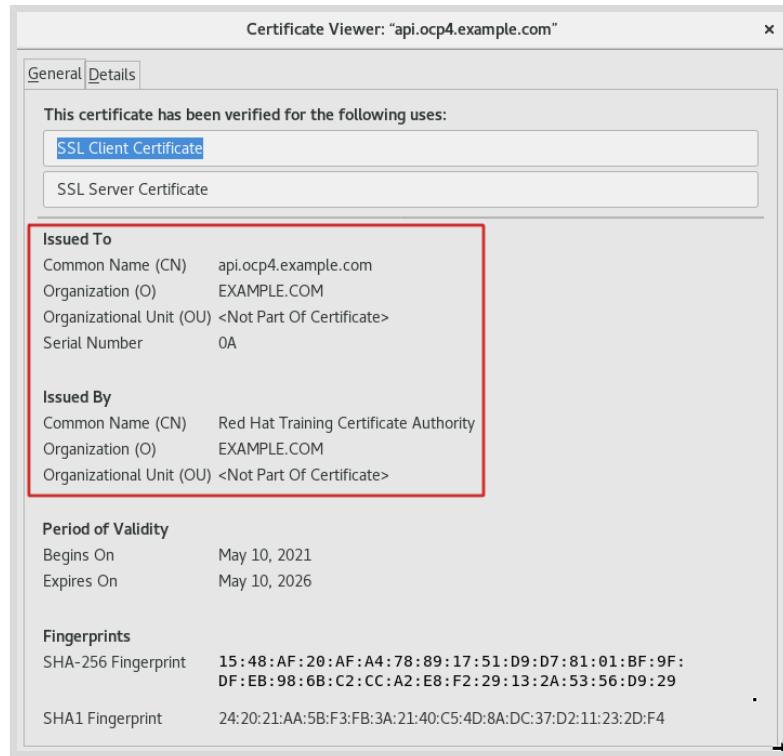
Geben Sie der Route den Hostnamen `todo-http.apps.ocp4.example.com`.

```
[student@workstation ~]$ cd ~/DO280/labs/network-ingress
[student@workstation network-ingress]$ oc create route edge todo-https \
>   --service todo-http \
>   --hostname todo-https.apps.ocp4.example.com
route.route.openshift.io/todo-https created
```

- 3.2. Öffnen Sie Firefox und <https://todo-https.apps.ocp4.example.com/>, um die Route zu testen und das Zertifikat zu lesen. Klicken Sie auf das grüne Schloss und dann auf den Pfeil neben dem Bereich „Connection“. Klicken Sie auf **More Information**, und klicken Sie auf **View Certificate**, um das Zertifikat anzuzeigen.



Suchen Sie nach dem Eintrag „CN“, um anzuzeigen, dass der OpenShift-Ingress-Operator das Zertifikat mit seiner eigenen Zertifizierungsstelle erstellt hat.



- 3.3. Führen Sie den Befehl `curl` im Terminal mit den Optionen `-I` und `-v` aus, um die Verbindungs-Header abzurufen.

Im Abschnitt `Server certificate` werden einige Informationen über das Zertifikat angezeigt, und der alternative Name stimmt mit dem Namen der Route überein.

```
[student@workstation network-ingress]$ curl -I -v \
> https://todo-https.apps.ocp4.example.com
...output omitted...
* Server certificate:
*   subject: O=EXAMPLE.COM; CN=.api.ocp4.example.com
*   start date: May 10 11:18:41 2021 GMT
*   expire date: May 10 11:18:41 2026 GMT
*   subjectAltName: host "todo-https.apps.ocp4.example.com" matched cert's
"*.apps.ocp4.example.com"
*   issuer: O=EXAMPLE.COM; CN=Red Hat Training Certificate Authority
*   SSL certificate verify ok.
...output omitted...
```

In der Ausgabe sehen Sie, dass dem Remote-Zertifikat vertraut wird, da es mit der CA übereinstimmt.

- 3.4. Obwohl der Datenverkehr am Edge mit einem Zertifikat verschlüsselt ist, können Sie weiterhin auf den ungesicherten Datenverkehr auf der Service-Ebene zugreifen, da der Pod hinter dem Service keine verschlüsselte Route bereitstellt.

Rufen Sie die IP-Adresse des Service `todo-http` ab.

```
[student@workstation network-ingress]$ oc get svc todo-http \
> -o jsonpath=".spec.clusterIP\n\""
172.30.102.29
```

- 3.5. Erstellen Sie in der Bereitstellung todo-http einen Debug-Pod. Verwenden Sie das Red Hat Universal Base Image (UBI), das einige grundlegende Tools für die Interaktion mit Containern enthält.

```
[student@workstation network-ingress]$ oc debug -t deployment/todo-http \
> --image registry.access.redhat.com/ubi8/ubi:8.4
Starting pod/todo-http-debug ...
Pod IP: 10.131.0.255
If you don't see a command prompt, try pressing enter.
sh-4.4$
```

- 3.6. Greifen Sie auf dem Debug-Pod mit dem Befehl curl per HTTP auf den Service zu. Ersetzen Sie dabei die IP-Adresse durch die IP-Adresse, die Sie im vorherigen Schritt abgerufen haben.

In der Ausgabe wird angezeigt, dass die Anwendung über HTTP verfügbar ist.

```
sh-4.4$ curl -v 172.30.102.29
* Rebuilt URL to: 172.30.102.29/
*   Trying 172.30.102.29...
* TCP_NODELAY set
* Connected to 172.30.102.29 (172.30.102.29) port 80 (#0)
> GET / HTTP/1.1
> Host: 172.30.102.29
> User-Agent: curl/7.61.1
> Accept: */*
>
< HTTP/1.1 200 OK
...output omitted...
```

- 3.7. Beenden Sie den Debug-Pod.

```
sh-4.4$ exit
Removing debug pod ...
```

- 3.8. Löschen Sie die Edge-Route. In den nächsten Schritten definieren Sie die Passthrough-Route.

```
[student@workstation network-ingress]$ oc delete route todo-https
route.route.openshift.io "todo-https" deleted
```

► 4. Generieren Sie TLS-Zertifikate für die Anwendung.

In den folgenden Schritten generieren Sie ein von der CA signiertes Zertifikat, das Sie dem Pod als Secret zuordnen. Anschließend konfigurieren Sie eine gesicherte Route im Passthrough-Modus und stellen das Zertifikat durch die Anwendung bereit.

- 4.1. Wechseln Sie zum Verzeichnis `~/D0280/labs/network-ingress/certs`, und listen Sie die Dateien auf.

```
[student@workstation network-ingress]$ cd certs
[student@workstation certs]$ ls -l
total 20
-rw-rw-r--. 1 student student 604 Nov 29 17:35 openssl-commands.txt
-rw-r--r--. 1 student student 33 Nov 29 17:35 passphrase.txt
-rw-r--r--. 1 student student 1743 Nov 29 17:35 training-CA.key
-rw-r--r--. 1 student student 1363 Nov 29 17:35 training-CA.pem
-rw-r--r--. 1 student student 406 Nov 29 17:35 training.ext
```

- 4.2. Generieren Sie den Private Key für Ihr von der CA signiertes Zertifikat.



Anmerkung

Die folgenden Befehle zum Generieren eines signierten Zertifikats sind im Verzeichnis in der Datei `openssl-commands.txt` verfügbar.

```
[student@workstation certs~]$ openssl genrsa -out training.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
```

- 4.3. Generieren Sie die Anforderung für die Signierung des Zertifikats (Certificate Signing Request, CSR) für `todo-https.apps.ocp4.example.com`. Geben Sie den Betreff der Anforderung in einer Zeile ein. Alternativ können Sie die Option `-subj` mitsamt dem Inhalt entfernen. Ohne die Option `-subj` werden Sie vom Befehl `openssl` aufgefordert, die Werte einzugeben. Achten Sie darauf, einen allgemeinen Namen (Common Name, CN) für `todo-https.apps.ocp4.example.com` anzugeben.

```
[student@workstation certs]$ openssl req -new \
> -subj "/C=US/ST=North Carolina/L=Raleigh/O=Red Hat/\
> CN=todo-https.apps.ocp4.example.com" \
> -key training.key -out training.csr
```

- 4.4. Generieren Sie abschließend das signierte Zertifikat. Beachten Sie die Verwendung der Optionen `-CA` und `-CAkey`, um das Zertifikat mit der CA zu signieren. Mit der Option `-passin` können Sie das Passwort der CA wieder verwenden. Mit der Option `extfile` können Sie einen *Alternativen Antragstellernamen (Subject Alternative Name, SAN)* definieren.

```
[student@workstation certs]$ openssl x509 -req -in training.csr \
> -passin file:passphrase.txt \
> -CA training-CA.pem -CAkey training-CA.key -CAcreateserial \
> -out training.crt -days 1825 -sha256 -extfile training.ext
Signature ok
subject=C = US, ST = North Carolina, L = Raleigh, O = Red Hat, CN = todo-
https.apps.ocp4.example.com
Getting CA Private Key
```

- 4.5. Überprüfen Sie, ob das neu erstellte Zertifikat und der Schlüssel im aktuellen Verzeichnis vorhanden sind.

```
[student@workstation certs]$ ls -lrt
total 36
-rw-r--r-- 1 student student 599 Jul 31 09:35 openssl-commands.txt
-rw-r--r-- 1 student student 33 Aug 3 12:38 passphrase.txt
-rw-r--r-- 1 student student 352 Aug 3 12:38 training.ext
-rw----- 1 student student 1743 Aug 3 12:38 training-CA.key
-rw-r--r-- 1 student student 1334 Aug 3 12:38 training-CA.pem
-rw----- 1 student student 1675 Aug 3 13:38 training.key
-rw-rw-r-- 1 student student 1017 Aug 3 13:39 training.csr
-rw-rw-r-- 1 student student 41 Aug 3 13:40 training-CA.srl
-rw-rw-r-- 1 student student 1399 Aug 3 13:40 training.crt
```

- 4.6. Kehren Sie zum Verzeichnis `network-ingress` zurück. Dies ist wichtig, da der nächste Schritt die Erstellung einer Route mit dem selbstsignierten Zertifikat umfasst.

```
[student@workstation certs]$ cd ~/D0280/labs/network-ingress
```

- 5. Stellen Sie eine neue Version Ihrer Anwendung bereit. Die neue Version der Anwendung erwartet ein Zertifikat und einen Schlüssel im Container unter `/usr/local/etc/ssl/certs`. Der Webserver in dieser Version ist mit SSL-Unterstützung konfiguriert. Erstellen Sie ein Secret, um das Zertifikat vom Rechner `workstation` zu importieren. In einem späteren Schritt fordert die Anwendungsbereitstellung das Secret an und stellt den Inhalt des Containers unter `/usr/local/etc/ssl/certs` bereit.
- 5.1. Erstellen Sie ein `tls`-OpenShift-Secret mit dem Namen `todo-certs`. Betten Sie mit den Optionen `--cert` und `--key` die TLS-Zertifikate ein. Verwenden Sie `training.csr` als Zertifikat und `training.key` als Schlüssel.

```
[student@workstation network-ingress]$ oc create secret tls todo-certs \
> --cert certs/training.crt \
> --key certs/training.key
secret/todo-certs created
```

- 5.2. Die Bereitstellungsdatei unter `~/D0280/labs/network-ingress/todo-app-v2.yaml` verweist auf Version 2 des Container-Image. Die neue Version der Anwendung ist so konfiguriert, dass SSL-Zertifikate unterstützt werden. Führen Sie `oc create` aus, um eine neue Bereitstellung mit diesem Image zu erstellen.

```
[student@workstation network-ingress]$ oc create -f todo-app-v2.yaml
deployment.apps/todo-https created
service/todo-https created
```

- 5.3. Warten Sie ein paar Minuten, um sicherzustellen, dass der Anwendungs-Pod ausgeführt wird. Kopieren Sie den Pod-Namen in die Zwischenablage.

```
[student@workstation network-ingress]$ oc get pods
NAME                      READY   STATUS    RESTARTS   AGE
...output omitted...
todo-https-59d8fc9d47-265ds   1/1     Running   0          62s
```

- 5.4. Überprüfen Sie die im Pod gemounteten Volumes. In der Ausgabe ist angegeben, dass die Zertifikate unter /usr/local/etc/ssl/certs gemountet sind.

```
[student@workstation network-ingress]$ oc describe pod \
> todo-https-59d8fc9d47-265ds | grep Mounts -A2
Mounts:
  /usr/local/etc/ssl/certs from tls-certs (ro)
  /var/run/secrets/kubernetes.io/serviceaccount from default-token-gs7gx
(ro)
Conditions:
```

► 6. Erstellen Sie die gesicherte Route.

- 6.1. Führen Sie den Befehl `oc create route` aus, um die neue Route zu definieren. Geben Sie der Route den Hostnamen `todo-https.apps.ocp4.example.com`.

```
[student@workstation network-ingress]$ oc create route passthrough todo-https \
> --service todo-https --port 8443 \
> --hostname todo-https.apps.ocp4.example.com
route.route.openshift.io/todo-https created
```

- 6.2. Verwenden Sie den Befehl `curl` im Verbose-Modus, um die Route zu testen und das Zertifikat zu lesen. Verwenden Sie die Option `--cacert`, um das CA-Zertifikat an den Befehl `curl` zu übergeben.

In der Ausgabe wird eine Übereinstimmung zwischen der Zertifikatskette und dem Anwendungszertifikat angezeigt. Dies bedeutet, dass der OpenShift-Router nur Pakete weiterleitet, die mit dem Zertifikat des Anwendungswebservers verschlüsselt wurden.

```
[student@workstation network-ingress]$ curl -vvI \
> --cacert certs/training-CA.pem \
> https://todo-https.apps.ocp4.example.com
...output omitted...
* Server certificate:
*  subject: C=US; ST=North Carolina; L=Raleigh; O=Red Hat; CN=todo-
https.apps.ocp4.example.com
*  start date: Jun 15 01:53:30 2021 GMT
*  expire date: Jun 14 01:53:30 2026 GMT
```

```
* subjectAltName: host "todo-https.apps.ocp4.example.com" matched cert's
  "*.apps.ocp4.example.com"
* issuer: C=US; ST=North Carolina; L=Raleigh; O=Red Hat; CN=ocp4.example.com
* SSL certificate verify ok.
...output omitted...
```

- 7. Erstellen Sie einen neuen Debug-Pod, um die ordnungsgemäße Verschlüsselung auf Service-Ebene weiter zu überprüfen.

- 7.1. Rufen Sie die IP-Adresse des Service `todo-https` ab.

```
[student@workstation network-ingress]$ oc get svc todo-https \
> -o jsonpath="{.spec.clusterIP}{'\n'}"
172.30.121.154
```

- 7.2. Erstellen Sie in der Bereitstellung `todo-https` mit dem Red Hat UBI einen Debug-Pod.

```
[student@workstation network-ingress]$ oc debug -t deployment/todo-https \
> --image registry.access.redhat.com/ubi8/ubi:8.4
Starting pod/todo-https-debug ...
Pod IP: 10.128.2.129
If you don't see a command prompt, try pressing enter.
sh-4.4$
```

- 7.3. Greifen Sie auf dem Debug-Pod mit dem Befehl `curl` per HTTP auf den Service zu. Ersetzen Sie dabei die IP-Adresse durch die IP-Adresse, die Sie im vorherigen Schritt abgerufen haben.

In der Ausgabe wird angezeigt, dass die Anwendung nicht über HTTP verfügbar ist, und der Webserver leitet Sie zur gesicherten Version weiter.

```
sh-4.4$ curl -I http://172.30.121.154
HTTP/1.1 301 Moved Permanently
Server: nginx/1.14.1
Date: Tue, 15 Jun 2021 02:01:19 GMT
Content-Type: text/html
Connection: keep-alive
Location: https://172.30.121.154:8443/
```

- 7.4. Greifen Sie über HTTPS auf die Anwendung zu. Verwenden Sie die Option `-k`, da der Container keinen Zugriff auf das CA-Zertifikat hat.

```
sh-4.4$ curl -s -k https://172.30.121.154:8443 | head -n5
<!DOCTYPE html>
<html lang="en" ng-app="todoItemsApp" ng-controller="appCtl">
<head>
  <meta charset="utf-8">
  <title>ToDo app</title>
```

- 7.5. Beenden Sie den Debug-Pod.

```
sh-4.4$ exit  
Removing debug pod ...
```

- 8. Wechseln Sie zum Benutzerverzeichnis, und löschen Sie das Projekt `network-ingress`.

```
[student@workstation network-ingress]$ cd  
[student@workstation ~]$ oc delete project network-ingress  
project.project.openshift.io "network-ingress" deleted
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab network-ingress finish
```

Hiermit ist die angeleitete Übung beendet.

Konfigurieren von Netzwerk-Richtlinien

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, den Netzwerkdatenverkehr zwischen Projekten und Pods einzuschränken.

Verwalten von Netzwerkrichtlinien in OpenShift

Mit Netzwerkrichtlinien können Sie Isolierungsrichtlinien für einzelne Pods konfigurieren. Netzwerkrichtlinien erfordern keine Administratorrechte und bieten Entwicklern mehr Kontrolle über die Anwendungen in Ihren Projekten. Mit Netzwerkrichtlinien können Sie logische Zonen im SDN erstellen, die den Netzwerkzonen Ihrer Organisation zugeordnet sind. Dies hat den Vorteil, dass der Speicherort laufender Pods irrelevant wird, da Sie mit Netzwerkrichtlinien den Datenverkehr unabhängig vom Ursprungsort trennen können.

Um die Netzwerkkommunikation zwischen zwei Namespaces zu verwalten, weisen Sie demjenigen Namespace eine Bezeichnung zu, der auf einen anderen Namespace zugreifen muss. Der folgende Befehl weist die Bezeichnung `name=network-1` zum Namespace `network-1` zu:

```
[user@host ~]$ oc label namespace network-1 name=network-1
```

Die folgenden Beispiele beschreiben Netzwerkrichtlinien, die die Kommunikation zwischen den Namespaces `network-1` und `network-2` ermöglichen:

- Die folgende Netzwerkrichtlinie gilt für alle Pods mit dem Label `deployment="product-catalog"` im Namespace `network-1`. Die Richtlinie erlaubt den TCP-Datenverkehr über Port 8080 von Pods mit dem Label `role="qa"` im Namespace `network-2`.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: network-1-policy
spec:
  podSelector: ①
  matchLabels:
    deployment: product-catalog

  ingress: ②
  - from: ③
    - namespaceSelector:
        matchLabels:
          name: network-2
    podSelector:
      matchLabels:
        role: qa
  ports: ④
  - port: 8080
    protocol: TCP
```

- ❶ Das `podSelector`-Feld auf der obersten Ebene ist erforderlich und definiert, welche Pods die Netzwerkrichtlinie verwenden. Wenn `podSelector` leer ist, werden alle Pods im Namespace berücksichtigt.
 - ❷ Das Feld `Ingress` definiert eine Liste der eingehenden Datenverkehrsregeln, die auf die übereinstimmenden Pods des `podSelectors` der obersten Ebene angewendet werden.
 - ❸ Das Feld `from` definiert eine Liste von Regeln, die den Datenverkehr aus allen Quellen abgleichen. Die Selektoren sind nicht auf das Projekt beschränkt, in dem die Netzwerkrichtlinie definiert ist.
 - ❹ Das Feld `Ports` enthält eine Liste von Zielports, über die der Datenverkehr die ausgewählten Pods erreichen kann.
- Die folgende Netzwerkrichtlinie erlaubt den Datenverkehr von allen Pods und Ports im Namespace `network-1` zu allen Pods und Ports im Namespace `network-2`. Diese Richtlinie ist weniger restriktiv als die Richtlinie für `network-1`, da sie den Datenverkehr von den Pods im Namespace `network-1` nicht einschränkt.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: network-2-policy
spec:
  podSelector: {}

  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          name: network-1
```



Anmerkung

Netzwerkrichtlinien sind Kubernetes-Ressourcen. Daher können Sie sie mit den oc-Befehlen verwalten.

Mit Netzwerkrichtlinien können Sie außerdem die Sicherheit zwischen Projekten (Mandanten) verwalten. Dies ist mit Layer-2-Technologien wie VLANs nicht möglich. Auf diese Weise können Sie maßgeschneiderte Richtlinien zwischen Projekten erstellen, damit Benutzer nur auf die Elemente zugreifen können, für die sie Berechtigungen haben (Least Privilege-Ansatz).

Die Felder in der Netzwerkrichtlinie, die eine Liste von Objekten enthalten, können entweder im selben Objekt kombiniert oder als mehrere Objekte aufgelistet werden. Kombinierte Bedingungen werden mit einem logischen *UND* kombiniert. In einer Liste getrennte Bedingungen werden mit einem logischen *ODER* kombiniert. Mit den logischen Optionen können Sie sehr spezifische Richtlinienregeln erstellen. Die folgenden Beispiele zeigen die Unterschiede zwischen den verschiedenen Syntaxregeln:

- In diesem Beispiel werden die Selektoren in einer Regel kombiniert, um nur den Pods im Namespace `dev` mit der Bezeichnung `app=mobile` den Zugriff zu erlauben. Dies ist ein Beispiel für ein logisches *UND*.

```
...output omitted...
ingress:
- from:
  - namespaceSelector:
    matchLabels:
      name: dev
  podSelector:
    matchLabels:
      app: mobile
```

- Wenn Sie das Feld `podSelector` im vorherigen Beispiel so ändern, dass es ein Element in der Liste `from` ist, können alle Pods im `dev`-Namespace und alle Pods aus Namespaces mit der Bezeichnung `app=mobile` die Pods erreichen, die mit dem `podSelector`-Feld der obersten Ebene übereinstimmen. Dies ist ein Beispiel für ein logisches *ODER*.

```
...output omitted...
ingress:
- from:
  - namespaceSelector:
    matchLabels:
      name: dev
  - podSelector:
    matchLabels:
      app: mobile
```

Wenn ein Pod in einer oder mehreren Netzwerkrichtlinien mit Selektoren übereinstimmt, akzeptiert der Pod nur Verbindungen, die von mindestens einer dieser Netzwerkrichtlinien erlaubt sind. Ein striktes Beispiel ist eine Richtlinie, die den gesamten eingehenden Datenverkehr zu Pods in Ihrem Projekt verbietet, selbst von anderen Pods in Ihrem Projekt. Eine leere Pod-Auswahl bedeutet, dass diese Richtlinie für alle Pods in diesem Projekt gilt. Die folgende Richtlinie blockiert den gesamten Datenverkehr, da keine Ingress-Regeln definiert sind. Der Datenverkehr wird blockiert, es sei denn, Sie definieren eine explizite Richtlinie, die dieses Standardverhalten außer Kraft setzt.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny
spec:
  podSelector: {}
```

Wenn Sie eine Cluster-Überwachung oder freigegebene Routen verwenden, müssen dies auch bei der Ingress-Konfiguration berücksichtigen. Mit den folgenden Richtlinien können Sie den Ingress von den OpenShift-Überwachungs- und Ingress-Controllern erlauben:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-from-openshift-ingress
spec:
  podSelector: {}
  ingress:
  - from:
```

```
- namespaceSelector:  
  matchLabels:  
    network.openshift.io/policy-group: ingress  
---  
apiVersion: networking.k8s.io/v1  
kind: NetworkPolicy  
metadata:  
  name: allow-from-openshift-monitoring  
spec:  
  podSelector: {}  
  ingress:  
  - from:  
    - namespaceSelector:  
      matchLabels:  
        network.openshift.io/policy-group: monitoring
```



Wichtig

Wenn der standardmäßige Ingress-Controller die HostNetwork-Veröffentlichungsstrategie für Endpunkte verwendet, benötigt der default-Namespace die Bezeichnung `network.openshift.io/policy-group=ingress`.

Überprüfen Sie die Veröffentlichungsstrategie für Endpunkte mit dem Befehl `oc describe`, um die `ingresscontroller/default`-Ressource im Namespace `openshift-ingress-controller` zu beschreiben.

Weitere Informationen finden Sie in der Dokumentation, die in den Referenzen verlinkt ist.



Literaturhinweise

Weitere Informationen zu Netzwerkrichtlinien finden Sie im Kapitel *Network policy* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Networking* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/networking/index#network-policy

► Angeleitete Übung

Konfigurieren von Netzwerk-Richtlinien

In dieser Übung erstellen Sie Netzwerkrichtlinien und prüfen die durch diese Netzwerkrichtlinien erstellte Pod-Isolierung.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Konfigurieren von Netzwerkrichtlinien, um die Kommunikation zwischen Pods zu steuern.
- Überprüfen, ob der Ingress-Datenverkehr auf Pods beschränkt ist.

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Umgebung bereit ist, und lädt die für diese Übung benötigten Ressourcen herunter.

```
[student@workstation ~]$ lab network-policy start
```

Anweisungen

- 1. Melden Sie sich bei dem OpenShift-Cluster an, und erstellen Sie das Projekt `network-policy`.

- 1.1. Melden Sie sich als Benutzer `developer` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Erstellen Sie das Projekt `network-policy`.

```
[student@workstation ~]$ oc new-project network-policy
Now using project "network-policy" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

- 2. Erstellen Sie zwei Bereitstellungen, und erstellen Sie eine Route für eine davon.

- 2.1. Erstellen Sie zwei Bereitstellungen mit dem Befehl `oc new-app` und dem Image `quay.io/redhattraining/hello-world-nginx:v1.0`. Nennen Sie die erste Bereitstellung `hello` und die zweite Bereitstellung `test`.

```
[student@workstation ~]$ oc new-app --name hello --docker-image \
>   quay.io/redhattraining/hello-world-nginx:v1.0
...output omitted...
--> Creating resources ...
  imagestream.image.openshift.io "hello" created
  deployment.apps "hello" created
  service "hello" created
--> Success
...output omitted...
[student@workstation ~]$ oc new-app --name test --docker-image \
>   quay.io/redhattraining/hello-world-nginx:v1.0
...output omitted...
--> Creating resources ...
  imagestream.image.openshift.io "test" created
  deployment.apps "test" created
  service "test" created
--> Success
...output omitted...
```

- 2.2. Verwenden Sie den Befehl `oc expose`, um eine Route zum `hello`-Service zu erstellen.

```
[student@workstation ~]$ oc expose service hello
route.route.openshift.io/hello exposed
```

- 3. Überprüfen Sie den Zugriff auf den Pod `Hello` mit den Befehlen `oc rsh` und `curl`.

- 3.1. Öffnen Sie ein zweites Terminal, und führen Sie das Skript unter `~/D0280/labs/network-policy/display-project-info.sh` aus. Dieses Skript liefert Informationen über die Pods, den Service und die Route, die Sie im weiteren Verlauf dieser Übung benötigen.

```
[student@workstation ~]$ ~/D0280/labs/network-policy/display-project-info.sh
=====
PROJECT: network-policy

POD NAME          IP ADDRESS
hello-6c4984d949-g28c4  10.8.0.13
test-c4d74c9d5-5pq9s  10.8.0.14

SERVICE NAME    CLUSTER-IP
hello           172.30.137.226
test            172.30.159.119

ROUTE NAME      HOSTNAME                      PORT
hello           hello-network-policy.apps.ocp4.example.com  8080-tcp
=====
```

- 3.2. Verwenden Sie die Befehle `oc rsh` und `curl`, um zu überprüfen, ob der `test`-Pod auf die IP-Adresse des `hello`-Pods zugreifen kann.

```
[student@workstation ~]$ oc rsh test-c4d74c9d5-5pq9s curl 10.8.0.13:8080 | \
> grep Hello
<h1>Hello, world from nginx!</h1>
```

- 3.3. Verwenden Sie die Befehle `oc rsh` und `curl`, um zu überprüfen, ob der `test`-Pod auf die IP-Adresse des `hello`-Service zugreifen kann.

```
[student@workstation ~]$ oc rsh test-c4d74c9d5-5pq9s curl 172.30.137.226:8080 | \
> grep Hello
<h1>Hello, world from nginx!</h1>
```

- 3.4. Überprüfen Sie den Zugriff auf den `hello`-Pod mit dem Befehl `curl` für die URL der `hello`-Route.

```
[student@workstation ~]$ curl -s hello-network-policy.apps.ocp4.example.com | \
> grep Hello
<h1>Hello, world from nginx!</h1>
```

- ▶ 4. Erstellen Sie das Projekt `network-test` und eine Bereitstellung mit dem Namen `sample-app`.

- 4.1. Erstellen Sie das Projekt `network-test`.

```
[student@workstation ~]$ oc new-project network-test
Now using project "network-test" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

- 4.2. Erstellen Sie die Bereitstellung `sample-app` mit dem Image `quay.io/redhattraining/hello-world-nginx:v1.0`. Die Web-Anwendung überwacht den Port 8080.

```
[student@workstation ~]$ oc new-app --name sample-app --docker-image \
> quay.io/redhattraining/hello-world-nginx:v1.0
...output omitted...
--> Creating resources ...
imagestream.image.openshift.io "sample-app" created
deployment.apps "sample-app" created
service "sample-app" created
--> Success
...output omitted...
```

- ▶ 5. Überprüfen Sie, ob Pods in einem anderen Namespace auf die `hello`- und `test`-Pods im Namespace `network-policy` zugreifen können.

- 5.1. Führen Sie im zweiten Terminal das Skript `display-project-info.sh` erneut aus, um den vollständigen Namen des `sample-app`-Pods anzuzeigen.

```
[student@workstation ~]$ ~/D0280/labs/network-policy/display-project-info.sh  
...output omitted...  
PROJECT: network-test  
  
POD NAME  
sample-app-d5f945-spx9q  
=====
```

- 5.2. Kehren Sie zum ersten Terminal zurück, und verwenden Sie die Befehle `oc rsh` und `curl`, um zu überprüfen, ob der `sample-app`-Pod auf die IP-Adresse des `hello`-Pods zugreifen kann.

```
[student@workstation ~]$ oc rsh sample-app-d5f945-spx9q curl 10.8.0.13:8080 | \  
> grep Hello  
<h1>Hello, world from nginx!</h1>
```

- 5.3. Verwenden Sie die Befehle `oc rsh` und `curl`, um zu überprüfen, ob der `test`-Pod für den `sample-app`-Pod erreichbar ist. Verwenden Sie die IP-Adresse als Ziel, die Sie zuvor für den `test`-Pod abgerufen haben.

```
[student@workstation ~]$ oc rsh sample-app-d5f945-spx9q curl 10.8.0.14:8080 | \  
> grep Hello  
<h1>Hello, world from nginx!</h1>
```

- ▶ 6. Erstellen Sie im Projekt `network-policy` die Netzwerkrichtlinie `deny-all` mit der unter `~/D0280/labs/network-policy/deny-all.yaml` verfügbaren Ressourcendatei.

- 6.1. Wechseln Sie zum Projekt `network-policy`.

```
[student@workstation ~]$ oc project network-policy  
Now using project "network-policy" on server "https://api.ocp4.example.com:6443".
```

- 6.2. Wechseln Sie zum Verzeichnis `~/D0280/labs/network-policy/`.

```
[student@workstation ~]$ cd ~/D0280/labs/network-policy/
```

- 6.3. Verwenden Sie einen Texteditor, um die Datei `deny-all.yaml` mit einem leeren `podSelector` zu aktualisieren, der alle Pods im Namespace als Ziel verwendet. Eine Lösung finden Sie unter `~/D0280/solutions/network-policy/deny-all.yaml`.

```
kind: NetworkPolicy  
apiVersion: networking.k8s.io/v1  
metadata:  
  name: deny-all  
spec:  
  podSelector: {}
```

- 6.4. Erstellen Sie die Netzwerkrichtlinie mit dem Befehl `oc create`.

```
[student@workstation network-policy]$ oc create -f deny-all.yaml
networkpolicy.networking.k8s.io/deny-all created
```

- ▶ 7. Stellen Sie sicher, dass nicht mehr auf die Pods im Namespace `network-policy` zugegriffen werden kann.
 - 7.1. Stellen Sie sicher, dass kein Zugriff mehr auf den Pod `hello` über die bereitgestellte Route erfolgt. Warten Sie einige Sekunden, und drücken Sie dann Strg+C, um den curl-Befehl zu beenden, der nicht reagiert.

```
[student@workstation network-policy]$ curl -s \
>   hello-network-policy.apps.ocp4.example.com | grep Hello
^C
```



Wichtig

Wenn sich der `hello`-Pod auf demselben Knoten wie ein `router-default`-Pod befindet, funktioniert der Befehl `curl`, wenn der Datenverkehr diesen Router-Pod durchläuft. Dies ist nur in Clustern mit drei Knoten der Fall. In einem herkömmlichen OpenShift-Cluster, in dem die Control Plane- oder Infrastrukturknoten von den Computing-Knoten getrennt sind, wird die Netzwerkrichtlinie auf alle Router-Pods im Cluster angewendet.

Wenn der Befehl `curl` erfolgreich ist, führen Sie ihn erneut aus, um zu überprüfen, ob die Netzwerkrichtlinie wie erwartet funktioniert. Dieser zweite Versuch sollte den anderen Router-Pod im Cluster durchlaufen.

- 7.2. Stellen Sie sicher, dass der `test`-Pod nicht mehr auf die IP-Adresse des `hello`-Pods zugreifen kann. Warten Sie einige Sekunden, und drücken Sie dann Strg+C, um den `curl`-Befehl zu beenden, der nicht reagiert.

```
[student@workstation network-policy]$ oc rsh test-c4d74c9d5-5pq9s curl \
>   10.8.0.13:8080 | grep Hello
^Ccommand terminated with exit code 130
```

- 7.3. Vergewissern Sie sich im Projekt `network-test`, dass der Pod `sample-app` nicht mehr auf die IP-Adresse des `test`-Pods zugreifen kann. Warten Sie einige Sekunden, und drücken Sie dann Strg+C, um den `curl`-Befehl zu beenden, der nicht reagiert.

```
[student@workstation network-policy]$ oc project network-test
Now using project "network-test" on server "https://api.ocp4.example.com:6443".
[student@workstation network-policy]$ oc rsh sample-app-d5f945-spx9q curl \
>   10.8.0.14:8080 | grep Hello
^Ccommand terminated with exit code 130
```

- ▶ 8. Erstellen Sie eine Netzwerkrichtlinie, um den Datenverkehr zum `hello`-Pod im `network-policy`-Namespace vom `sample-app`-Pod im Namespace `network-test` über TCP auf Port 8080 zu erlauben. Verwenden Sie die unter `~/D0280/labs/network-policy/allow-specific.yaml` verfügbare Ressourcendatei.

- 8.1. Verwenden Sie einen Texteditor, um die Abschnitte mit CHANGE_ME in der Datei `allow-specific.yaml` wie folgt zu ersetzen. Eine Lösung finden Sie unter `~/DO280/solutions/network-policy/allow-specific.yaml`.

```
...output omitted...
spec:
  podSelector:
    matchLabels:
      deployment: hello
  ingress:
    - from:
      - namespaceSelector:
          matchLabels:
            name: network-test
        podSelector:
          matchLabels:
            deployment: sample-app
    ports:
      - port: 8080
        protocol: TCP
```

- 8.2. Erstellen Sie die Netzwerkrichtlinie mit dem Befehl `oc create`.

```
[student@workstation network-policy]$ oc create -n network-policy -f \
>   allow-specific.yaml
networkpolicy.networking.k8s.io/allow-specific created
```

- 8.3. Rufen Sie die Netzwerkrichtlinien im Namespace `network-policy` ab.

```
[student@workstation network-policy]$ oc get networkpolicies -n network-policy
NAME          POD-SELECTOR     AGE
allow-specific deployment=hello  11s
deny-all       <none>           5m6s
```

- 9. Verwenden Sie den `admin`-Benutzer, um dem Namespace `network-test` das Label `name=network-test` hinzuzufügen.

- 9.1. Melden Sie sich als Benutzer `admin` an.

```
[student@workstation network-policy]$ oc login -u admin -p redhat
Login successful.
...output omitted...
```

- 9.2. Verwenden Sie den Befehl `oc label`, um das Label `name=network-test` anzuwenden.

```
[student@workstation network-policy]$ oc label namespace network-test \
>   name=network-test
namespace/network-test labeled
```



Wichtig

Die Netzwerkrichtlinie `allow-specific` verwendet Labels, um dem Namen eines Namespace abzuleiten. Standardmäßig erhalten Namespaces und Projekte keine automatisch vergebenen Labels.

- 9.3. Bestätigen Sie, dass das Label angewendet wurde, und melden Sie sich als Benutzer `developer` an.

```
[student@workstation network-policy]$ oc describe namespace network-test
Name:           network-test
Labels:         name=network-test
...output omitted...
[student@workstation network-policy]$ oc login -u developer -p developer
Login successful.
...output omitted...
```



Anmerkung

Stellen Sie sicher, dass Sie sich im `network-test`-Projekt befinden, oder die nächsten Befehle schlagen fehl.

- 10. Vergewissern Sie sich, dass der Pod `sample-app` auf die IP-Adresse des `hello`-Pods zugreifen kann, aber keinen Zugriff auf die IP-Adresse des `test`-Pods hat.

- 10.1. Überprüfen Sie den Zugriff auf den `hello`-Pod im Namespace `network-policy` namespace.

```
[student@workstation network-policy]$ oc rsh sample-app-d5f945-spx9q curl \
>   10.8.0.13:8080 | grep Hello
<h1>Hello, world from nginx!</h1>
```

- 10.2. Vergewissern Sie sich, dass der `hello`-Pod nicht auf einem anderen Port antwortet. Da die Netzwerkrichtlinie nur den Zugriff auf Port 8080 auf dem `hello`-Pod zulässt, werden Anfragen an andere Ports ignoriert, und es tritt eine Zeitüberschreitung auf. Warten Sie einige Sekunden, und drücken Sie dann Strg+C, um den curl-Befehl zu beenden, der nicht reagiert.

```
[student@workstation network-policy]$ oc rsh sample-app-d5f945-spx9q curl \
>   10.8.0.13:8181 | grep Hello
^Ccommand terminated with exit code 130
```

- 10.3. Stellen Sie sicher, dass kein Zugriff auf den `test`-Pod möglich ist. Warten Sie einige Sekunden, und drücken Sie dann Strg+C, um den curl-Befehl zu beenden, der nicht reagiert.

```
[student@workstation network-policy]$ oc rsh sample-app-d5f945-spx9q curl \
>   10.8.0.14:8080 | grep Hello
^Ccommand terminated with exit code 130
```

- 11. Erstellen Sie eine Netzwerkrichtlinie, um den Datenverkehr zum hello-Pod über die freigegebene Route zu ermöglichen. Verwenden Sie die unter `~/DO280/labs/network-policy/allow-from-openshift-ingress.yaml` verfügbare Ressourcendatei.

- 11.1. Verwenden Sie einen Texteditor, um die Werte mit `CHANGE_ME` in der Datei `allow-from-openshift-ingress.yaml` wie folgt zu ersetzen. Eine Lösung finden Sie unter `~/DO280/solutions/network-policy/allow-from-openshift-ingress.yaml`.

```
...output omitted...
spec:
  podSelector: {}
  ingress:
    - from:
      - namespaceSelector:
          matchLabels:
            network.openshift.io/policy-group: ingress
```

- 11.2. Erstellen Sie die Netzwerkrichtlinie mit dem Befehl `oc create`.

```
[student@workstation network-policy]$ oc create -n network-policy -f \
>   allow-from-openshift-ingress.yaml
networkpolicy.networking.k8s.io/allow-from-openshift-ingress created
```

- 11.3. Rufen Sie die Netzwerkrichtlinien im Namespace `network-policy` ab.

```
[student@workstation network-policy]$ oc get networkpolicies -n network-policy
NAME                      POD-SELECTOR      AGE
allow-from-openshift-ingress <none>        10s
allow-specific              deployment=hello  8m16s
deny-all                   <none>         13m
```

- 11.4. Fügen Sie als `admin`-Benutzer `network.openshift.io/policy-group=ingress` zum Namespace `default` hinzu.

```
[student@workstation network-policy]$ oc login -u admin -p redhat
Login successful.
...output omitted...
[student@workstation network-policy]$ oc label namespace default \
>   network.openshift.io/policy-group=ingress
namespace/default labeled
```



Anmerkung

Dieses Label muss nur deshalb auf den `default`-Namespace angewendet werden, weil der standardmäßige Ingress-Controller des Kursraums die HostNetwork-Veröffentlichungsstrategie für Endpunkte verwendet.

- 11.5. Testen Sie den Zugriff auf den Pod `hello` über die bereitgestellte Route.

```
[student@workstation network-policy]$ curl -s \
>   hello-network-policy.apps.ocp4.example.com | grep Hello
<h1>Hello, world from nginx!</h1>
```

- 12. Schließen Sie das Terminalfenster mit der Ausgabe des Skripts `display-project-info.sh`. Navigieren Sie zum Benutzerverzeichnis.

```
[student@workstation network-policy]$ cd
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab network-policy finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Konfigurieren des OpenShift-Netzwerks für Anwendungen

In dieser praktischen Übung konfigurieren Sie eine TLS-Passthrough-Route für Ihre Anwendung.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Bereitstellen einer Anwendung und Konfigurieren einer ungesicherten Route
- Einschränken des Datenverkehrs auf die Anwendungen
- Generieren eines TLS-Zertifikats für eine Anwendung
- Konfigurieren einer Passthrough-Route für eine Anwendung mit einem TLS-Zertifikat

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt die selbstsignierte Zertifizierungsstelle (Certification Authority, CA), die Sie in dieser Übung verwenden.

```
[student@workstation ~]$ lab network-review start
```

Anweisungen

In dieser Übung stellen Sie eine PHP-Anwendung bereit, die Informationen zum System ausgibt. Die Anwendung ist mit zwei verschiedenen Konfigurationen verfügbar: entweder mit einem unverschlüsselten Netzwerk und Überwachung von Port 8080 oder mit einem TLS-Zertifikat zur Verschlüsselung des Netzwerkdatenverkehrs und Port 8443.

Das Container-Image für diese Übung kann unter `quay.io/redhattraining/php-ssl` abgerufen werden. Es verfügt über zwei Tags: `v1.0` für die ungesicherte Version der Anwendung und `v1.1` für die gesicherte Version.

1. Erstellen Sie als OpenShift-Benutzer `developer` das Projekt `network-review`.
2. Stellen Sie als Benutzer `developer` die erste ungesicherte Version der PHP-Anwendung im Projekt `network-review` bereit. Verwenden Sie die unter `~/D0280/labs/network-review/php-http.yaml` verfügbare Ressourcendatei.

Nehmen Sie vor der Bereitstellung der Anwendung die erforderlichen Änderungen an der Datei vor, insbesondere den Speicherort des Container-Images und den zu überwachenden Port.

Warten Sie nach dem Erstellen der Anwendung einen Moment, um sicherzustellen, dass ein Pod ausgeführt wird.

3. Erstellen Sie eine Route mit dem Namen `php-http` und dem Hostnamen `php-http.apps.ocp4.example.com`, um auf die Anwendung zuzugreifen.
Greifen Sie auf dem Rechner `workstation` mit Firefox auf `http://php-http.apps.ocp4.example.com` zu. Vergewissern Sie sich, dass die Anwendung verfügbar ist, bevor Sie mit dem nächsten Schritt fortfahren.
4. Erstellen Sie eine Netzwerkrichtlinie im Namespace `network-review`, um den gesamten eingehenden Datenverkehr standardmäßig zu blockieren. Bei ordnungsgemäßer Konfiguration verhindert die Netzwerkrichtlinie auch, dass Pods innerhalb des Namespace `network-review` miteinander kommunizieren.
Verwenden Sie die unter `~/D0280/labs/network-review/deny-all.yaml` verfügbare Ressourcendatei. Nehmen Sie die erforderlichen Änderungen vor, um alle Pods im Namespace als Ziel zu verwenden.
5. Erstellen Sie eine Netzwerkrichtlinie, um den gesamten eingehenden Datenverkehr zu Routen im Namespace `network-review` zu erlauben.
Verwenden Sie die unter `~/D0280/labs/network-review/allow-from-openshift-ingress.yaml` verfügbare Ressourcendatei. Nehmen Sie die erforderlichen Änderungen vor, um alle Pods im Namespace als Ziel zu verwenden und den Datenverkehr vom standardmäßigen Ingress-Controller zu erlauben.
Da die Kursumgebung die `HostNetwork`-Strategie für Endpunkte verwendet, fügen Sie das Label `network.openshift.io/policy-group=ingress` zum Namespace `default` hinzu. Diese Aktion muss als `admin`-Benutzer ausgeführt werden.
6. Erstellen und signieren Sie ein TLS-Zertifikat für die verschlüsselte Version der Anwendung als Benutzer `developer`.
Generieren Sie die Anforderung für die Signierung des Zertifikats (Certificate Signing Request, CSR) für den Hostnamen `php-https.apps.ocp4.example.com`. Speichern Sie die CSR in `~/D0280/labs/network-review/certs/training.csr`.
Erstellen Sie mit der CSR ein Zertifikat, und speichern Sie es in `~/D0280/labs/network-review/certs/training.crt`. Übergeben Sie als Argumente das unter `~/D0280/labs/network-review/certs/training-CA.pem` verfügbare CA-Zertifikat und die CSR, um das Zertifikat zu generieren.
Sie können die Textdatei `~/D0280/labs/network-review/certs/openssl-commands.txt` als Hilfe verwenden. Diese Datei enthält die Befehle, um die Anforderung für die Signierung des Zertifikats und das Zertifikat zu generieren. Ersetzen Sie die Werte in der Datei, bevor Sie die OpenSSL-Befehle kopieren und ausführen.
7. Erstellen Sie ein OpenShift-TLS-Secret mit dem Namen `php-certs` im Projekt `network-review`. Verwenden Sie die Datei `~/D0280/labs/network-review/certs/training.crt` für das Zertifikat und die Datei `~/D0280/labs/network-review/certs/training.key` für den Schlüssel.
8. Verwenden Sie die unter `~/D0280/labs/network-review/php-https.yaml` verfügbare Ressourcendatei, um die gesicherte Version der PHP-Anwendung bereitzustellen. Stellen Sie die Anwendung im Projekt `network-review` bereit.
Nehmen Sie vor der Bereitstellung der Anwendung die erforderlichen Änderungen an der Ressourcendatei vor. Ändern Sie:
 - Den Speicherort des Containers
 - Den Port, den die Anwendung überwacht
 - Den Namen des Secret, das als Volume gemountet werden soll

Kapitel 5 | Konfigurieren des OpenShift-Netzwerks für Anwendungen

9. Erstellen Sie eine gesicherte Passthrough-Route mit dem Namen `php-https` und dem Hostnamen `php-https.apps.ocp4.example.com`, um auf die gesicherte Version der Anwendung zuzugreifen.
Greifen Sie auf dem Rechner `workstation` mit Firefox auf `https://php-https.apps.ocp4.example.com` zu. Akzeptieren Sie das signierte Zertifikat, und bestätigen Sie, dass die Anwendung verfügbar ist.
10. *Optionaler Schritt:* Führen Sie den Befehl `curl` auf dem Rechner `workstation`, aus um die HTTPS-Version der Anwendung zu überprüfen.
Übergeben Sie das CA-Zertifikat an den `curl`-Befehl, um die sichere Verbindung zu validieren.
11. Kehren Sie zum Benutzerverzeichnis zurück, da der Befehl `lab network-review finish` den Ordner `network-review` löscht.

```
[student@workstation network-review]$ cd
```

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.

```
[student@workstation ~]$ lab network-review grade
```

Beenden

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab network-review finish
```

Hiermit ist die praktische Übung beendet.

► Lösung

Konfigurieren des OpenShift-Netzwerks für Anwendungen

In dieser praktischen Übung konfigurieren Sie eine TLS-Passthrough-Route für Ihre Anwendung.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Bereitstellen einer Anwendung und Konfigurieren einer ungesicherten Route
- Einschränken des Datenverkehrs auf die Anwendungen
- Generieren eines TLS-Zertifikats für eine Anwendung
- Konfigurieren einer Passthrough-Route für eine Anwendung mit einem TLS-Zertifikat

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt die selbstsignierte Zertifizierungsstelle (Certification Authority, CA), die Sie in dieser Übung verwenden.

```
[student@workstation ~]$ lab network-review start
```

Anweisungen

In dieser Übung stellen Sie eine PHP-Anwendung bereit, die Informationen zum System ausgibt. Die Anwendung ist mit zwei verschiedenen Konfigurationen verfügbar: entweder mit einem unverschlüsselten Netzwerk und Überwachung von Port 8080 oder mit einem TLS-Zertifikat zur Verschlüsselung des Netzwerksdatenverkehrs und Port 8443.

Das Container-Image für diese Übung kann unter `quay.io/redhattraining/php-ssl` abgerufen werden. Es verfügt über zwei Tags: `v1.0` für die ungesicherte Version der Anwendung und `v1.1` für die gesicherte Version.

1. Erstellen Sie als OpenShift-Benutzer `developer` das Projekt `network-review`.

- 1.1. Melden Sie sich als Benutzer `developer` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Erstellen Sie das Projekt `network-review`.

```
[student@workstation ~]$ oc new-project network-review
Now using project "network-review" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

2. Stellen Sie als Benutzer developer die erste ungesicherte Version der PHP-Anwendung im Projekt `network-review` bereit. Verwenden Sie die unter `~/D0280/labs/network-review/php-http.yaml` verfügbare Ressourcendatei.

Nehmen Sie vor der Bereitstellung der Anwendung die erforderlichen Änderungen an der Datei vor, insbesondere den Speicherort des Container-Images und den zu überwachenden Port.

Warten Sie nach dem Erstellen der Anwendung einen Moment, um sicherzustellen, dass ein Pod ausgeführt wird.

- 2.1. Wechseln Sie zum Verzeichnis `~/D0280/labs/network-review/`.

```
[student@workstation ~]$ cd ~/D0280/labs/network-review/
```

- 2.2. Aktualisieren Sie in einem Texteditor die Datei `php-http.yaml` wie folgt:

- Suchen Sie den Eintrag `image`. Legen Sie ihn so fest, dass das Container-Image unter `quay.io/redhattraining/php-ssl:v1.0` verwendet wird.

```
...output omitted...
cpu: '0.5'
image: 'quay.io/redhattraining/php-ssl:v1.0'
name: php-http
...output omitted...
```

- Suchen Sie den Eintrag `containerPort`. Legen Sie den Wert auf `8080` fest, der dem ungesicherten Endpunkt entspricht.

```
...output omitted...
ports:
- containerPort: 8080
  name: php-http
...output omitted...
```

Nachdem Sie die Änderungen vorgenommen haben, speichern und beenden Sie die Datei.

- 2.3. Stellen Sie die Anwendung mit dem Befehl `oc create` bereit. Dadurch werden eine Bereitstellung und ein Service erstellt.

```
[student@workstation network-review]$ oc create -f php-http.yaml
deployment.apps/php-http created
service/php-http created
```

- 2.4. Warten Sie einen Moment, und führen Sie dann `oc get pods` aus, um zu überprüfen, ob ein aktiver Pod vorhanden ist.

```
[student@workstation network-review]$ oc get pods
NAME          READY   STATUS    RESTARTS   AGE
php-http-6cb58c847b-7qsbd  1/1     Running   0          8m11s
```

3. Erstellen Sie eine Route mit dem Namen `php-http` und dem Hostnamen `php-http.apps.ocp4.example.com`, um auf die Anwendung zuzugreifen. Greifen Sie auf dem Rechner `workstation` mit Firefox auf `http://php-http.apps.ocp4.example.com` zu. Vergewissern Sie sich, dass die Anwendung verfügbar ist, bevor Sie mit dem nächsten Schritt fortfahren.
- 3.1. Erstellen Sie mit dem Befehl `oc expose` eine Route für den Zugriff auf die Anwendung. Geben Sie der Route den Hostnamen `php-http.apps.ocp4.example.com`.

```
[student@workstation network-review]$ oc expose svc php-http \
>   --hostname php-http.apps.ocp4.example.com
route.route.openshift.io/php-http exposed
```

- 3.2. Rufen Sie den Namen der Route ab, und kopieren Sie ihn in die Zwischenablage.

```
[student@workstation network-review]$ oc get routes
NAME      HOST/PORT           PATH  SERVICES  PORT  ...
php-http  php-http.apps.ocp4.example.com  php-http  8080  ...
```

- 3.3. Öffnen Sie Firefox auf dem Rechner `workstation` und öffnen Sie `http://php-http.apps.ocp4.example.com`. Vergewissern Sie sich, dass die Anwendung angezeigt wird.

About this application

⚠ The application is currently served over HTTP

- **Current system load:** 2.5
- **Number of connections:** 1
- **Memory usage:** 8 Mb

4. Erstellen Sie eine Netzwerkrichtlinie im Namespace `network-review`, um den gesamten eingehenden Datenverkehr standardmäßig zu blockieren. Bei ordnungsgemäßer Konfiguration verhindert die Netzwerkrichtlinie auch, dass Pods innerhalb des Namespace `network-review` miteinander kommunizieren. Verwenden Sie die unter `~/D0280/labs/network-review/deny-all.yaml` verfügbare Ressourcendatei. Nehmen Sie die erforderlichen Änderungen vor, um alle Pods im Namespace als Ziel zu verwenden.
- 4.1. Verwenden Sie einen Texteditor, um die Datei `deny-all.yaml` mit einem leeren Pod-Selektor zu aktualisieren, der alle Pods im Namespace als Ziel verwendet.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: deny-all
spec:
  podSelector: {}
```

- 4.2. Erstellen Sie die Netzwerkrichtlinie mit dem Befehl `oc create`.

```
[student@workstation network-review]$ oc create -f deny-all.yaml
networkpolicy.networking.k8s.io/deny-all created
```

- 4.3. Vergewissern Sie sich mit dem Befehl `curl`, dass über die Route kein Zugriff auf den `php-http`-Pod möglich ist. Warten Sie einige Sekunden, und drücken Sie dann `Strg+C`, um den `curl`-Befehl zu beenden.

```
[student@workstation network-review]$ curl http://php-http.apps.ocp4.example.com
^C
```

Wichtig

Wenn sich der `php-http`-Pod auf demselben Knoten wie ein `router-default`-Pod befindet, funktioniert der Befehl `curl`, wenn der Datenverkehr diesen Router-Pod durchläuft. Dies ist nur in Clustern mit drei Knoten der Fall. In einem herkömmlichen OpenShift-Cluster, in dem die Control Plane- oder Infrastrukturknoten von den Computing-Knoten getrennt sind, wird die Netzwerkrichtlinie auf alle Router-Pods im Cluster angewendet.

Wenn der Befehl `curl` erfolgreich ist, führen Sie ihn erneut aus, um zu überprüfen, ob die Netzwerkrichtlinie wie erwartet funktioniert. Dieser zweite Versuch sollte den anderen Router-Pod im Cluster durchlaufen.

5. Erstellen Sie eine Netzwerkrichtlinie, um den gesamten eingehenden Datenverkehr zu Routen im Namespace `network-review` zu erlauben.

Verwenden Sie die unter `~/DO280/labs/network-review/allow-from-openshift-ingress.yaml` verfügbare Ressourcendatei. Nehmen Sie die erforderlichen Änderungen vor, um alle Pods im Namespace als Ziel zu verwenden und den Datenverkehr vom standardmäßigen Ingress-Controller zu erlauben.

Da die Kursumgebung die `HostNetwork`-Strategie für Endpunkte verwendet, fügen Sie das Label `network.openshift.io/policy-group=ingress` zum Namespace `default` hinzu. Diese Aktion muss als `admin`-Benutzer ausgeführt werden.

- 5.1. Verwenden Sie einen Texteditor, um die Datei `allow-from-openshift-ingress.yaml` mit einem leeren Pod-Selektor zu aktualisieren, der alle Pods im Namespace als Ziel verwendet. Fügen Sie einen Namespace-Selektor für das Label `network.openshift.io/policy-group=ingress` hinzu.

```
...output omitted...
spec:
  podSelector: {}
  ingress:
    - from:
      - namespaceSelector:
          matchLabels:
            network.openshift.io/policy-group: ingress
```

- 5.2. Erstellen Sie die Netzwerkrichtlinie mit dem Befehl `oc create`.

```
[student@workstation network-review]$ oc create -f \
>   allow-from-openshift-ingress.yaml
networkpolicy.networking.k8s.io/allow-from-openshift-ingress created
```

- 5.3. Fügen Sie als admin-Benutzer `network.openshift.io/policy-group=ingress` zum Namespace `default` hinzu.

```
[student@workstation network-review]$ oc login -u admin -p redhat
Login successful.
...output omitted...
[student@workstation network-policy]$ oc label namespace default \
>   network.openshift.io/policy-group=ingress
namespace/default labeled
```

- 5.4. Vergewissern Sie sich mit dem Befehl `curl`, dass über die Route Zugriff auf den `php-http`-Pod möglich ist. Da im Kursraum ein Cluster mit drei Knoten ausgeführt wird, müssen Sie den Befehl `curl` mehrmals ausführen, um den Zugriff über alle Router-Pods zu überprüfen.

```
[student@workstation network-review]$ for X in {1..4}
>   do
>     curl -s http://php-http.apps.ocp4.example.com | grep "PHP"
>   done
<title>PHP Application</title>
<title>PHP Application</title>
<title>PHP Application</title>
<title>PHP Application</title>
```

6. Erstellen und signieren Sie ein TLS-Zertifikat für die verschlüsselte Version der Anwendung als Benutzer `developer`.

Generieren Sie die Anforderung für die Signierung des Zertifikats (Certificate Signing Request, CSR) für den Hostnamen `php-https.apps.ocp4.example.com`. Speichern Sie die CSR in `~/D0280/labs/network-review/certs/training.csr`.

Erstellen Sie mit der CSR ein Zertifikat, und speichern Sie es in `~/D0280/labs/network-review/certs/training.crt`. Übergeben Sie als Argumente das unter `~/D0280/labs/network-review/certs/training-CA.pem` verfügbare CA-Zertifikat und die CSR, um das Zertifikat zu generieren.

Sie können die Textdatei `/D0280/labs/network-review/certs/openssl-commands.txt` als Hilfe verwenden. Diese Datei enthält die Befehle, um die Anforderung für

Kapitel 5 | Konfigurieren des OpenShift-Netzwerks für Anwendungen

die Signierung des Zertifikats und das Zertifikat zu generieren. Ersetzen Sie die Werte in der Datei, bevor Sie die OpenSSL-Befehle kopieren und ausführen.

6.1. Melden Sie sich als Benutzer **developer** an, um den Rest dieser Übung abzuschließen.

```
[student@workstation network-review]$ oc login -u developer -p developer
Login successful.
...output omitted...
```

6.2. Wechseln Sie zum Verzeichnis `~/D0280/labs/network-review/certs`.

```
[student@workstation network-review]$ cd certs
```

6.3. Generieren Sie die Anforderung für die Signierung des Zertifikats (Certificate Signing Request, CSR) für `php-https.apps.ocp4.example.com`. Geben Sie den Betreff der Anforderung in einer Zeile ein. Alternativ können Sie die Option `-subj` mitsamt dem Inhalt entfernen. Dieser Befehl fordert Sie auf, die Werte einzugeben. Achten Sie darauf, einen allgemeinen Namen (Common Name, CN) für `php-https.apps.ocp4.example.com` anzugeben.



Anmerkung

Stellen Sie sicher, dass nach dem abschließenden Schrägstrich der Organisation (Red Hat) und dem allgemeinen Namen (CN) kein Leerzeichen vorhanden ist.

```
[student@workstation certs]$ openssl req -new -key training.key \
> -subj "/C=US/ST=North Carolina/L=Raleigh/O=Red Hat/\
> CN=php-https.apps.ocp4.example.com" \
> -out training.csr
```

Alternativ können Sie die Textdatei `openssl-commands.txt` öffnen. Kopieren Sie den ersten `openssl`-Befehl, und fügen Sie ihn in Ihr Terminal ein. Ersetzen Sie die Platzhalter-Domain durch `apps.ocp4.example.com` und die Ausgabedatei durch `training.csr`.



Anmerkung

Der Befehl generiert keine Ausgabe.

6.4. Generieren Sie das signierte Zertifikat. Beachten Sie die Verwendung der Optionen `-CA` und `-CAkey` für das Signieren des Zertifikats bei der CA.

```
[student@workstation certs]$ openssl x509 -req -in training.csr \
> -CA training-CA.pem -CAkey training-CA.key -CAcreateserial \
> -passin file:passphrase.txt \
> -out training.crt -days 3650 -sha256 -extfile training.ext
Signature ok
subject=C = US, ST = North Carolina, L = Raleigh, O = Red Hat, CN = php-
https.apps.ocp4.example.com
Getting CA Private Key
```

Kapitel 5 | Konfigurieren des OpenShift-Netzwerks für Anwendungen

Alternativ können Sie den zweiten openssl-Befehl in der Datei `openssl-commands.txt` kopieren und in Ihr Terminal einfügen. Ersetzen Sie die CSR-Datei durch `training.csr`, die CA durch `training-CA.pem` und das Ausgabezertifikat durch `training.crt`.

- 6.5. Überprüfen Sie, ob das neu erstellte Zertifikat und der Schlüssel im aktuellen Verzeichnis vorhanden sind.

```
[student@workstation certs]$ ls -l
total 36
-rw-rw-r-- 1 student student 566 abr 26 07:43 openssl-commands.txt
-rw-rw-r-- 1 student student 33 jun 15 22:20 passphrase.txt
-rw----- 1 student student 1743 jun 15 22:20 training-CA.key
-rw-r--r-- 1 student student 1334 jun 15 22:20 training-CA.pem
-rw-rw-r-- 1 student student 41 jun 15 22:33 training-CA.srl
-rw-rw-r-- 1 student student 1395 jun 15 22:33 training.crt
-rw-rw-r-- 1 student student 1021 jun 15 22:33 training.csr
-rw-r--r-- 1 student student 352 jun 15 22:20 training.ext
-rw----- 1 student student 1679 jun 15 22:20 training.key
```

- 6.6. Kehren Sie zum Verzeichnis `network-review` zurück. Dies ist wichtig, da der nächste Schritt die Erstellung einer Route mit dem signierten Zertifikat umfasst.

```
[student@workstation certs]$ cd ~/D0280/labs/network-review
```

7. Erstellen Sie ein OpenShift-TLS-Secret mit dem Namen `php-certs` im Projekt `network-review`. Verwenden Sie die Datei `~/D0280/labs/network-review/certs/training.crt` für das Zertifikat und die Datei `~/D0280/labs/network-review/certs/training.key` für den Schlüssel.

- 7.1. Erstellen Sie mit dem Befehl `oc create secret` das TLS-Secret `php-certs`. Übergeben Sie `training.csr` als Zertifikat und `training.key` als Schlüssel.

```
[student@workstation network-review]$ oc create secret tls php-certs \
>   --cert certs/training.crt \
>   --key certs/training.key
secret/php-certs created
```

- 7.2. Rufen Sie die Liste der Secrets ab, um zu überprüfen, ob sie vorhanden ist.

```
[student@workstation network-review]$ oc get secrets
NAME          TYPE           DATA   AGE
...output omitted...
php-certs     kubernetes.io/tls    2      93s
```

8. Verwenden Sie die unter `~/D0280/labs/network-review/php-https.yaml` verfügbare Ressourcendatei, um die gesicherte Version der PHP-Anwendung bereitzustellen. Stellen Sie die Anwendung im Projekt `network-review` bereit.

Nehmen Sie vor der Bereitstellung der Anwendung die erforderlichen Änderungen an der Ressourcendatei vor. Ändern Sie:

- Den Speicherort des Containers

Kapitel 5 | Konfigurieren des OpenShift-Netzwerks für Anwendungen

- Den Port, den die Anwendung überwacht
- Den Namen des Secret, das als Volume gemountet werden soll

8.1. Aktualisieren Sie in einem Texteditor die Datei `php-https.yaml` wie folgt:

- Suchen Sie den Eintrag `image`. Legen Sie ihn so fest, dass das Container-Image unter `quay.io/redhattraining/php-ssl:v1.1` verwendet wird.

```
...output omitted...
cpu: '0.5'
image: 'quay.io/redhattraining/php-ssl:v1.1'
name: php-https
...output omitted...
```

- Suchen Sie den Eintrag `containerPort`. Legen Sie den Wert auf **8443** fest, der dem gesicherten Endpunkt entspricht.

```
...output omitted...
name: php-https
ports:
- containerPort: 8443
  name: php-https
...output omitted...
```

- Suchen Sie den Eintrag `secretName`. Legen Sie ihn auf `php-certs` fest. Das ist der Name des Secret, das Sie in einem vorherigen Schritt erstellt haben.

```
...output omitted...
volumes:
- name: tls-certs
  secret:
    secretName: php-certs
...output omitted...
```

Nachdem Sie die Änderungen vorgenommen haben, speichern und beenden Sie die Datei.

8.2. Stellen Sie die abgesicherte Anwendung mit dem Befehl `oc create` bereit. Dadurch werden eine Bereitstellung und ein Service erstellt.

```
[student@workstation network-review]$ oc create -f php-https.yaml
deployment.apps/php-https created
service/php-https created
```

8.3. Warten Sie einen Moment, und führen Sie dann den Befehl `oc get pods` aus, um zu überprüfen, ob der Pod `php-https` ausgeführt wird.

```
[student@workstation network-review]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
php-https-6cb58c847b-7qsbd   1/1     Running   0          8m11s
php-https-84498cd794-hvf7w   1/1   Running  0          26s
```

9. Erstellen Sie eine gesicherte Passthrough-Route mit dem Namen `php-https` und dem Hostnamen `php-https.apps.ocp4.example.com`, um auf die gesicherte Version der Anwendung zuzugreifen.

Greifen Sie auf dem Rechner `workstation` mit Firefox auf `https://php-https.apps.ocp4.example.com` zu. Akzeptieren Sie das signierte Zertifikat, und bestätigen Sie, dass die Anwendung verfügbar ist.

- 9.1. Führen Sie den Befehl `oc create route` aus, um eine Passthrough-Route für den Zugriff auf die Anwendung zu erstellen. Geben Sie der Route den Hostnamen `php-https.apps.ocp4.example.com`. Geben Sie mit der Option `port` den gesicherten Port 8443 an.

```
[student@workstation network-review]$ oc create route passthrough php-https \
>   --service php-https --port 8443 --hostname php-https.apps.ocp4.example.com
route.route.openshift.io/php-https created
```

- 9.2. Rufen Sie den Namen der Route ab, und kopieren Sie ihn in die Zwischenablage.

```
[student@workstation network-review]$ oc get routes
NAME      HOST/PORT            ... SERVICES    PORT  TERMINATION
php-http   php-http.apps.ocp4.example.com  ...  php-http   8080
php-https  php-https.apps.ocp4.example.com ...  php-https  8443  passthrough
```

- 9.3. Öffnen Sie Firefox auf dem Rechner `workstation` und öffnen Sie `https://php-https.apps.ocp4.example.com`.

Akzeptieren Sie das signierte Zertifikat, und überprüfen Sie, ob die gesicherte Version der Anwendung angezeigt wird.

About this application

🔒 The application is currently served over TLS

- **Current system load:** 1
- **Number of connections:** 0
- **Memory usage:** 8 Mb

10. *Optionaler Schritt:* Führen Sie den Befehl `curl` auf dem Rechner `workstation`, aus um die HTTPS-Version der Anwendung zu überprüfen.

Übergeben Sie das CA-Zertifikat an den `curl`-Befehl, um die sichere Verbindung zu validieren.

Verwenden Sie die Option `--cacert`, um das CA-Zertifikat an den Befehl `curl` zu übergeben.

```
[student@workstation network-review]$ curl -v --cacert certs/training-CA.pem \
>   https://php-https.apps.ocp4.example.com
...output omitted...
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
```

```
...output omitted...
* Server certificate:
* subject: C=US; ST=North Carolina; L=Raleigh; O=Red Hat; \
CN=php-https.apps.ocp4.example.com
...output omitted...
The application is currently served over TLS      </span></strong>
...output omitted...
```

11. Kehren Sie zum Benutzerverzeichnis zurück, da der Befehl `lab network-review finish` den Ordner `network-review` löscht.

```
[student@workstation network-review]$ cd
```

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.

```
[student@workstation ~]$ lab network-review grade
```

Beenden

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab network-review finish
```

Hiermit ist die praktische Übung beendet.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- OpenShift implementiert Software-Defined Networking (SDN), um die Netzwerkinfrastruktur des Clusters zu verwalten. SDN entkoppelt die Software, die den Datenverkehr verarbeitet, von den zugrunde liegenden Mechanismen, die den Datenverkehr weiterleiten.
- Kubernetes stellt Services bereit, mit denen Pods unter einer allgemeinen Zugriffsroute logisch gruppiert werden können. Services fungieren als Load Balancer vor einem oder mehreren Pods.
- Services verwenden Selektoren (Labels), die angeben, auf welchen Pods der Service ausgeführt wird.
- Es gibt zwei Arten von Routen: Gesicherte und ungesicherte. Gesicherte Routen verschlüsseln den Datenverkehr mit TLS-Zertifikaten, und ungesicherte Routen leiten den Datenverkehr über eine unverschlüsselte Verbindung weiter.

Gesicherte Routen unterstützen drei Modi: Edge, Passthrough und Wiederverschlüsselung.

- Netzwerkrichtlinien steuern den Netzwerkdatenverkehr zu Pods. Im SDN können logische Zonen erstellt werden, um den Datenverkehr zwischen Pods in beliebigen Namespaces zu trennen.

Kapitel 6

Steuern der Pod-Zuordnung (Scheduling)

Ziel

Steuern der Knoten, auf denen ein Pod ausgeführt wird

Ziele

- Beschreiben der Algorithmen für die Pod-Zuordnung, der Methoden zur Steuerung der Zuordnung und Anwenden dieser Methoden
- Einschränken der Ressourcen, die von Containern, Pods und Projekten beansprucht werden
- Steuern der Anzahl von Replikaten eines Pods, Angeben der Anzahl von Replikaten in einer Bereitstellung, manuelles Skalieren der Anzahl der Replikate und Erstellen einer Horizontal Pod Autoscaler (HPA)-Ressource.

Abschnitte

- Steuern des Pod-Zuordnungsverhaltens (und angeleitete Übung)
- Beschränken der Ressourcennutzung einer Anwendung (und angeleitete Übung)
- Skalieren einer Anwendung (und angeleitete Übung)

Praktische Übung

Steuern der Pod-Zuordnung (Scheduling)

Steuern des Pod-Zuordnungsverhaltens

Ziele

Nach Abschluss dieses Kapitels sollten Sie die Algorithmen für die Pod-Zuordnung und die Methoden zur Steuerung der Zuordnung beschreiben und die Methoden anwenden können.

Einführung in den OpenShift-Scheduler-Algorithmus

Der Pod-Scheduler bestimmt die Platzierung neuer Pods an Knoten innerhalb des OpenShift-Clusters. Er ist in hohem Maß konfigurierbar und anpassbar an verschiedene Cluster. Die Standardkonfiguration von Red Hat OpenShift Container Platform unterstützt die allgemein üblichen Rechenzentrumsbegriffe von Regionen und Zonen, durch die Verwendung von Knotenbezeichnungen, Affinitätsregeln und Anti-Affinitätsregeln.

Der Pod-Scheduler-Algorithmus von OpenShift besteht aus einem dreistufigen Prozess:

1. Filtern von Knoten

Der Scheduler filtert die Liste der ausgeführten Knoten, indem jeder Knoten anhand einer Reihe von Prädikaten bewertet wird, wie Verfügbarkeit von Host-Ports, oder ob ein Pod auf einem Knoten mit hoher Disk- oder Arbeitsspeicherbelastung zugeordnet werden kann.

Außerdem kann ein Pod einen Knoten-Selektor definieren, der zu den Labels in den Cluster-Knoten passt. Knoten deren Labels nicht passen, erfüllen nicht die Voraussetzungen.

Ein Pod kann auch Ressourcenanforderungen für Computing-Ressourcen wie etwa Prozessor, Speicher und Storage definieren. Knoten, deren freie Computer-Ressourcen nicht ausreichen, erfüllen nicht die Voraussetzungen.

Bei einer weiteren Filterprüfung wird bewertet, ob die Liste der Knoten Taints aufweist, und wenn ja, ob der Pod über eine zugehörige Tolerierung verfügt, die den Taint akzeptieren kann. Wenn ein Pod den Taint eines Knotens nicht akzeptieren kann, erfüllt der Knoten nicht die Voraussetzungen. Standardmäßig enthalten Control Plane-Knoten den Taint `node-role.kubernetes.io/master:NoSchedule`. Pods ohne passende Tolerierung für diesen Taint werden nicht zu einem Control Plane-Knoten zugeordnet.



Anmerkung

Das Kursumgebung verwendet einen Cluster mit drei Knoten ohne zusätzliche Computing-Knoten. Der Cluster mit drei Knoten ist für die Bare-Metal-Installation in OpenShift Container Platform 4.6 verfügbar. Dieser Clustertyp eignet sich für Umgebungen mit eingeschränkten Ressourcen, z. B. für Far-Edge-Bereitstellungen.

Die Control Plane-Knoten in der Kursumgebung haben den `node-role.kubernetes.io/master:NoSchedule`-Taint nicht. Reguläre Anwendungs-Pods können für die Control Plane-Knoten geplant werden.

Das Endergebnis des Filterschritts ist in der Regel eine kürzere Liste der Knotenkandidaten, die für die Ausführung des Pods geeignet sind. In einigen Fällen wird keiner der Knoten herausgefiltert, d. h. der Pod kann auf einem beliebigen Knoten ausgeführt werden. In

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

anderen Fällen werden alle Knoten herausgefiltert, d. h., der Pod kann erst zugeordnet werden, wenn ein Knoten mit den gewünschten Voraussetzungen verfügbar wird.

Eine vollständige Liste der Prädikate und deren Beschreibungen finden Sie im Abschnitt „References“.

2. Priorisieren der gefilterten Knotenliste

Die Liste von Knotenkandidaten wird anhand mehrerer Prioritätskriterien evaluiert, die zusammen eine gewichtete Wertung ergeben. Je höher die Punktzahl eines Knotens, desto besser eignet er sich für die Ausführung des jeweiligen Pods.

Zu den Kriterien gehören **Affinitätsregeln** und **Anti-Affinitätsregeln**. Knoten mit höherer Affinität zu dem Pod haben eine höhere, Knoten mit höherer Anti-Affinität eine niedrigere Punktzahl.

Affinitäts-Regeln werden allgemein dazu verwendet, zwecks Leistungssteigerung zusammengehörige Knoten dicht beieinander anzuordnen. Ein Beispiel ist die Verwendung desselben Netzwerk-Backbones für Pods, die sich gegenseitig synchronisieren.

Anti-Affinitätsregeln werden allgemein dazu verwendet, zusammengehörige Pods zur Verbesserung der Hochverfügbarkeit nicht zu dicht beieinander anzuordnen. Zum Beispiel vermeidet man es, alle Pods aus der derselben Anwendung demselben Knoten zuzuordnen.

3. Auswählen des am besten geeigneten Knotens

Die Kandidatenliste wird nach diesen Punktzahlen gefiltert, und der Knoten mit der höchsten Punktzahl wird als Host des Pods ausgewählt. Wenn mehrere Knoten die gleiche Punktzahl erreichen, wird nach dem Round-Robin-Verfahren ein Knoten ausgewählt.

Der Scheduler ist flexibel und kann an erweiterte Zuordnungssituationen angepasst werden. Der Schwerpunkt dieses Kurses liegt auf der Pod-Platzierung mithilfe von Knoten-Labels und Knoten-Selektoren, Pods können aber auch mit Pod-Affinitäts- und -Anti-Affinitätsregeln sowie mit Knoten-Affinitäts- und -Anti-Affinitätsregeln platziert werden. Das Anpassen des Scheduler und diese alternativen Pod-Platzierungsszenarien werden im Rahmen dieses Kurses nicht behandelt.

Zuordnung und Topologie

Eine allgemein übliche Topologie für große Rechenzentren, die z. B. von Cloud-Anbietern verwendet wird, ist die Organisation in **Regionen** und **Zonen**:

- Eine **Region** ist eine Gruppe von Hosts innerhalb eines begrenzten geographischen Gebiets, wodurch Hochgeschwindigkeitsverbindungen zwischen ihnen garantiert sind.
- Eine **Zone**, auch als **Verfügbarkeitszone** bezeichnet, ist eine Gruppe von Hosts, die gemeinsam ausfallen könnten, weil sie kritische Infrastrukturkomponenten gemeinsam nutzen, wie einen Netzwerk-Switch, ein Storage-Array oder eine unterbrechungsfreie Stromversorgung (Uninterruptible Power Supply, UPS).

Als Beispiel für Regionen und Zonen kann Amazon Web Services (AWS) genannt werden. AWS hat eine Region in Nordvirginia (us-east-1) mit 6 Verfügbarkeitszonen und eine weitere Region in Ohio (us-east-2) mit 3 Verfügbarkeitszonen. Jede der AWS-Verfügbarkeitszonen kann mehrere Rechenzentren enthalten, die möglicherweise aus Hunderttausenden Servern bestehen.

Die Standardkonfiguration des OpenShift-Pod-Schedulers unterstützt diese Art von Cluster-Topologie durch die Definition von Prädikaten, die auf `region-` und `zone-`Labels basieren. Die Prädikate werden so definiert, dass:

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

- Von derselben Bereitstellung erstellte Pod-Replikate sollten solchen Knoten zugeordnet werden, die für das Label `region` denselben Wert aufweisen.
- Pod-Replikate Knoten zugeordnet werden, die unterschiedliche Werte für das Label `zone` aufweisen.

Nachfolgende Abbildung zeigt das Beispiel einer Topologie, die aus mehreren Regionen besteht, jede mit mehreren Zonen, und jede Zone mit mehreren Knoten:

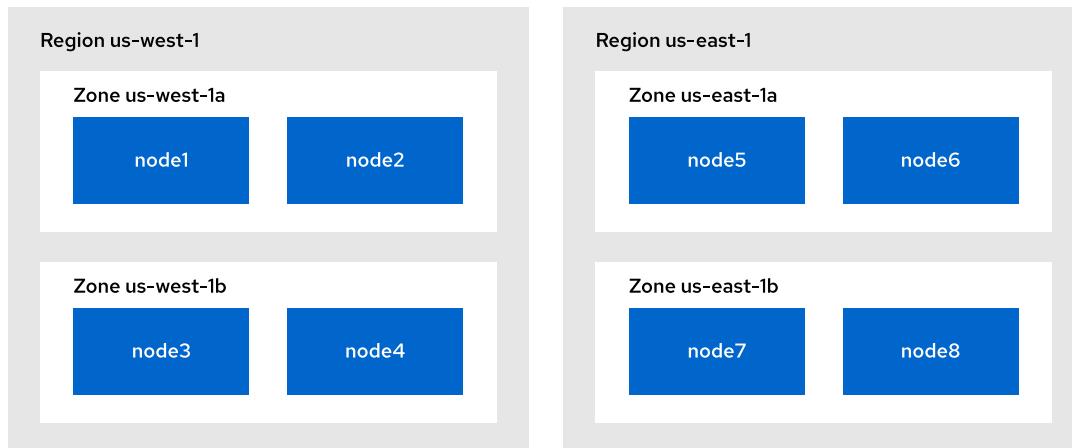


Abbildung 6.1: Beispiel einer Clustertopologie mit Regionen und Zonen

Kennzeichnen von Knoten

Als OpenShift-Cluster-Administrator können Sie Ihren Knoten zusätzliche Labels hinzufügen. So können Sie beispielsweise Knoten mit dem Label `env` und den Werten `dev`, `qa` oder `prod` kennzeichnen, sodass Entwicklungs-, Qualitätssicherungs- und Produktions-Workloads auf einer bestimmten Teilmenge von Knoten bereitgestellt werden. Die von Ihnen ausgewählten Labels sind willkürlich, Sie müssen jedoch die Labels und die zugehörigen Werte für Ihre Entwickler veröffentlichen, damit sie ihre Anwendungen entsprechend konfigurieren können.

Verwenden Sie den Befehl `oc label` als Cluster-Administrator, um ein Knoten-Label sofort hinzuzufügen, zu aktualisieren oder zu entfernen. Mit dem folgenden Befehl kennzeichnen Sie beispielsweise einen Knoten mit `env=dev`:

```
[user@host ~]$ oc label node node1.us-west-1.compute.internal env=dev
```

Mit der Option `--overwrite` ändern Sie ein vorhandenes Label:

```
[user@host ~]$ oc label node node1.us-west-1.compute.internal env=prod --overwrite
```

Geben Sie den Namen des Labels, gefolgt von einem Bindestrich wie z. B. `env-` an, um ein Label zu entfernen:

```
[user@host ~]$ oc label node node1.us-west-1.compute.internal env-
```



Wichtig

Bei Labels und deren Werten wird die Groß-/Kleinschreibung berücksichtigt.
Ein Anwendungsknoten-Selektor muss mit der Groß-/Kleinschreibung des tatsächlichen Labels und dem auf den Knoten angewendeten Wert übereinstimmen.

Mit dem Befehl `oc nodes` und der Option `--show-labels` zeigen Sie die einem Knoten zugeordneten Labels unter Berücksichtigung der Groß-/Kleinschreibung an:

```
[user@host ~]$ oc get node node2.us-west-1.compute.internal --show-labels
NAME           ... ROLES   ... LABELS
node2.us-west-1.compute.internal ... worker ... beta.kubernetes.io/
arch=amd64,beta.kubernetes.io/instance-type=m4.xlarge,beta.kubernetes.io/
os=linux,tier=gold,failure-domain.beta.kubernetes.io/region=us-
west-1,failure-domain.beta.kubernetes.io/zone=us-west-1c,kubernetes.io/
arch=amd64,kubernetes.io/hostname=node2,kubernetes.io/os=linux,node-
role.kubernetes.io/worker=,node.openshift.io/os_id=rhcos
```

Beachten Sie, dass ein Knoten möglicherweise über mehrere von OpenShift zugewiesenen Standard-Labels verfügt. Labels, deren Schlüssel das Suffix `kubernetes.io` enthalten, sollten nicht durch einen Cluster-Administrator geändert werden, denn sie werden intern vom Scheduler verwendet. Die in diesen Beispielbefehlen gezeigten Knoten verwenden das AWS Full-Stack-Automatisierungs-Setup.

Cluster-Administratoren können auch die Option `-L` verwenden, um den Wert eines einzelnen Labels zu bestimmen. Beispiel:

```
[user@host ~]$ oc get node -L failure-domain.beta.kubernetes.io/region
NAME           ... ROLES   ... REGION
ip-10-0-131-214.us-west-1.compute.internal ... master   ... us-west-1
ip-10-0-139-250.us-west-1.compute.internal ... worker   ... us-west-1
ip-10-0-141-144.us-west-1.compute.internal ... master   ... us-west-1
ip-10-0-152-57.us-west-1.compute.internal ... master   ... us-west-1
ip-10-0-154-226.us-west-1.compute.internal ... worker   ... us-west-1
```

Mehrere `-L`-Optionen im selben `oc get`-Befehl werden unterstützt. Beispiel:

```
[user@host ~]$ oc get node -L failure-domain.beta.kubernetes.io/region \
>     -L failure-domain.beta.kubernetes.io/zone -L env
NAME           ... REGION   ZONE      ENV
ip-10-0-131-214.us-west-1.compute.internal ... us-west-1  us-west-1b
ip-10-0-139-250.us-west-1.compute.internal ... us-west-1  us-west-1b  dev
ip-10-0-141-144.us-west-1.compute.internal ... us-west-1  us-west-1b
ip-10-0-152-57.us-west-1.compute.internal ... us-west-1  us-west-1c
ip-10-0-154-226.us-west-1.compute.internal ... us-west-1  us-west-1c
```

Kennzeichnen von Rechnersätzen

Knoten-Labels sind zwar persistent, aber wenn Ihr OpenShift-Cluster MachineSets enthält, sollten Sie auch Labels zur Rechnersatz-Konfiguration hinzufügen. Dadurch wird sichergestellt, dass neue Rechner (und die von ihnen generierten Knoten) auch die gewünschten Labels enthalten. Rechnersätze existieren in Clustern mit Full-Stack-Automatisierung und in bestimmten Clustern

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

mit einer bereits vorhandenen Infrastruktur und einer Integration von Cloud-Anbietern. Bare-Metal-Cluster verwenden keine Rechnersätze.

Sie können die Beziehung zwischen Rechnern und Knoten ermitteln, indem Sie Rechner im Namespace `openshift-machine-api` auflisten und die Option `-o wide` angeben:

```
[user@host ~]$ oc get machines -n openshift-machine-api -o wide
NAME          ... NODE
...output omitted...
ocp-qz7hf-worker-us-west-1b-rvx6w ... ip-10-0-139-250.us-west-1.compute.internal
ocp-qz7hf-worker-us-west-1c-v4n4n ... ip-10-0-154-226.us-west-1.compute.internal
```

Rechner, die für Worker-Knoten verwendet werden, sollten aus einem Rechnersatz stammen. Der Name eines Rechners enthält den Namen des Rechnersatzes, aus dem er generiert wurde. Verwenden Sie den folgenden Befehl, um Rechnersätze auflisten:

```
[user@host ~]$ oc get machineset -n openshift-machine-api
NAME        DESIRED  CURRENT  READY  AVAILABLE ...
ocp-qz7hf-worker-us-west-1b  1         1         1         1         ...
ocp-qz7hf-worker-us-west-1c  1         1         1         1         ...
```

Bearbeiten Sie den Rechnersatz so, dass neue Rechner, die daraus generiert werden, das gewünschte Label oder die gewünschten Labels aufweisen. Durch das Ändern eines Rechnersatzes werden keine Änderungen an bestehenden Rechnern oder Knoten vorgenommen. Verwenden Sie den folgenden Befehl, um einen Rechnersatz zu bearbeiten:

```
[user@host ~]$ oc edit machineset ocp-qz7hf-worker-us-west-1b \
>   -n openshift-machine-api
```

Die unten hervorgehobenen Zeilen zeigen, wo ein Label in einem Rechnersatz hinzugefügt werden sollte:

```
...output omitted...
spec:
  metadata:
    creationTimestamp: null
  labels:
    env: dev
  providerSpec:
...output omitted...
```

Steuern der Pod-Platzierung

Infrastrukturberechtigte Pods in OpenShift-Clustern sind oft so konfiguriert, dass sie auf Control Plane-Knoten ausgeführt werden. Beispiele hierfür sind Pods für den DNS-Operator, den OAuth-Operator und den OpenShift-API-Server. In einigen Fällen wird dies durch die Verwendung des Knoten-Selektors `node-role.kubernetes.io/master: ''` bei der Konfiguration eines DaemonSets oder eines Replikatsatzes erreicht.

Ebenso müssen manche Benutzeranwendungen möglicherweise in einer bestimmten Knoten-Gruppe ausgeführt werden. Beispielsweise sorgen bestimmte Knoten bei bestimmten Workload-Typen für eine Hardware-Beschleunigung, oder der Cluster-Administrator möchte

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

Produktionsanwendungen nicht mit Entwicklungsanwendungen mischen. Verwenden Sie Knoten-Labels und Knoten-Selektoren für diese Art von Szenarien.

Ein Knoten-Selektor ist Teil einer einzelnen Pod-Definition. Definieren Sie einen Knoten-Selektor in einer Ressource für die Bereitstellung, damit jeder neue Pod, der aus dieser Ressource generiert wird, den gewünschten Knoten-Selektor aufweist. Wenn Ihre Ressource für die Bereitstellung unter Versionskontrolle steht, dann ändern Sie die Ressourcendatei, und wenden Sie die Änderungen mit dem Befehl `oc apply -f` an.

Alternativ kann ein Knoten-Selektor entweder mit dem Befehl `oc edit` oder mit dem Befehl `oc patch` hinzugefügt bzw. geändert werden. Um beispielsweise die Bereitstellung `myapp` so zu konfigurieren dass ihre Pods nur auf solchen Knoten ausgeführt werden, die über das Label `env=qa` verfügen, verwenden Sie den Befehl `oc edit`:

```
[user@host ~]$ oc edit deployment/myapp

...
spec:
  ...
  template:
    metadata:
      annotations:
        openshift.io/generated-by: OpenShiftNewApp
      creationTimestamp: null
      labels:
        deployment: myapp
    spec:
      nodeSelector:
        env: dev
      containers:
        - image: quay.io/redhattraining/scaling:v1.0
...

```

Mit dem folgenden `oc patch`-Befehl erreichen Sie dasselbe:

```
[user@host ~]$ oc patch deployment/myapp --patch \
>   '{"spec":{"template":{"spec":{"nodeSelector":{"env":"dev"}}}}}'
```

Ob Sie nun den Befehl `oc edit` oder den Befehl `oc patch` verwenden, die Änderung löst eine neue Bereitstellung aus, und die neuen Pods werden dem Knoten-Selektor entsprechend zugeordnet.

Konfigurieren eines Knoten-Selektors für ein Projekt

Wenn ein Cluster-Administrator nicht möchte, dass Entwickler den Knoten-Selektor für ihre Pods steuern, sollte ein Standard-Knoten-Selektor für die Projektressource konfiguriert werden. Ein Cluster-Administrator kann entweder beim Erstellen eines Projekts einen Knoten-Selektor definieren, oder er kann einen Knoten-Selektor hinzufügen oder aktualisieren, nachdem ein Projekt erstellt wurde. Mit dem Befehl `oc adm new-project` fügen Sie den Knoten-Selektor beim Erstellen des Projekts hinzu. Der folgende Befehl erstellt beispielsweise ein neues Projekt mit dem Namen `demo`, in dem alle Pods auf Knoten bereitgestellt werden, die das Label `tier=1` aufweisen.

```
[user@host ~]$ oc adm new-project demo --node-selector "tier=1"
```

To configure a default node selector for an existing project, add an annotation to the namespace resource with the `openshift.io/node-selector` key. Mit dem Befehl `oc annotate` kann eine Knoten-Selektor-Anmerkung hinzugefügt, geändert oder entfernt werden:

```
[user@host ~]$ oc annotate namespace demo \
>   openshift.io/node-selector="tier=2" --overwrite
```

Skalieren der Anzahl der Pod-Replikate

Die meisten Ressourcen für die Bereitstellung beginnen mit der Erstellung eines einzelnen Pods, die Anzahl der Replikate (oder Kopien) eines Pods wird jedoch häufig erhöht. Dies wird durch die Skalierung der Bereitstellung erreicht. Verschiedene Methoden für die Skalierung werden später behandelt, aber eine Methode verwendet den Befehl `oc scale`. Die Anzahl der Pods in der Bereitstellung `myapp` kann beispielsweise mit dem folgenden Befehl auf drei skaliert werden:

```
[user@host ~]$ oc scale --replicas 3 deployment/myapp
```



Literaturhinweise

Weitere Informationen finden Sie im Kapitel *Controlling pod placement onto nodes (scheduling)* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Nodes* unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/nodes/index#controlling-pod-placement-onto-nodes-scheduling

Regionen und Verfügbarkeitszonen von Amazon Web Services

https://aws.amazon.com/about-aws/global-infrastructure/regions_az/

► Angeleitete Übung

Steuern des Pod-Zuordnungsverhaltens

In dieser Übung konfigurieren Sie eine Anwendung so, dass sie auf einer Teilmenge der Cluster-Computing-Knoten ausgeführt wird.

Ergebnisse

Sie sollten in der Lage sein, die OpenShift-Befehlszeilenschnittstelle zu verwenden, um:

- Einem Knoten ein neues Label hinzuzufügen
- Pods auf Knoten bereitzustellen, die mit einem bestimmten Label übereinstimmen
- Ein Label von einem Knoten zu entfernen
- Fehler zu beheben, wenn Pods nicht auf einem Knoten bereitgestellt werden können

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt ein Projekt für die Übung.

```
[student@workstation ~]$ lab schedule-pods start
```

Anweisungen

► 1. Erstellen Sie als Benutzer `developer` ein neues Projekt mit dem Namen `schedule-pods`.

1.1. Melden Sie sich als Benutzer `developer` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

1.2. Erstellen Sie ein neues Projekt mit dem Namen `schedule-pods`.

```
[student@workstation ~]$ oc new-project schedule-pods
Now using project "schedule-pods" on server "https://api.ocp4.example.com".
...output omitted...
```

► 2. Stellen Sie eine Testanwendung bereit, und skalieren Sie sie.

2.1. Erstellen Sie mit dem Container unter `quay.io/redhattraining/hello-world-nginx:v1.0` eine neue Anwendung mit dem Namen `hello`.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
[student@workstation ~]$ oc new-app --name hello \
>   --docker-image quay.io/redhattraining/hello-world-nginx:v1.0
...output omitted...
--> Creating resources ...
  imagestream.image.openshift.io "hello" created
  deployment.apps "hello" created
  service "hello" created
--> Success
...output omitted...
```

- 2.2. Erstellen Sie eine Route für die Anwendung.

```
[student@workstation ~]$ oc expose svc/hello
route.route.openshift.io/hello exposed
```

- 2.3. Skalieren Sie die Anwendung manuell so, dass vier ausgeführte Pods vorhanden sind.

```
[student@workstation ~]$ oc scale --replicas 4 deployment/hello
deployment.apps/hello scaled
```

- 2.4. Überprüfen Sie, ob die vier ausgeführten Pods auf die beiden Knoten verteilt sind.

NAME	READY	STATUS	...	IP	NODE	...
hello-6c4984d949-78qsp	1/1	Running	...	10.9.0.30	master02	...
hello-6c4984d949-cf6tb	1/1	Running	...	10.10.0.20	master01	...
hello-6c4984d949-kwgbg	1/1	Running	...	10.8.0.38	master03	...
hello-6c4984d949-mb8z7	1/1	Running	...	10.10.0.19	master01	...

**Anmerkung**

Abhängig von der vorhandenen Last auf jedem Knoten kann Ihre Ausgabe abweichen. Obwohl der Scheduler versucht, die Pods zu verteilen, ist die Verteilung möglicherweise nicht gleichmäßig.

- 3. Bereiten Sie die Knoten so vor, dass Anwendungs-Workloads an Entwicklungs- oder Produktionsknoten verteilt werden können, indem Sie das Label `env` zuweisen. Weisen Sie das Label `env=dev` dem `master01`-Knoten zu und das Label `env=prod` dem `master02`-Knoten.

- 3.1. Melden Sie sich als Benutzer `admin` bei Ihrem OpenShift-Cluster an. Ein regulärer Benutzer verfügt nicht über die Berechtigung, Informationen zu Knoten anzuzeigen und kann Knoten nicht mit Labels versehen.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
```

- 3.2. Stellen Sie sicher, dass keiner der Knoten das Label `env` verwendet.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
[student@workstation ~]$ oc get nodes -L env
NAME      STATUS    ROLES      AGE      VERSION      ENV
master01   Ready     master,worker  5d18h   v1.19.0+a5a0987
master02   Ready     master,worker  5d18h   v1.19.0+a5a0987
master03   Ready     master,worker  5d18h   v1.19.0+a5a0987
```

- 3.3. Fügen Sie das Label `env=dev` zum `master01`-Knoten hinzu, um anzugeben, dass es sich um einen Entwicklungsknoten handelt.

```
[student@workstation ~]$ oc label node master01 env=dev
node/master01 labeled
```

- 3.4. Fügen Sie das Label `env=prod` zum `master02`-Knoten hinzu, um anzugeben, dass es sich um einen Produktionsknoten handelt.

```
[student@workstation ~]$ oc label node master02 env=prod
node/master02 labeled
```

- 3.5. Überprüfen Sie, ob die Knoten das korrekte `env`-Label aufweisen. Notieren Sie sich den Knoten mit dem Label `env=dev`, da Sie später prüfen, ob die Anwendungs-Pods auf diesem Knoten bereitgestellt wurden.

```
[student@workstation ~]$ oc get nodes -L env
NAME      STATUS    ROLES      AGE      VERSION      ENV
master01   Ready     master,worker  5d18h   v1.19.0+a5a0987   dev
master02   Ready     master,worker  5d18h   v1.19.0+a5a0987   prod
master03   Ready     master,worker  5d18h   v1.19.0+a5a0987
```

- 4. Wechseln Sie zurück zum Benutzer `developer`, und ändern Sie die Anwendung `hello` so, dass sie auf dem Entwicklungsknoten bereitgestellt wird. Nachdem Sie diese Änderung überprüft haben, löschen Sie das Projekt `schedule-pods`.

- 4.1. Melden Sie sich als Benutzer `developer` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer
Login successful.
...output omitted...
Using project "schedule-pods".
```

- 4.2. Ändern Sie die Ressource `deployment` für die Anwendung `hello` so, dass ein Entwicklungsknoten ausgewählt wird. Stellen Sie sicher, dass der Gruppe `spec` im Abschnitt `template` der Knoten-Selektor hinzugefügt wird.

```
[student@workstation ~]$ oc edit deployment/hello
```

Fügen Sie der Bereitstellungsressource die hervorgehobenen Zeilen mit der angezeigten Einrückung hinzu.

```
...output omitted...
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
dnsPolicy: ClusterFirst
nodeSelector:
  env: dev
restartPolicy: Always
...output omitted...
```

Nachdem Sie Ihre Änderungen gespeichert haben, wird die folgende Ausgabe von oc edit angezeigt.

```
deployment.apps/hello edited
```

- 4.3. Überprüfen Sie, ob die Anwendungs-Pods auf dem Knoten mit dem Label env=dev bereitgestellt werden. Auch wenn die erneute Bereitstellung etwas Zeit in Anspruch nimmt, müssen die Anwendungs-Pods auf dem master01-Knoten bereitgestellt werden.

```
[student@workstation ~]$ oc get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP          NODE   ...
hello-b556ccf8b-8scxd  1/1     Running   0          80s    10.10.0.14  master01 ...
hello-b556ccf8b-hb24w  1/1     Running   0          77s    10.10.0.16  master01 ...
hello-b556ccf8b-qxlj8  1/1     Running   0          80s    10.10.0.15  master01 ...
hello-b556ccf8b-sdxpj  1/1     Running   0          76s    10.10.0.17  master01 ...
```

- 4.4. Entfernen Sie das Projekt schedule-pods.

```
[student@workstation ~]$ oc delete project schedule-pods
project.project.openshift.io "schedule-pods" deleted
```

- 5. Entfernen Sie das Label env von allen Knoten, um diesen Teil der Übung zu bereinigen.

- 5.1. Melden Sie sich als Benutzer admin bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
```

- 5.2. Entfernen Sie das Label env von allen Knoten.

```
[student@workstation ~]$ oc label node -l env env-
node/master01 labeled
node/master02 labeled
```

- 6. Das Projekt schedule-pods-ts enthält eine Anwendung, die nur auf Knoten ausgeführt wird, die als client=ACME gekennzeichnet sind. Im folgenden Beispiel ist der Anwendungs-Pod ausstehend, und Sie müssen das Problem anhand der folgenden Schritte diagnostizieren:

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

- 6.1. Melden Sie sich als Benutzer developer bei Ihrem OpenShift-Cluster an, und stellen Sie sicher, dass Sie das Projekt schedule-pods-ts verwenden.

```
[student@workstation ~]$ oc login -u developer -p developer
Login successful.
...output omitted...
Using project "schedule-pods-ts".
```

Wenn in der obige Ausgabe nicht angezeigt wird, dass Sie das Projekt schedule-pods-ts verwenden, wechseln Sie zu diesem Projekt.

```
[student@workstation ~]$ oc project schedule-pods-ts
Now using project "schedule-pods-ts" on server
"https://api.ocp4.example.com:6443".
```

- 6.2. Verifizieren Sie, dass die Anwendung den Status Pending aufweist.

```
[student@workstation ~]$ oc get pods
NAME                  READY   STATUS    RESTARTS   AGE
hello-ts-5dbff9f44-w6csj   0/1     Pending   0          6m19s
```

- 6.3. Da ein Pod mit dem Status „pending“ keine Protokolle bereitstellt, überprüfen Sie die Details des Pods mit dem Befehl oc describe pod. Auf diese Weise können Sie feststellen, ob die Beschreibung des Pods nützliche Informationen enthält.

```
[student@workstation ~]$ oc describe pod hello-ts-5dbff9f44-8h7c7
...output omitted...
QoS Class:      BestEffort
Node-Selectors: client=acme
Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
Type      Reason           ...           Message
----      -----           ...           -----
Warning  FailedScheduling ...  0/3 nodes are available: 3 node(s) didn't match
node selector.
```

Entsprechend diesen Informationen sollte der Pod einem Knoten mit dem Label client=acme zugeordnet werden, aber keiner der drei Knoten weist dieses Label auf.

- 6.4. Melden Sie sich als Benutzer admin bei Ihrem OpenShift-Cluster an, und überprüfen Sie das Label des Computing-Knotens. Um dies zu überprüfen, führen Sie oc get nodes -L client aus, um die Details der verfügbaren Knoten aufzulisten.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...

[student@workstation ~]$ oc get nodes -L client
NAME      STATUS    ROLES           AGE     VERSION      CLIENT
master01   Ready     master,worker  10d    v1.19.0+a5a0987  ACME
```

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

In den bereitgestellten Informationen ist angegeben, dass mindestens ein Computing-Knoten das Label `client=ACME` aufweist. Sie haben das Problem gefunden. Die Anwendung muss so geändert werden, dass sie den richtigen Knoten-Selektor verwendet.

- 6.5. Melden Sie sich bei Ihrem OpenShift-Cluster als Entwickler an, und bearbeiten Sie die Bereitstellungsressource für die Anwendung so, dass der richtige Knoten-Selektor verwendet wird

```
[student@workstation ~]$ oc login -u developer -p developer
Login successful.
...output omitted...
```

```
[student@workstation ~]$ oc edit deployment/hello-ts
```

Ändern Sie `acme` wie unten gezeigt in ACME.

```
...output omitted...
dnsPolicy: ClusterFirst
nodeSelector:
  client: ACME
restartPolicy: Always
...output omitted...
```

Nachdem Sie Ihre Änderungen gespeichert haben, wird die folgende Ausgabe von `oc edit` angezeigt.

```
deployment.apps/hello-ts edited
```

- 6.6. Verifizieren Sie, dass der neue Anwendungs-Pod bereitgestellt ist und den Status `Running` aufweist.

```
[student@workstation ~]$ oc get pods
NAME                  READY   STATUS    RESTARTS   AGE
hello-ts-69769f64b4-wwhpc   1/1     Running   0          11s
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab schedule-pods finish
```

Hiermit ist die angeleitete Übung beendet.

Beschränken der Ressourcennutzung einer Anwendung

Ziele

Am Ende dieses Abschnitts sollten Sie in der Lage sein, die von Containern, Pods und Projekten genutzten Ressourcen einzuschränken.

Definieren Ressourcenanforderungen und -beschränkungen für Pods

Eine Pod-Definition kann sowohl Ressourcenanforderungen als auch Ressourceneinschränkungen umfassen:

Ressourcenanforderungen

Dienen Planungszwecken und zeigen an, dass ein Pod nicht ohne die angegebene Menge an Rechenressourcen ausgeführt werden kann. Der Scheduler sucht dann nach einem Knoten mit ausreichend Rechenressourcen, um die Pod-Anforderungen zu erfüllen.

Ressourceneinschränkungen

Werden eingesetzt, um zu vermeiden, dass ein Pod alle Rechenressourcen eines Knotens aufbraucht. Der Knoten, auf dem der Pod ausgeführt wird, konfiguriert die Linux-Kernel-Funktion cgroups, um die Ressourceneinschränkungen für den Pod zu erzwingen.

Ressourcenanforderungen und Ressourcenbeschränkungen sollten für jeden Container in einer Ressource für die Bereitstellung oder für die Bereitstellungskonfiguration definiert werden. Wenn keine Anforderungen und Beschränkungen definiert wurden, dann ist für jeden Container die Zeile `resources: {}` vorhanden.

Ändern Sie die Zeile `resources: {}`, um die gewünschten Anforderungen und/oder Beschränkungen anzugeben. Beispiel:

```
...output omitted...
spec:
  containers:
    - image: quay.io/redhattraining/hello-world-nginx:v1.0
      name: hello-world-nginx
      resources:
        requests:
          cpu: "10m"
          memory: 20Mi
        limits:
          cpu: "80m"
          memory: 100Mi
  status: {}
```

Wenn Sie den Befehl `oc edit` verwenden, um eine Bereitstellung oder eine Bereitstellungskonfiguration zu ändern, sollten Sie auf die korrekte Einrückung achten. Fehler bei der Einrückung können dazu führen, dass der Editor Änderungen nicht speichert. Um Einrückungsprobleme zu vermeiden, können Sie mit dem Befehl `oc set resources`

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

Ressourcenanforderungen und -beschränkungen angeben. Mit dem folgenden Befehl werden dieselben Anforderungen und Beschränkungen wie im vorherigen Beispiel festgelegt:

```
[user@host ~]$ oc set resources deployment hello-world-nginx \
>   --requests cpu=10m,mem=20Mi --limits cpu=80m,mem=100Mi
```

Wenn ein Ressourcenkontingent für eine Ressourcenanforderung gilt, sollte der Pod eine Ressourcenanforderung definieren. Wenn ein Ressourcenkontingent für eine Ressourcenbeschränkung gilt, sollte der Pod ebenfalls eine Ressourcenbeschränkung definieren. Red Hat empfiehlt, Ressourcenanforderungen und -beschränkungen zu definieren, selbst wenn keine Kontingente verwendet werden.

Anzeigen von Anforderungen, Beschränkungen und der tatsächlichen Nutzung

Über die Befehlszeilenschnittstelle von OpenShift können Cluster-Administratoren Informationen zur Rechennutzung auf einzelnen Knoten anzeigen. Der Befehl `oc describe node` zeigt detaillierte Informationen zu einem Knoten an, einschließlich Informationen zu den Pods, die auf dem Knoten ausgeführt werden. Für jeden Pod werden CPU-Anforderungen und -Beschränkungen sowie Arbeitsspeicheranforderungen und -beschränkungen angezeigt. Wenn keine Anforderung oder Beschränkung angegeben wurde, zeigt der Pod eine 0 in dieser Spalte an. Außerdem wird eine Zusammenfassung aller Ressourcenanforderungen und -beschränkungen angezeigt.

```
[user@host ~]$ oc describe node node1.us-west-1.compute.internal
Name:           node1.us-west-1.compute.internal
Roles:          worker
Labels:         beta.kubernetes.io/arch=amd64
                beta.kubernetes.io/instance-type=m4.xlarge
                beta.kubernetes.io/os=linux
...
Non-terminated Pods: (20 in total)
...  Name          CPU Requests ...  Memory Requests  Memory Limits  AGE
...  ---          ----- ...  -----          -----      -----
...  tuned-vdwt4    10m (0%)   ...  50Mi (0%)       0 (0%)       8d
...  dns-default-2rpwf  110m (3%)  ...  70Mi (0%)     512Mi (3%)   8d
...  node-ca-6xwmn  10m (0%)   ...  10Mi (0%)       0 (0%)       8d
...
Resource        Requests      Limits
-----          -----      -----
cpu             600m (17%)  0 (0%)
memory          1506Mi (9%) 512Mi (3%)
...
...output omitted...
```



Anmerkung

In den Zusammenfassungsspalten für Requests und Limits werden die Gesamtsummen der definierten Anforderungen und Beschränkungen angezeigt.

In der obigen Ausgabe ist nur für einen der auf dem Knoten ausgeführten 20 Pods eine Arbeitsspeicherbeschränkung definiert, und diese Beschränkung betrug 512 Mi.

Der Befehl `oc describe node` zeigt Anforderungen und Beschränkungen an, und der Befehl `oc adm top` zeigt die tatsächliche Nutzung an. Wenn beispielsweise ein Pod 10 m CPU anfordert, stellt der Scheduler sicher, dass der Pod auf einem Knoten mit verfügbarer Kapazität platziert wird. Obwohl der Pod 10 m CPU angefordert hat, kann er mehr oder weniger als diesen Wert verwenden, es sei denn, er ist auch durch eine CPU-Beschränkung limitiert. Ähnlich verwendet ein Pod, für den keine Ressourcenanforderungen angegeben sind, trotzdem einige Ressourcen. Der Befehl `oc adm top nodes` zeigt die tatsächliche Nutzung für einen oder mehrere Knoten im Cluster an, und der Befehl `oc adm top pods` zeigt die tatsächliche Nutzung für jeden Pod in einem Projekt an.

```
[user@host ~]$ oc adm top nodes -l node-role.kubernetes.io/worker
NAME                  CPU(cores)   CPU%    MEMORY(bytes)  MEMORY%
node1.us-west-1.compute.internal  519m       14%    3126Mi        20%
node2.us-west-1.compute.internal  167m       4%     1178Mi        7%
```

Anwenden von Kontingenzen

OpenShift Container Platform kann Quoten erzwingen, mit denen die Verwendung zweier Arten von Ressourcen nachverfolgt und eingeschränkt werden kann:

Objektzählung

Die Anzahl der Kubernetes-Ressourcen wie Pods, Services und Routen.

Rechenressourcen

Die Anzahl physischer oder virtueller Hardware-Ressourcen wie CPU-, Speicher- und Storage-Kapazität.

Mit einem Kontingent für die Anzahl der Kubernetes-Ressourcen kann die Stabilität der OpenShift Control Plane verbessert werden, da die Etcd-Datenbank nicht unbegrenzt wachsen kann. Außerdem können Sie mit Kontingenzen für Kubernetes-Ressourcen verhindern, dass andere begrenzte Software-Ressourcen wie IP-Adressen für Services aufgebraucht werden.

Dementsprechend wird durch das Anwenden einer Quote auf die Menge der Rechenressourcen vermieden, dass die Rechenkapazität eines einzelnen Knotens in einem OpenShift-Cluster aufgebraucht wird. Außerdem wird vermieden, dass eine Anwendung andere Anwendungen in einem gemeinsam genutzten Cluster behindert, indem sie die gesamte Clusterkapazität nutzt.

OpenShift verwaltet Kontingente für die Anzahl der Ressourcen und die Nutzung von Rechenressourcen in einem Cluster mithilfe der Ressource `ResourceQuota` oder kurz `quota`. Ein Kontingent bestimmt die Nutzungsbeschränkungen für die Hardwareressourcen eines Projekts. Alle Attribute einer Quote sind optional; das bedeutet, dass jede Ressource, die nicht durch eine Quote eingeschränkt wird, ohne Begrenzung genutzt werden kann.



Anmerkung

Obwohl eine einzelne Kontingentressource alle Kontingente für ein Projekt definieren kann, kann ein Projekt auch mehrere Kontingente enthalten. Beispielsweise kann eine Kontingentressource die Rechenressourcen begrenzen, z. B. die zulässige CPU-Auslastung oder den zulässigen Arbeitsspeicher. Eine andere Kontingentressource kann die Objektanzahl beschränken, beispielsweise die Anzahl der zulässigen Pods oder die Anzahl der zulässigen Services. Der Effekt mehrerer Kontingente ist kumulativ. Es wird jedoch davon ausgegangen, dass zwei unterschiedliche ResourceQuota-Ressourcen eines Projekts nicht versuchen, die Nutzung derselben Kubernetes- oder Rechenressource zu beschränken. Beispielsweise sollten zwei unterschiedliche Kontingente in einem Projekt nicht beide versuchen, die maximale Anzahl der zulässigen Pods zu beschränken.

In der folgenden Tabelle werden einige Ressourcen beschrieben, die durch ein Kontingent nach Anzahl oder Menge beschränkt werden können:

Ressourcenname	Kontingentbeschreibung
pods	Gesamtanzahl der Pods
replicationcontrollers	Gesamtanzahl der Replikationscontroller
services	Gesamtanzahl der Services
secrets	Gesamtanzahl der Geheimnisse
persistentvolumeclaims	Gesamtanzahl der Persistent Volume Claims

In der folgenden Tabelle werden einige Rechenressourcen beschrieben, die durch ein Kontingent eingeschränkt werden können:

Name der Rechenressource	Kontingentbeschreibung
cpu (requests.cpu)	Gesamt-CPU-Nutzung auf allen Containern
memory (requests.memory)	Gesamtspeichernutzung auf allen Containern
storage (requests.storage)	Gesamte Storage-Anforderungen durch Container für alle Anforderungen für ein persistentes Volume

Mithilfe von Kontingentattributen können Ressourcenanforderungen oder -beschränkungen für alle Pods im Projekt nachverfolgt werden. Standardmäßig werden mit Kontingentattributen Ressourcenanforderungen nachverfolgt. Wenn Sie stattdessen die Ressourceneinschränkungen nachverfolgen möchten, stellen Sie dem Namen der Rechenressource das Präfix `limits` voran, z. B. `limits.cpu`.

Das folgende Listing zeigt eine in der YAML-Syntax definierte ResourceQuota-Ressource. In diesem Beispiel werden Kontingente für die Anzahl der Ressourcen und die Verwendung von Rechenressourcen angegeben:

```
apiVersion: v1
kind: ResourceQuota
metadata:
```

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
name: dev-quota
spec:
  hard:
    services: "10"
    cpu: "1300m"
    memory: "1.5Gi"
```

Pod-Ressourcenanforderungen und -einschränkungen verwenden dieselben Ressourceneinheiten. Gi steht beispielsweise für GiB, und m bedeutet Millicores. Ein Millicore entspricht 1/1000 eines einzelnen CPU-Kerns.

Ressourcenkontingente können wie jede andere Ressource von OpenShift Container Platform erstellt werden, d. h. durch Übergabe einer YAML- oder JSON-Ressourcendefinitionsdatei an den Befehl `oc create`:

```
[user@host ~]$ oc create --save-config -f dev-quota.yml
```

Zudem können Ressourcenkontingente mit dem Befehl `oc create quota` erstellt werden. Zum Beispiel:

```
[user@host ~]$ oc create quota dev-quota --hard services=10,cpu=1300,memory=1.5Gi
```

Verwenden Sie den Befehl `oc get resourcequota`, um die verfügbaren Kontingente aufzulisten, und den Befehl `oc describe resourcequota`, um Nutzungsstatistiken zu allen Hardwarebeschränkungen anzuzeigen, die in dem Kontingent definiert sind. Zum Beispiel:

```
[user@host ~]$ oc get resourcequota
NAME          AGE     REQUEST
compute-quota 51s    cpu: 500m/10, memory: 300Mi/1Gi ...
count-quota   28s    pods: 1/3, replicationcontrollers: 1/5, services: 1/2 ...
```

Ohne Argumente zeigt der Befehl `oc describe quota` die kumulativen Beschränkungen an, die für alle `ResourceQuota`-Ressourcen im Projekt festgelegt sind:

```
[user@host ~]$ oc describe quota
Name:           compute-quota
Namespace:      schedule-demo
Resource        Used     Hard
-----
cpu            500m     10
memory         300Mi    1Gi

Name:           count-quota
Namespace:      schedule-demo
Resource        Used     Hard
-----
pods           1        3
replicationcontrollers 1        5
services        1        2
```

Aktive Kontingente können anhand des Namens mit dem Befehl `oc delete` gelöscht werden:

```
[user@host ~]$ oc delete resourcequota QUOTA
```

Wenn eine Quote zum ersten Mal in einem Projekt erstellt wird, wird die Erstellung neuer Ressourcen im Projekt, die gegen eine Quotenbeschränkung verstößen könnten, verhindert, bis die aktualisierten Nutzungsstatistiken berechnet wurden. Nach dem Erstellen des Kontingents und dem Aktualisieren der Nutzungsstatistiken akzeptiert das Projekt die Erstellung neuer Inhalte. Wenn Sie eine neue Ressource erstellen, wird die Kontingentnutzung sofort erhöht. Wenn Sie eine Ressource löschen, wird die Kontingentnutzung bei der nächsten vollständigen Neuberechnung der Kontingentstatistik für das Projekt verringert.

Kontingente werden auf neue Ressourcen angewendet, wirken sich jedoch nicht auf vorhandene Ressourcen aus. Wenn Sie beispielsweise ein Kontingent erstellen, um ein Projekt auf 15 Pods zu beschränken, aber es werden bereits 20 Pods ausgeführt, werden die zusätzlichen 5 Pods, die das Kontingent überschreiten, nicht durch das Kontingent entfernt.



Wichtig

ResourceQuota-Beschränkungen werden auf das gesamte Projekt angewendet. Viele OpenShift-Prozesse wie Builds und Bereitstellungen erstellen jedoch Pods innerhalb des Projekts. Diese Prozesse können fehlschlagen, da beim Starten der Pods das Projektkontingent überschritten wird.

Wenn bei einer Projektänderung das Kontingent für eine Ressourcenanzahl überschritten wird, blockiert OpenShift die Aktion, und der Benutzer erhält eine entsprechende Fehlermeldung.

Wenn bei der Änderung das Kontingent für eine Rechenressource überschritten wird, schlägt der Vorgang jedoch nicht sofort fehl. OpenShift versucht mehrmals, den Vorgang durchzuführen und ermöglicht dem Administrator, das Kontingent zu erhöhen oder eine andere Gegenmaßnahme zu ergreifen, z. B. einen neuen Knoten online zu stellen.



Wichtig

Beim Festlegen eines Kontingents, das die Nutzung von Rechenressourcen für ein Projekt beschränkt, lehnt OpenShift die Erstellung von Pods ab, die keine Ressourcenanforderungen oder -beschränkungen für Rechenressourcen festsetzen. Für die Verwendung der meisten Vorlagen und Builder in einem Projekt, das durch Kontingente beschränkt wird, muss das Projekt auch eine Ressource für den Beschränkungsbereich enthalten, der die Standardwerte für Container-Ressourcenanforderungen festlegt.

Anwenden von Beschränkungsbereichen

LimitRange-Ressourcen (auch als `limit` bezeichnet) definieren die Standard-, Mindest- und Höchstwerte für die Anforderung und Beschränkung von Rechenressourcen für einzelne Pods oder Container im Projekt. Eine Ressourcenanforderung bzw. -beschränkung für einen Pod entspricht der Summe aller zugehörigen Container.

Bedenken Sie zum besseren Verständnis des Unterschieds zwischen einer Ressource für Beschränkungsbereiche und eines Ressourcenkontingents, dass ein Beschränkungsbereich gültige Bereiche und Standardwerte für einen einzelnen Pod definiert, wohingegen ein Ressourcenkontingent nur die Höchstwerte für die Summe aller Pods in einem Projekt definiert. Cluster-Administratoren, die die Ressourcennutzung in einem OpenShift-Cluster verwalten, definieren sowohl Beschränkungen als auch Kontingente für ein Projekt.

Eine Ressource für Beschränkungsbereiche kann auch die Standard-, Mindest- und Höchstwerte für die Storage-Kapazität definieren, die für ein Image, einen Image-Stream oder eine Anforderung eines persistenten Volume angefordert werden. Wenn eine Ressource, die einem Projekt hinzugefügt wird, keine Rechenressourcenanforderung festlegt, wird der durch die Beschränkungsbereiche festgelegte Standardwert für das Projekt verwendet. Wenn eine neue Ressource Rechenressourcenanforderungen oder -beschränkungen festlegt, die unter dem durch die Beschränkungsbereiche des Projekts festgelegten Mindestwert liegen, wird die Ressource nicht erstellt. Wenn also eine neue Ressource Rechenressourcenanforderungen oder -beschränkungen festlegt, die über dem durch die Beschränkungsbereiche des Projekts festgelegten Höchstwert liegen, wird die Ressource nicht erstellt.

Im folgenden Eintrag wird ein Grenzbereich angezeigt, der durch die YAML-Syntax definiert wird:

```
apiVersion: "v1"
kind: "LimitRange"
metadata:
  name: "dev-limits"
spec:
  limits:
    - type: "Pod"
      max: ①
        cpu: "500m"
        memory: "750Mi"
      min: ②
        cpu: "10m"
        memory: "5Mi"
    - type: "Container"
      max: ③
        cpu: "500m"
        memory: "750Mi"
      min: ④
        cpu: "10m"
        memory: "5Mi"
    default: ⑤
      cpu: "100m"
      memory: "100Mi"
    defaultRequest: ⑥
      cpu: "20m"
      memory: "20Mi"
    - type: openshift.io/Image ⑦
      max:
        storage: 1Gi
    - type: openshift.io/ImageStream ⑧
      max:
        openshift.io/image-tags: 10
        openshift.io/images: 20
    - type: "PersistentVolumeClaim" ⑨
      min:
        storage: "1Gi"
      max:
        storage: "50Gi"
```

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

- ❶ Die maximale Menge an CPU und Arbeitsspeicher, die alle Container in einem Pod nutzen können. Ein neuer Pod, der die Obergrenzen überschreitet, wird nicht erstellt. Ein vorhandener Pod, der die Obergrenzen überschreitet, wird neu gestartet.
- ❷ Die minimale Menge an CPU und Arbeitsspeicher, die alle Container in einem Pod nutzen. Ein Pod, der die Mindestanforderungen nicht erfüllt, wird nicht erstellt. Da viele Pods nur einen Container haben, können Sie als Mindestwerte für die Pods dieselben Werte wie für die Container verwenden.
- ❸ Die maximale Menge an CPU und Arbeitsspeicher, die ein einzelner Container in einem Pod nutzen darf. Ein neuer Container, der die Obergrenzen überschreitet, wird nicht erstellt. Ein vorhandener Pod, der die Obergrenzen überschreitet, startet den gesamten Pod neu.
- ❹ Die minimale Menge an CPU und Arbeitsspeicher, die ein einzelner Container in einem Pod nutzen darf. Wenn ein Container die Mindestanforderungen nicht erfüllt, kann der zugehörige Pod nicht erstellt werden.
- ❺ Die standardmäßige Obergrenze für CPU und Arbeitsspeicher, die ein einzelner Container nutzen darf. Dies wird verwendet, wenn für den Container keine Limits für CPU oder Arbeitsspeicher festgelegt sind.
- ❻ Die standardmäßige Menge an CPU und Arbeitsspeicher, die ein einzelner Container anfordert. Dieser Standardwert wird verwendet, wenn für den Container keine CPU-Ressourcenanforderung oder Arbeitsspeicheranforderung festgelegt ist. Wenn CPU- und Arbeitsspeicherkontingente für einen Namespace aktiviert sind, können den Abschnitt `defaultRequest` konfigurieren, um den Start von Pods zu erlauben, selbst wenn die Container keine Ressourcenanforderungen enthalten.
- ❼ Die maximale Größe von Images, die an die interne Registry weitergeleitet werden können.
- ❽ Die maximale Anzahl von Image-Tags und Versionen, auf die eine Image-Stream-Ressource verweisen kann.
- ❾ Die Ober- und Untergrenze für die Größe von Anforderungen für persistente Volumes.

Benutzer können Ressourcen für Beschränkungsbereiche auf dieselbe Weise wie andere OpenShift-Ressourcen erstellen, indem sie eine YAML- oder JSON-Ressourcendefinitionsdatei an den Befehl `oc create` übergeben:

```
[user@host ~]$ oc create --save-config -f dev-limits.yml
```

Red Hat OpenShift Container Platform bietet keinen spezifischen `oc create`-Befehl für Beschränkungsbereiche, im Gegensatz zu Ressourcenkontingenzen. Alternativ können nur YAML- oder JSON-Dateien verwendet werden.

Verwenden Sie den Befehl `oc describe limitrange`, um in einem Projekt erzwungene Beschränkungen anzuzeigen:

```
[user@host ~]$ oc describe limitrange dev-limits
Name:          dev-limits
Namespace:    schedule-demo
Type           Resource        Min   Max   Default Request ...
Pod            cpu            10m   500m  -      ...
Pod            memory         5Mi   750Mi -      ...
Container      memory         5Mi   750Mi 20Mi ...
```

Container	cpu	10m	500m	20m	...
openshift.io/Image	storage	-	1Gi	-	...
openshift.io/ImageStream	openshift.io/image-tags	-	10	-	...
openshift.io/ImageStream	openshift.io/images	-	20	-	...
PersistentVolumeClaim	storage	1Gi	50Gi	-	...

Aktive Beschränzungsbereiche können anhand des Namens mit dem Befehl `oc delete` gelöscht werden:

```
[user@host ~]$ oc delete limitrange dev-limits
```

Nach dem Erstellen eines Beschränzungsbereichs in einem Projekt werden alle Anforderungen zur Erstellung neuer Ressourcen anhand der jeweiligen Beschränzungsbereichsressource im Projekt bewertet. Wenn die neue Ressource gegen den von einer Beschränzungsbereichsressource angegebenen minimalen oder maximalen Wert verstößt, wird die Ressource abgelehnt. Wenn die neue Ressource keinen expliziten Wert angibt und gemäß Beschränkungsangabe ein Standardwert zulässig ist, dann wird der Standardwert als Nutzungswert auf die neue Ressource angewendet.

Alle Anforderungen zur Aktualisierung einer Ressource werden anhand der jeweiligen Grenzbereichsressource im Projekt bewertet. Wenn die aktualisierte Ressource gegen eine Beschränkung verstößt, wird die Aktualisierung abgelehnt.



Wichtig

Vermeiden Sie zu hohe LimitRange- bzw. zu niedrige ResourceQuota-Beschränkungen. Bei einem Verstoß gegen die LimitRange-Beschränkungen wird die Erstellung neuer Pods verhindert, und Fehlermeldungen werden angezeigt. Bei einem Verstoß gegen die ResourceQuota-Beschränkungen wird verhindert, dass Pods auf einem Knoten geplant werden. Pods können dann möglicherweise erstellt werden, verbleiben jedoch im Status „Pending“.

Anwenden von Kontingenten auf mehrere Projekte

Die Ressource ClusterResourceQuota wird ähnlich wie persistente Volumes auf der Cluster-Ebene erstellt und legt Ressourcenbeschränkungen fest, die für mehrere Projekte gelten.

Cluster-Administratoren können auf zweierlei Art bestimmen, welche Projekte Cluster-Ressourcenkontingenten unterliegen sollen:

- Mithilfe der Annotation `openshift.io/requester` zur Angabe des Projektinhabers. Alle Projekte mit einem festgelegten Inhaber unterliegen dem Kontingent.
- Mithilfe eines Selektors. Alle Projekte, deren Label mit dem Selektor übereinstimmen, unterliegen dem Kontingent.

Im Folgenden wird die Erstellung eines Cluster-Ressourcenkontingents für alle Projekte beschrieben, die dem Benutzer qa gehören:

```
[user@host ~]$ oc create clusterquota user-qa \
>   --project-annotation-selector openshift.io/requester=qa \
>   --hard pods=12,secrets=20
```

Im folgenden Beispiel wird die Erstellung eines Cluster-Ressourcenkontingents für alle Projekte beschrieben, denen das Label `environment=qa` zugewiesen wurde:

```
[user@host ~]$ oc create clusterquota env-qa \
>   --project-label-selector environment=qa \
>   --hard pods=10,services=5
```

Projektbenutzer können mit dem Befehl `oc describe QUOTA` die Cluster-Ressourcenkontingente anzeigen, die für das aktuelle Projekt gelten (sofern vorhanden).

Verwenden Sie den Befehl `oc delete`, um ein Cluster-Ressourcenkontingent zu löschen:

```
[user@host ~]$ oc delete clusterquota QUOTA
```



Anmerkung

Um große Overheads beim Sperren zu vermeiden, sollten Sie keine einzelnen Clusterressourcenkontingente für mehr als Hundert Projekte verwenden. Wenn in einem Projekt Ressourcen erstellt oder aktualisiert werden, wird während der Suche nach den anwendbaren Ressourcenquoten das Projekt gesperrt.

Anpassen der Standardprojektvorlage

Cluster-Administratoren können die Standardprojektvorlage anpassen. Wenn ein Benutzer ein neues Projekt erstellt, werden zusätzliche Ressourcen wie Kontingente, Einschränzungsbereiche und Netzwerkrichtlinien erstellt.

Erstellen Sie als Cluster-Administrator eine neue Projektvorlage mit dem Befehl `oc adm create-bootstrap-project-template`, und leiten Sie die Ausgabe in eine Datei um:

```
[user@host ~]$ oc adm create-bootstrap-project-template \
>   -o yaml > /tmp/project-template.yaml
```

Passen Sie die Vorlagenressourcen Datei an, um zusätzliche Ressourcen hinzuzufügen, z. B. Kontingente, Einschränzungsbereiche und Netzwerkrichtlinien. Denken Sie daran, dass Kontingente von Cluster-Administratoren konfiguriert werden und nicht von Projektadministratoren hinzugefügt, geändert oder gelöscht werden können. Projektadministratoren können Begrenzungsbereiche und Netzwerkrichtlinien ändern und löschen, selbst wenn diese Ressourcen von der Projektvorlage erstellt wurden.

Neue Projekte erstellen Ressourcen, die im Abschnitt `objects` angegeben sind. Zusätzliche Objekte sollten dieselbe Einrückung wie die `Project-` und `RoleBinding-`Ressourcen verwenden.

Im folgenden Beispiel wird ein Kontingent mit dem Namen des Projekts erstellt. Das Kontingent legt eine Grenze von 3 CPUs, 10 GB Arbeitsspeicher und 10 Pods für das Projekt fest:

```
apiVersion: template.openshift.io/v1
kind: Template
metadata:
  creationTimestamp: null
  name: project-request
objects:
- apiVersion: project.openshift.io/v1
  kind: Project
```

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
...output omitted...
- apiVersion: rbac.authorization.k8s.io/v1
  kind: RoleBinding
...output omitted...
- apiVersion: v1
  kind: ResourceQuota
  metadata:
    name: ${PROJECT_NAME}-quota
  spec:
    hard:
      cpu: "3"
      memory: 10Gi
      pods: "10"
  parameters:
- name: PROJECT_NAME
- name: PROJECT_DISPLAYNAME
- name: PROJECT_DESCRIPTION
- name: PROJECT_ADMIN_USER
- name: PROJECT_REQUESTING_USER
```

Verwenden Sie den Befehl `oc create`, um eine neue Vorlagenressource im Namespace `openshift-config` zu erstellen:

```
[user@host ~]$ oc create -f /tmp/project-template.yaml -n openshift-config
template.template.openshift.io/project-request created
```

Aktualisieren Sie die Ressource `projects.config.openshift.io/cluster` so, dass sie die neue Projektvorlage verwendet. Ändern Sie den Abschnitt `spec`. Standardmäßig lautet der Name der Projektvorlage `project-request`.

```
apiVersion: config.openshift.io/v1
kind: Project
metadata:
...output omitted...
  name: cluster
...output omitted...
spec:
  projectRequestTemplate:
    name: project-request
```

Nach einer erfolgreichen Aktualisierung der Ressource `projects.config.openshift.io/cluster` werden neue `apiserver`-Pods im Namespace `openshift-apiserver` erstellt. Wenn die neuen `apiserver`-Pods ausgeführt wurden, erstellen neue Projekte die Ressourcen aus der angepassten Projektvorlage.

Um zur ursprünglichen Projektvorlage zurückzukehren, ändern Sie die Ressource `projects.config.openshift.io/cluster`. Löschen Sie die `spec`-Ressource so, dass sie mit `spec: {}` übereinstimmt.



Literaturhinweise

Weitere Informationen finden Sie im Kapitel *Quotas* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Applications* unter
https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/applications/index#quotas

Weitere Informationen zu Einschränkungsbereichen finden Sie im Abschnitt *Setting limit ranges* des Kapitels *Working with clusters* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Nodes* unter
https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/nodes/index#nodes-cluster-limit-ranges

Anpassen der OpenShift-Projekterstellung

<https://developers.redhat.com/blog/2020/02/05/customizing-openshift-project-creation/>

► Angeleitete Übung

Beschränken der Ressourcennutzung einer Anwendung

In dieser Übung konfigurieren Sie eine Anwendung so, dass nicht alle Rechenressourcen auf dem Cluster und den zugehörigen Computing-Knoten verwendet werden.

Ergebnisse

Sie sollten in der Lage sein, die OpenShift-Befehlszeilschnittstelle zu verwenden, um:

- Eine Anwendung so zu konfigurieren, dass Ressourcenanforderungen für CPU- und Arbeitsspeichernutzung angegeben werden
- Eine Anwendung für die Arbeit mit vorhandenen Cluster-Beschränkungen zu ändern
- Ein Kontingent zur Beschränkung der Gesamtmenge an CPU, Arbeitsspeicher und Konfigurations-Maps zu erstellen, die einem Projekt zur Verfügung stehen

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt die Ressourcendateien für die Übung.

```
[student@workstation ~]$ lab schedule-limit start
```

Anweisungen

- 1. Erstellen Sie als Benutzer `developer` ein neues Projekt mit dem Namen `schedule-limit`.

- 1.1. Melden Sie sich als Benutzer `developer` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Erstellen Sie für diese angeleitete Übung ein neues Projekt mit dem Namen `schedule-limit`.

```
[student@workstation ~]$ oc new-project schedule-limit
Now using project "schedule-limit" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

- 2. Stellen Sie eine Testanwendung für diese Übung bereit, die explizit Container-Ressourcen für CPU und Arbeitsspeicher anfordert.
- 2.1. Erstellen Sie eine Bereitstellungsressourcendatei, und speichern Sie sie als `~/D0280/labs/schedule-limit/hello-limit.yaml`. Nennen Sie die Anwendung `hello-limit`, und verwenden Sie das Container-Image unter `quay.io/redhattraining/hello-world-nginx:v1.0`.

```
[student@workstation ~]$ oc create deployment hello-limit \
>   --image quay.io/redhattraining/hello-world-nginx:v1.0 \
>   --dry-run=client -o yaml > ~/D0280/labs/schedule-limit/hello-limit.yaml
```

- 2.2. Bearbeiten Sie die Datei `~/D0280/labs/schedule-limit/hello-limit.yaml`, um die Zeile `resources: {}` durch die unten hervorgehobenen Zeilen zu ersetzen. Vergewissern Sie sich vor dem Speichern der Datei, dass Sie die korrekte Einrückung verwendet haben.

```
[student@workstation ~]$ vim ~/D0280/labs/schedule-limit/hello-limit.yaml

...output omitted...

spec:
  containers:
    - image: quay.io/redhattraining/hello-world-nginx:v1.0
      name: hello-world-nginx
    resources:
      requests:
        cpu: "3"
        memory: 20Mi
  status: {}
```

- 2.3. Erstellen Sie die neue Anwendung mit Ihrer Ressourcendatei.

```
[student@workstation ~]$ oc create --save-config \
>   -f ~/D0280/labs/schedule-limit/hello-limit.yaml
deployment.apps/hello-limit created
```

- 2.4. Obwohl eine neue Bereitstellung für die Anwendung erstellt wurde, sollte der Anwendungs-Pod den Status Pending aufweisen.

```
[student@workstation ~]$ oc get pods
NAME                  READY   STATUS    RESTARTS   AGE
hello-limit-d86874d86b-fpmrt   0/1     Pending   0          10s
```

- 2.5. Der Pod kann nicht zugeordnet werden, da keiner der Computing-Knoten über ausreichende CPU-Ressourcen verfügt. Dies können Sie den angezeigten Warnereignissen entnehmen.

```
[student@workstation ~]$ oc get events --field-selector type=Warning
LAST SEEN   TYPE      REASON           OBJECT           MESSAGE
88s         Warning   FailedScheduling   pod/hello-limit-d86874d86b-fpmrt  0/3
nodes are available: 3 Insufficient cpu.
```

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

- 3. Stellen Sie die Anwendung erneut so bereit, dass weniger CPU-Ressourcen angefordert werden.

- 3.1. Bearbeiten Sie `~/D0280/labs/schedule-limit/hello-limit.yaml` so, dass 1,2 CPUs für den Container angefordert werden. Ändern Sie die Zeile `cpu: "3"` so, dass sie der unten hervorgehobenen Zeile entspricht.

```
[student@workstation ~]$ vim ~/D0280/labs/schedule-limit/hello-limit.yaml
...output omitted...
resources:
  requests:
    cpu: "1200m"
    memory: 20Mi
```

- 3.2. Wenden Sie die Änderungen auf Ihre Anwendung an.

```
[student@workstation ~]$ oc apply -f ~/D0280/labs/schedule-limit/hello-limit.yaml
deployment.apps/hello-limit configured
```

- 3.3. Vergewissern Sie sich, dass die Anwendung erfolgreich bereitgestellt wird. Möglicherweise müssen Sie `oc get pods` mehrmals ausführen, bis ein ausgeführter Pod angezeigt wird. Der vorherige Pod mit dem Status „Pending“ wird beendet und schließlich ausgeblendet.

```
[student@workstation ~]$ oc get pods
NAME           READY   STATUS      RESTARTS   AGE
hello-limit-d86874d86b-fpmrt   0/1     Terminating   0          2m19s
hello-limit-7c7998ff6b-ctsjp   1/1     Running      0          6s
```

**Anmerkung**

Wenn Ihr Anwendungs-Pod nicht geplant wird, ändern Sie die Datei `~/D0280/labs/schedule-limit/hello-limit.yaml` so, dass die CPU-Anforderung auf `1100m` reduziert wird. Wenden Sie die Änderungen erneut an und stellen Sie sicher, dass der Pod-Status `Running` lautet.

- 4. Versuchen Sie, die Anwendung von einem Pod auf vier Pods zu skalieren. Nachdem Sie verifiziert haben, dass diese Änderung die Kapazität Ihres Clusters überschreiten würde, löschen Sie die Ressourcen, die der Anwendung `hello-limit` zugeordnet sind.

- 4.1. Skalieren Sie die Anwendung `hello-limit` manuell auf vier Pods hoch.

```
[student@workstation ~]$ oc scale --replicas 4 deployment/hello-limit
deployment.apps/hello-limit scaled
```

- 4.2. Überprüfen Sie, ob alle vier Pods ausgeführt werden. Möglicherweise müssen Sie `oc get pods` mehrmals ausführen, bis angezeigt wird, dass sich ein Pod im Status „Pending“ befindet. Je nach aktueller Cluster-Last können sich mehrere Pods im Status „Pending“ befinden.



Anmerkung

Wenn Ihr Anwendungs-Pod immer noch nicht bereitgestellt wird, skalieren Sie die Anzahl der Anwendungs-Pods auf 0 und reduzieren Sie die CPU-Anforderung in `~/D0280/labs/schedule-limit/hello-limit.yaml` auf `1000m`. Führen Sie `oc apply -f ~/D0280/labs/schedule-limit/hello-limit.yaml` aus, um die Änderungen anzuwenden, und führen Sie dann den Befehl `oc scale` aus, um eine Skalierung auf vier Pods auszuführen.

```
[student@workstation ~]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
hello-limit-d55cd65c-765s9  1/1     Running   0          12s
hello-limit-d55cd65c-hmlbw  0/1     Pending   0          12s
hello-limit-d55cd65c-r8lvw  1/1     Running   0          12s
hello-limit-d55cd65c-vkrhk  0/1     Pending   0          12s
```

- 4.3. Warnereignisse deuten darauf hin, dass einer oder mehrere Pods nicht zugeordnet werden können, da keiner der Knoten über ausreichende CPU-Ressourcen verfügt. Ihre Warnmeldungen weichen möglicherweise etwas ab.

```
[student@workstation ~]$ oc get events --field-selector type=Warning
LAST SEEN      TYPE        REASON          OBJECT
MESSAGE
...output omitted...
76s           Warning     FailedScheduling   pod/hello-limit-d55cd65c-vkrhk      0/3
nodes are available: 3 Insufficient cpu.
```

- 4.4. Löschen Sie alle Ressourcen, die der Anwendung `hello-limit` zugeordnet sind.

```
[student@workstation ~]$ oc delete all -l app=hello-limit
pod "hello-limit-d55cd65c-765s9" deleted
pod "hello-limit-d55cd65c-hmlbw" deleted
pod "hello-limit-d55cd65c-r8lvw" deleted
pod "hello-limit-d55cd65c-vkrhk" deleted
deployment.apps "hello-limit" deleted
replicaset.apps "hello-limit-5cc86ff6b8" deleted
replicaset.apps "hello-limit-7d6bdcc99b" deleted
```

- 5. Stellen Sie eine zweite Anwendung bereit, um die Arbeitsspeichernutzung zu testen. Diese zweite Anwendung legt eine Arbeitsspeicherbeschränkung von 200 MB pro Container fest.
- 5.1. Verwenden Sie die Ressourcendatei unter `/home/student/D0280/labs/schedule-limit/loadtest.yaml`, um die Anwendung `loadtest` zu erstellen. Zusätzlich zu einer Bereitstellung erstellt diese Ressourcendatei auch einen Service und eine Route.

```
[student@workstation ~]$ oc create --save-config \
>   -f ~/DO280/labs/schedule-limit/loadtest.yaml
deployment.apps/loadtest created
service/loadtest created
route.route.openshift.io/loadtest created
```

- 5.2. Das Container-Image **loadtest** ist so konzipiert, dass die CPU- oder Arbeitsspeichernutzung des Containers erhöht wird, indem eine Anforderung an die Anwendungs-API gestellt wird. Ermitteln Sie den vollständig qualifizierten Domain Name, der für die Route verwendet wird.

```
[student@workstation ~]$ oc get routes
NAME      HOST/PORT      ...
loadtest  loadtest.apps.ocp4.example.com ...
```

► 6. Generieren Sie zusätzliche Arbeitsspeicherlast, die vom Container verarbeitet werden kann.

- 6.1. Öffnen Sie zwei zusätzliche Terminalfenster, um die Auslastung Ihrer Anwendung kontinuierlich zu überwachen. Greifen Sie im ersten Terminal auf die Anwendungs-API zu, um zusätzliche Last für den Arbeitsspeicher auf dem Container zu simulieren.
 Führen Sie im zweiten Terminalfenster `watch oc get pods` aus, um kontinuierlich den Status jedes Pods zu überwachen.
 Führen Sie im dritten Terminalfenster `watch oc adm top pod` aus, um kontinuierlich die CPU- und Arbeitsspeichernutzung jedes Pods zu überwachen.
- 6.2. Verwenden Sie im ersten Terminalfenster die Anwendungs-API, um die Arbeitsspeicherlast für 60 Sekunden um 150 MB zu erhöhen. Verwenden Sie den vollständig qualifizierten Domain Name, den Sie zuvor für die Route ermittelt haben. Beobachten Sie die Ausgabe in den anderen beiden Terminalfenstern, während Sie auf den Abschluss des Befehls `curl` warten.

```
[student@workstation ~]$ curl -X GET \
>   http://loadtest.apps.ocp4.example.com/api/loadtest/v1/mem/150/60
curl: (52) Empty reply from server
```

- 6.3. Beobachten Sie im zweiten Terminalfenster die Ausgabe von `watch oc get pods`. Da der Container die zusätzliche Last verarbeiten kann, sollten Sie feststellen, dass der einzelne Anwendungs-Pod den Status **Running** für die gesamte `curl`-Anforderung aufweist.

Every 2.0s: <code>oc get pods</code>					
NAME	READY	STATUS	RESTARTS	AGE	...
loadtest-f7495948-tlxgm	1/1	Running	0	7m34s	

- 6.4. Beobachten Sie im dritten Terminalfenster die Ausgabe von `watch oc adm top pod`. Zu Beginn beträgt die Arbeitsspeichernutzung für den Pod ca. 20–30 Mi.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
Every 2.0s: oc adm top pod ...
```

NAME	CPU(cores)	MEMORY(bytes)
loadtest-f7495948-tlxgm	0m	20Mi

Bei der API-Anforderung sollte die Arbeitsspeichernutzung für den Pod auf ca. 170–180 Mi erhöht werden.

```
Every 2.0s: oc adm top pod ...
```

NAME	CPU(cores)	MEMORY(bytes)
loadtest-f7495948-tlxgm	0m	172Mi

Kurz nach Abschluss der curl-Anforderung sollte die Arbeitsspeichernutzung auf etwa 20–30 Mi zurückfallen.

```
Every 2.0s: oc adm top pod ...
```

NAME	CPU(cores)	MEMORY(bytes)
loadtest-f7495948-tlxgm	0m	20Mi

- 7. Generieren Sie zusätzliche Arbeitsspeicherlast, die nicht vom Container verarbeitet werden kann.

- 7.1. Verwenden Sie die Anwendungs-API, um die Arbeitsspeicherlast für 60 Sekunden um 200 MB zu erhöhen. Beobachten Sie die Ausgabe in den beiden anderen Terminalfenstern.

```
[student@workstation ~]$ curl -X GET \
>   http://loadtest.apps.ocp4.example.com/api/loadtest/v1/mem/200/60
<html><body><h1>502 Bad Gateway</h1>
The server returned an invalid or incomplete response.
</body></html>
```

- 7.2. Beobachten Sie im zweiten Terminalfenster die Ausgabe von `watch oc get pods`. Fast unmittelbar nach dem Ausführen des Befehls `curl` wechselt der Status des Pods zu `OOMKilled`. Es könnte sogar der Status `Error` angezeigt werden. Der Pod verfügt über keinen Arbeitsspeicher mehr und muss abgebrochen und neu gestartet werden. Der Status kann in `CrashLoopBackOff` geändert werden, bevor er zum Status `Running` zurückkehrt. Die Anzahl der Neustarts wird ebenfalls erhöht.

```
Every 2.0s: oc get pods ...
```

NAME	READY	STATUS	RESTARTS	AGE
loadtest-f7495948-tlxgm	0/1	0OMKilled	0	9m13s

In einigen Fällen könnte der Pod neu gestartet und in den Status `Running` zurückgekehrt sein, bevor Sie zum zweiten Terminalfenster wechseln konnten. Die Anzahl der Neustarts wird von 0 auf 1 erhöht.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
Every 2.0s: oc get pods           ...

NAME          READY   STATUS    RESTARTS   AGE
loadtest-f7495948-tlxgm   1/1     Running   1          9m33s
```

- 7.3. Beobachten Sie im dritten Terminalfenster die Ausgabe von `watch oc adm top pod`. Nachdem der Pod beendet wurde, zeigen Metriken für eine kurze Zeit an, dass der Pod wenige bis gar keine Ressourcen verwendet.

```
Every 2.0s: oc adm top pod           ...

NAME          CPU(cores)   MEMORY(bytes)
loadtest-f7495948-tlxgm   8m          0Mi
```

- 7.4. Löschen Sie im ersten Terminalfenster alle Ressourcen, die der zweiten Anwendung zugeordnet sind.

```
[student@workstation ~]$ oc delete all -l app=loadtest
pod "loadtest-f7495948-tlxgm" deleted
service "loadtest" deleted
deployment.apps "loadtest" deleted
route.route.openshift.io "loadtest" deleted
```

Drücken Sie im zweiten und dritten Terminalfenster Strg+C, um den `watch`-Befehl zu beenden. Schließen Sie optional das zweite und dritte Terminalfenster.

- 8. Erstellen Sie als Cluster-Administrator Kontingente für das Projekt `schedule-limit`.

- 8.1. Melden Sie sich als Benutzer `admin` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
```

- 8.2. Erstellen Sie ein Kontingent mit dem Namen `project-quota`, das das Projekt `schedule-limit` auf 3 CPUs, 1 GB Arbeitsspeicher und 3 Konfigurations-Maps beschränkt.

```
[student@workstation ~]$ oc create quota project-quota \
>   --hard cpu="3",memory="1G",configmaps="3" \
>   -n schedule-limit
resourcequota/project-quota created
```

**Anmerkung**

In dieser Übung wird ein Kontingent für Konfigurations-Maps festgelegt, um zu demonstrieren, was geschieht, wenn ein Benutzer versucht, das Kontingent zu überschreiten.

- 9. Versuchen Sie als Entwickler, das Konfigurations-Maps-Kontingent für das Projekt zu überschreiten.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

- 9.1. Melden Sie sich als Benutzer developer bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer
Login successful.
...output omitted...
```

- 9.2. Verwenden Sie eine Schleife (Loop), um zu versuchen, vier Konfigurations-Maps zu erstellen. Die ersten drei sollten erfolgreich sein, und die vierte sollte fehlschlagen, da sie das Kontingent überschreitet.

```
[student@workstation ~]$ for X in {1..4}
>   do
>     oc create configmap my-config${X} --from-literal key${X}=value${X}
>   done
configmap/my-config1 created
configmap/my-config2 created
configmap/my-config3 created
Error from server (Forbidden): configmaps "my-config4" is forbidden: exceeded
  quota: project-quota, requested: configmaps=1, used: configmaps=3, limited:
  configmaps=3
```

- 10. Konfigurieren Sie als Cluster-Administrator alle neuen Projekte mit Standardressourcen für Kontingente und Einschränkungsbereiche.

- 10.1. Melden Sie sich als Benutzer admin bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
```

- 10.2. Leiten Sie die Ausgabe des Befehls `oc adm create-bootstrap-project-template` um, sodass Sie die Projekterstellung anpassen können.

```
[student@workstation ~]$ oc adm create-bootstrap-project-template \
>   -o yaml > /tmp/project-template.yaml
```

- 10.3. Bearbeiten Sie die Datei `/tmp/project-template.yaml`, und fügen Sie die in der Datei `/home/student/D0280/labs/schedule-limit/quota-limits.yaml` definierten Ressourcen für Kontingente und Einschränkungsbereiche hinzu. Fügen Sie die Zeilen vor dem Abschnitt `parameters` hinzu.

```
...output omitted...
- apiVersion: v1
  kind: ResourceQuota
  metadata:
    name: ${PROJECT_NAME}-quota
  spec:
    hard:
      cpu: 3
      memory: 10G
- apiVersion: v1
  kind: LimitRange
```

```

metadata:
  name: ${PROJECT_NAME}-limits
spec:
  limits:
    - type: Container
      defaultRequest:
        cpu: 30m
        memory: 30M
parameters:
- name: PROJECT_NAME
- name: PROJECT_DISPLAYNAME
- name: PROJECT_DESCRIPTION
- name: PROJECT_ADMIN_USER
- name: PROJECT_REQUESTING_USER

```



Anmerkung

Die Datei /home/student/D0280/solutions/schedule-limit/project-template.yaml enthält die richtige Konfiguration und kann als Vergleich verwendet werden.

- 10.4. Verwenden Sie die Datei /tmp/project-template.yaml, um eine neue Vorlagenressource im Namespace openshift-config zu erstellen.

```
[student@workstation ~]$ oc create -f /tmp/project-template.yaml \
>   -n openshift-config
template.template.openshift.io/project-request created
```

- 10.5. Aktualisieren Sie den Cluster, um die benutzerdefinierte Projektvorlage zu verwenden.

```
[student@workstation ~]$ oc edit projects.config.openshift.io/cluster
```

Ändern Sie den Abschnitt spec und fügen Sie die folgenden, fett gedruckten Zeilen ein.

```

...output omitted...
spec:
  projectRequestTemplate:
    name: project-request
```

Überprüfen Sie die Einrückung und speichern Sie Ihre Änderungen.

```
project.config.openshift.io/cluster edited
```

- 10.6. Wenn die Änderung erfolgreich war, werden die apiserver-Pods im openshift-apiserver-Namespace neu erstellt.

```
[student@workstation ~]$ watch oc get pods -n openshift-apiserver
```

Warten Sie, bis alle drei neuen apiserver-Pods bereit sind und ausgeführt werden.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
Every 2.0s: oc get pods -n openshift-apiserver
NAME          READY   STATUS    RESTARTS   AGE
apiserver-868dccf5fb-885dz  2/2     Running   0          63s
apiserver-868dccf5fb-8j4vh  2/2     Running   0          39s
apiserver-868dccf5fb-r4j9b  2/2     Running   0          24s
```

Drücken Sie Strg+C, um den watch-Befehl zu beenden.

- 10.7. Erstellen Sie ein Testprojekt, um zu überprüfen, ob die benutzerdefinierte Projektvorlage erwartungsgemäß funktioniert.

```
[student@workstation ~]$ oc new-project template-test
Now using project "template-test" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

- 10.8. Listen Sie das Ressourcenkontingent und die Einschränkungsbereichsressourcen im Testprojekt auf.

```
[student@workstation ~]$ oc get resourcequotas,limitranges
NAME          AGE   REQUEST           LIMIT
resourcequota/template-test-quota  87s   cpu: 0/3, memory: 0/10G

NAME          CREATED AT
limitrange/template-test-limits   2021-06-02T15:46:37Z
```

- 11. Bereinigen Sie die Lab-Umgebung, indem Sie das Projekt aus dieser Übung löschen.

- 11.1. Löschen Sie das Projekt `schedule-limit`.

```
[student@workstation ~]$ oc delete project schedule-limit
project.project.openshift.io "schedule-limit" deleted
```

- 11.2. Löschen Sie das Projekt `template-test`.

```
[student@workstation ~]$ oc delete project template-test
project.project.openshift.io "template-test" deleted
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab schedule-limit finish
```

Hiermit ist die angeleitete Übung beendet.

Skalieren einer Anwendung

Ziele

Am Ende dieses Abschnitts sollten Sie zu Folgendem in der Lage sein: Steuern der Anzahl von Replikaten eines Pods, Angeben der Anzahl von Replikaten in einer Bereitstellung, manuelles Skalieren der Anzahl der Replikate und Erstellen einer Horizontal Pod Autoscaler-Ressource (HPA).

Angeben von Pod-Replikaten in Konfigurations-Workloads

Die Anzahl der Pod-Replikate für eine bestimmte Bereitstellung kann je nach Bedarf erhöht oder verringert werden. Trotz der Ressourcen `ReplicaSet` und `ReplicationController` ist die Anzahl der für eine Anwendung erforderlichen Replikate in der Regel in einer Ressource für die Bereitstellung definiert. Ein Replikatsatz oder ein Replikationscontroller (verwaltet durch eine Bereitstellung) garantiert, dass die angegebene Anzahl von Replikaten eines Pods jederzeit ausgeführt wird. Der Replikatsatz oder der Replikationscontroller kann Pods hinzufügen oder entfernen, um der gewünschten Replikatanzahl zu entsprechen.

Bereitstellungsressourcen enthalten:

- Die gewünschten Anzahl von Replikaten
- Einen Selektor zur Identifizierung der verwalteten Pods
- Eine Pod-Definition oder -Vorlage zum Erstellen eines replizierten Pods (einschließlich der für den Pod zu verwendenden Labels)

Die folgende Bereitstellungsressource (erstellt mit dem Befehl `oc create deployment`) zeigt diese Elemente an:

```
apiVersion: apps/v1
kind: Deployment
...output omitted...
spec:
  replicas: 1 ①
  selector:
    matchLabels:
      deployment: scale ②
  strategy: {}
  template: ③
    metadata:
      labels:
        deployment: scale ④
    spec:
      containers:
        ...output omitted...
```

① Gibt die gewünschte Anzahl der auszuführenden Kopien (oder Replikate) des Pods an.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

- ② Ein Replikatsatz verwendet einen Selektor, um die Anzahl der laufenden Pods zu bestimmen, so wie ein Service einen Selektor verwendet, um Pods für das Load Balancing zu finden.
- ③ Eine Vorlage für Pods, die der Replikatsatz oder Replikationscontroller erstellt.
- ④ Labels für Pods, die von der Vorlage erstellt werden, müssen mit denen für den Selektor übereinstimmen.

Wenn die Ressource für die Bereitstellung unter Versionskontrolle steht, dann ändern Sie in der Ressourcendatei die Zeile `replicas` und wenden die Änderungen mit dem Befehl `oc apply` an.

In einer Ressource für die Bereitstellung ist ein Selektor ein Label-Satz, mit dem alle vom Replikatsatz verwalteten Pods übereinstimmen müssen. Derselbe Label-Satz muss in der Pod-Definition enthalten sein, die die Bereitstellung instanziert. Dieser Selektor wird verwendet, um zu bestimmen, wie viele Instanzen des Pods bereits ausgeführt werden, um die gewünschte Anzahl einzuhalten.

**Anmerkung**

Der Replikationscontroller führt keine Autoskalierung aus, da er die Last oder den Datenverkehr nicht überwacht. Die Ressource horizontaler Pod-Autoscaler, die später in diesem Kapitel vorgestellt wird, verwaltet das automatische Skalieren.

Manuelles Skalieren der Anzahl der Pod-Replikate

Entwickler und Administratoren können die Anzahl der Pod-Replikate in einem Projekt manuell skalieren. Für eine erwartete Zunahme des Datenverkehrs sind möglicherweise mehr Pods erforderlich, oder die Pod-Anzahl kann reduziert werden, um Ressourcen zurückzufordern, die an anderer Stelle im Cluster verwendet werden können. Unabhängig davon, ob die Anzahl der Pod-Replikate erhöht oder verringert wird, besteht der erste Schritt darin, die entsprechende Bereitstellung zu ermitteln, die mit dem Befehl `oc get` skaliert werden soll:

```
[user@host ~]$ oc get deployment
NAME      READY     UP-TO-DATE   AVAILABLE   AGE
scale     1/1       1           1           8h
```

Die Anzahl von Replikaten in einer Ressource für die Bereitstellung kann mit dem Befehl `oc scale` manuell geändert werden:

```
[user@host ~]$ oc scale --replicas 5 deployment/scale
```

Die Bereitstellungsressource leitet die Änderung an den Replikatsatz weiter. Dieses reagiert auf die Änderung, indem es neue Pods (Replikate) erstellt oder vorhandene löscht, je nachdem, ob die neue gewünschte Anzahl an Replikaten kleiner oder größer als die vorhandene Anzahl ist.

Auch wenn eine Ressource für den Replikatsatz direkt bearbeitet werden kann, wird allgemein empfohlen, stattdessen die Ressource für die Bereitstellung zu bearbeiten. Bei einer neuen Bereitstellung wird entweder ein neuer Replikatsatz oder Replikationscontroller erstellt, und Änderungen, die direkt an einem früheren ReplicaSet oder Replikationscontroller vorgenommen wurden, werden ignoriert.

Automatisches Skalieren von Pods

OpenShift kann mithilfe eines `HorizontalPodAutoscaler`-Ressourcentyps eine Bereitstellung, entsprechend der aktuellen Belastung der Anwendungs-Pods, automatisch skalieren.

Eine „Horizontaler Pod-Autoscaler“-Ressource verwendet vom OpenShift Metrics-Subsystem erfasste Leistungsmetriken. Das Metrik-Subsystem ist in OpenShift 4 vorinstalliert und muss nicht wie in OpenShift 3 separat installiert werden. Um eine Bereitstellung automatisch zu skalieren, müssen Sie Ressourcenanforderungen für Pods angeben, damit der Horizontal Pod Autoscaler den prozentualen Anteil der Nutzung berechnen kann.

Es wird empfohlen, für die Erstellung einer „Horizontaler Pod-Autoscaler“-Ressource den Befehl `oc autoscale` zu verwenden, zum Beispiel:

```
[user@host ~]$ oc autoscale deployment/hello \
>   --min 1 --max 10 --cpu-percent 80
```

Der vorherige Befehl erstellt eine Horizontal Pod Autoscaler-Ressource, die die Anzahl von Replikaten in der Bereitstellung `hello` ändert, um ihre Pods unter 80 % ihrer angeforderten CPU-Gesamtlast zu halten.

Der Befehl `oc autoscale` erstellt anhand des Namens der Bereitstellung als Argument (`hello` im vorherigen Beispiel) eine Horizontal Pod Autoscaler-Ressource.



Anmerkung

Die automatische Skalierung für die Arbeitsspeichernutzung ist nach wie vor ein Technologievorschau-Feature für Red Hat OpenShift Container Platform 4.6.

Die Maximal- und Minimalwerte für die Ressource „Horizontaler Pod-Autoscaler“ dienen dazu, Belastungsspitzen auszugleichen und eine Überlastung des OpenShift-Clusters zu vermeiden. Wenn sich die Last der Anwendung zu schnell ändert, dann ist es möglicherweise ratsam, eine gewisse Anzahl zusätzlicher Pods vorzuhalten, um eine plötzlich erhöhte Anzahl von Benutzeranforderungen zu bewältigen. Umgekehrt können zu viele Pods die gesamte Cluster-Kapazität aufbrauchen und andere Anwendungen beeinträchtigen, die denselben OpenShift-Cluster teilen.

Information über „Horizontaler Pod-Autoscaler“-Ressourcen im aktuellen Projekt erhalten Sie mit dem Befehl `oc get`. Beispiel:

```
[user@host ~]$ oc get hpa
NAME      REFERENCE          TARGETS          MINPODS  MAXPODS  REPLICAS  ...
hello    Deployment/hello    <unknown>/80%       1         10        1         ...
scale    Deployment/scale    60%/80%          2         10        2         ...
```



Wichtig

Der horizontale Pod-Autoscaler hat in der Spalte „TARGETS“ zunächst den Wert „<unknown>“. Es kann bis zu fünf Minuten dauern, bevor „<unknown>“ den Prozentsatz der aktuellen Nutzung anzeigt.

Der persistente Wert <unknown> in der Spalte „TARGETS“ gibt an, dass die Bereitstellung keine Ressourcenanforderungen für die Metrik definiert. Der horizontale Pod-Autoscaler skaliert diese Pods nicht.

Die meisten Pods, die mit dem Befehl `oc new-app` erstellt wurden, definieren keine Ressourcenanforderungen. Für die Verwendung des OpenShift-Autoscalers ist es deshalb möglicherweise erforderlich, dass Sie die Ressourcen für die Bereitstellung bearbeiten, benutzerdefinierte YAML- oder JSON-Ressourcendateien für Ihre Anwendung erstellen oder Ihrem Projekt Ressourcen für Beschränkungsbereiche hinzufügen, die Standardressourcenanforderungen definieren.



Literaturhinweise

Weitere Informationen finden Sie im Abschnitt *Automatically scaling pods with the Horizontal Pod Autoscaler* des Kapitels *Working with pods* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 Nodes unter
https://access.redhat.com/documentation/en-us/redshift_container_platform/4.6/html-single/nodes/index#nodes-pods-autoscaling

► Angeleitete Übung

Skalieren einer Anwendung

In dieser Übung skalieren Sie eine Anwendung manuell und automatisch.

Ergebnisse

Sie sollten in der Lage sein, die OpenShift-Befehlszeilenschnittstelle zu verwenden, um:

- Eine Anwendung manuell zu skalieren
- Eine Anwendung so zu konfigurieren, dass sie je nach Verwendung automatisch skaliert wird

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt die Ressourcendateien für die Übung.

```
[student@workstation ~]$ lab schedule-scale start
```

Anweisungen

- 1. Erstellen Sie als Benutzer `developer` ein neues Projekt mit dem Namen `schedule-scale`.

- 1.1. Melden Sie sich als Benutzer `developer` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer \  
> https://api.ocp4.example.com:6443  
Login successful.  
...output omitted...
```

- 1.2. Erstellen Sie für diese angeleitete Übung ein neues Projekt mit dem Namen `schedule-scale`.

```
[student@workstation ~]$ oc new-project schedule-scale  
Now using project "schedule-scale" on server "https://api.ocp4.example.com:6443".  
...output omitted...
```

- 2. Stellen Sie eine Testanwendung für diese Übung bereit, die explizit Container-Ressourcen für CPU und Arbeitsspeicher anfordert.

- 2.1. Ändern Sie die Ressourcendatei unter `~/D0280/labs/schedule-scale/loadtest.yaml`, um sowohl Anforderungen als auch Beschränkungen für die CPU- und Arbeitsspeichernutzung festzulegen.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
[student@workstation ~]$ vim ~/D0280/labs/schedule-scale/loadtest.yaml
```

- 2.2. Ersetzen Sie die Zeile `resources: {}` durch die unten hervorgehobenen Zeilen. Vergewissern Sie sich vor dem Speichern der Datei, dass Sie die korrekte Einrückung verwendet haben.

```
...output omitted...
spec:
  containers:
    - image: quay.io/redhattraining/loadtest:v1.0
      name: loadtest
      resources:
        requests:
          cpu: "25m"
          memory: 25Mi
        limits:
          cpu: "100m"
          memory: 100Mi
  status: {}
```

- 2.3. Erstellen Sie die neue Anwendung mit Ihrer Ressourcendatei.

```
[student@workstation ~]$ oc create --save-config \
>   -f ~/D0280/labs/schedule-scale/loadtest.yaml
deployment.apps/loadtest created
service/loadtest created
route.route.openshift.io/loadtest created
```

- 2.4. Verifizieren Sie, dass Ihre Anwendung den Status `Running` aufweist. Möglicherweise müssen Sie den Befehl `oc get pods` mehrmals ausführen.

```
[student@workstation ~]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
loadtest-5f9565dbfb-jm9md   1/1     Running   0          23s
```

- 2.5. Stellen Sie sicher, dass der Anwendungs-Pod Ressourcenbeschränkungen und -anforderungen angibt.

```
[student@workstation ~]$ oc describe pod/loadtest-5f9565dbfb-jm9md \
>   | grep -A2 -E "Limits|Requests"
Limits:
  cpu:      100m
  memory:  100Mi
Requests:
  cpu:      25m
  memory:  25Mi
```

- 3. Skalieren Sie die Bereitstellung `loadtest` manuell, indem Sie zuerst die Anzahl der ausgeführten Pods erhöhen und dann verringern.

- 3.1. Skalieren Sie die Bereitstellung `loadtest` auf fünf Pods hoch.

```
[student@workstation ~]$ oc scale --replicas 5 deployment/loadtest
deployment.apps/loadtest scaled
```

- 3.2. Überprüfen Sie, ob alle fünf Anwendungs-Pods ausgeführt werden. Möglicherweise müssen Sie den Befehl `oc get pods` mehrmals ausführen.

```
[student@workstation ~]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
loadtest-5f9565dbfb-22f9s  1/1     Running   0          54s
loadtest-5f9565dbfb-8l2rn  1/1     Running   0          54s
loadtest-5f9565dbfb-jm9md  1/1     Running   0          3m17s
loadtest-5f9565dbfb-lfhns  1/1     Running   0          54s
loadtest-5f9565dbfb-prjkl  1/1     Running   0          54s
```

- 3.3. Skalieren Sie die Bereitstellung `loadtest` zurück auf einen Pod.

```
[student@workstation ~]$ oc scale --replicas 1 deployment/loadtest
deployment.apps/loadtest scaled
```

- 3.4. Überprüfen Sie, ob nur ein Anwendungs-Pod ausgeführt wird. Möglicherweise müssen Sie den Befehl `oc get pods` mehrmals ausführen, während Sie darauf warten, dass die anderen Pods beendet werden.

```
[student@workstation ~]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
loadtest-5f9565dbfb-prjkl  1/1     Running   0          72s
```

- 4. Konfigurieren Sie die Anwendung `loadtest` so, dass sie auf Grundlage der Last automatisch skaliert wird, und testen Sie dann die Anwendung, indem Sie Last hinzufügen.

- 4.1. Erstellen Sie einen horizontalen Pod-Autoscaler, der sicherstellt, dass in der Anwendung `loadtest` immer 2 Anwendungs-Pods ausgeführt werden. Diese Anzahl kann auf maximal 10 Pods erhöht werden, wenn die CPU-Last 50 % übersteigt.

```
[student@workstation ~]$ oc autoscale deployment/loadtest \
>   --min 2 --max 10 --cpu-percent 50
horizontalpodautoscaler.autoscaling/loadtest autoscaled
```

- 4.2. Warten Sie bis der horizontale Pod-Autoscaler `loadtest` die Nutzung in der Spalte TARGETS angibt.

```
[student@workstation ~]$ watch oc get hpa/loadtest
```

Drücken Sie Strg+C, um den Befehl `watch` zu beenden, sobald für <unknown> ein Prozentwert angezeigt wird.

```
Every 2.0s: oc get hpa/loadtest
...
NAME      REFERENCE        TARGETS      MINPODS   MAXPODS   REPLICAS   ...
loadtest  Deployment/loadtest  0%/50%       2          10         2          ...
```



Anmerkung

Es kann bis zu fünf Minuten dauern, bis der Eintrag <unknown> in der Spalte TARGETS in 0% geändert wird. Wenn der Eintrag <unknown> nicht geändert wird, verwenden Sie den Befehl `oc describe`, um zu überprüfen, ob die Container für die `loadtest`-Anwendung CPU-Ressourcen anfordern.

- 4.3. Das Container-Image `loadtest` ist so konzipiert, dass die CPU- oder Arbeitsspeichernutzung des Containers erhöht wird, indem eine Anforderung an die Anwendungs-API gestellt wird. Ermitteln Sie den vollständig qualifizierten Domain Name, der für die Route verwendet wird.

```
[student@workstation ~]$ oc get route/loadtest
NAME      HOST/PORT
loadtest  loadtest-schedule-scale.apps.ocp4.example.com ...
```

- 4.4. Greifen Sie auf die Anwendungs-API zu, um zusätzliche Last für die CPU auf dem Container zu simulieren. Fahren Sie mit dem nächsten Schritt fort, während Sie darauf warten, dass der Befehl `curl` abgeschlossen wird.

```
[student@workstation ~]$ curl -X GET \
>   http://loadtest-schedule-scale.apps.ocp4.example.com/api/loadtest/v1/cpu/1
curl: (52) Empty reply from server
```

- 4.5. Öffnen Sie ein zweites Terminalfenster, und überwachen Sie kontinuierlich den Status des horizontalen Pod-Autoscalers.



Anmerkung

Die erhöhte Aktivität der Anwendung löst den Autoscaler nicht sofort aus. Warten Sie einige Augenblicke, wenn keine Änderungen der Replikatanzahl angezeigt werden.

```
[student@workstation ~]$ watch oc get hpa/loadtest
```

Wenn die Last zunimmt (sichtbar in der Spalte TARGETS), sollte die Anzahl unter REPLICAS erhöht werden. Beobachten Sie die Ausgabe eine oder zwei Minuten, bevor Sie mit dem nächsten Schritt fortfahren. Ihre Ausgabe weicht wahrscheinlich von der unten gezeigten ab.

```
Every 2.0s: oc get hpa/loadtest ...
NAME      REFERENCE          TARGETS      MINPODS   MAXPODS   REPLICAS   ...
loadtest  Deployment/loadtest  172%/50%    2          10        9          ...
```



Anmerkung

Die „Horizontale Pod-Autoscaler“-Ressource kann schnell eine horizontale Erweiterung durchführen, eine horizontale Verkleinerung ist jedoch langsamer. Anstatt darauf zu warten, dass die Anwendung `loadtest` auf zwei Pods herunterskaliert wird, fahren Sie mit dem Rest der Übung fort.

- 5. Erstellen Sie im ersten Terminalfenster eine zweite Anwendung mit dem Namen `scaling`. Skalieren Sie die Anwendung, und überprüfen Sie dann die Antworten von den Anwendungs-Pods.
- 5.1. Erstellen Sie eine neue Anwendung mit dem Befehl `oc new-app` und dem Container-Image unter `quay.io/redhattraining/scaling:v1.0`.

```
[student@workstation ~]$ oc new-app --name scaling \
>   --docker-image quay.io/redhattraining/scaling:v1.0
...output omitted...
--> Creating resources ...
  imagestream.image.openshift.io "scaling" created
  deployment.apps "scaling" created
  service "scaling" created
--> Success
  Application is not exposed. You can expose services to the outside world by
executing one or more of the commands below:
  'oc expose svc/scaling'
Run 'oc status' to view your app.
```

- 5.2. Erstellen Sie eine Route zu der Anwendung, indem Sie den Service für die Anwendung bereitstellen.

```
[student@workstation ~]$ oc expose svc/scaling
route.route.openshift.io/scaling exposed
```

- 5.3. Skalieren Sie die Anwendung mit der Bereitstellungsressource für die Anwendung auf drei Pods hoch.

```
[student@workstation ~]$ oc scale --replicas 3 deployment/scaling
deployment.apps/scaling scaled
```

- 5.4. Überprüfen Sie, ob alle drei Pods für die Anwendung `scaling` ausgeführt werden, und ermitteln Sie die zugehörigen IP-Adressen.

```
[student@workstation ~]$ oc get pods -o wide -l deployment=scaling
NAME        READY   STATUS    RESTARTS   AGE      IP          NODE
scaling-1-bm4m2  1/1    Running   0          45s     10.10.0.29  master01 ...
scaling-1-w7whl  1/1    Running   0          45s     10.8.0.45   master03 ...
scaling-1-xqvs2  1/1    Running   0          6m1s   10.9.0.58   master02 ...
```

- 5.5. Zeigen Sie den Hostnamen an, der zum Weiterleiten von Anforderungen an die Anwendung `scaling` verwendet wird.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
[student@workstation ~]$ oc get route/scaling
NAME      HOST/PORT
scaling   scaling-schedule-scale.apps.ocp4.example.com ...
```

- 5.6. Wenn Sie auf den Hostnamen für Ihre Anwendung zugreifen, wird auf der PHP-Seite die IP-Adresse des Pods ausgegeben, der auf die Anforderung geantwortet hat. Senden Sie mehrere Anforderungen an Ihre Anwendung, und sortieren Sie dann die Antworten, um die Anzahl der an die einzelnen Pods gesendeten Anforderungen zu zählen. Führen Sie das Skript unter ~/D0280/labs/schedule-scale/curl-route.sh aus.

```
[student@workstation ~]$ ~/D0280/labs/schedule-scale/curl-route.sh
34 Server IP: 10.10.0.29
34 Server IP: 10.8.0.45
32 Server IP: 10.9.0.58
```

- ▶ 6. Optional: Überprüfen Sie den Status des horizontalen Pod-Autoscalers, der für die Anwendung loadtest ausgeführt wird. Wenn der Befehl `watch oc get hpa/loadtest` weiterhin im zweiten Terminalfenster ausgeführt wird, wechseln Sie zu diesem Terminalfenster, und beobachten Sie die Ausgabe. Wenn genügend Zeit vergangen ist, sollte die Replikatanzahl wieder auf zwei zurückgesetzt werden. Drücken Sie anschließend **Strg+C**, um den `watch`-Befehl zu beenden, und schließen Sie dann das zweite Terminalfenster.

```
Every 2.0s: oc get hpa/loadtest ...
NAME      REFERENCE          TARGETS  MINPODS  MAXPODS  REPLICAS  ...
loadtest  Deployment/loadtest  0%/50%    2         10        2          ...
```

- ▶ 7. Bereinigen Sie die Übungsumgebung, indem Sie das Projekt `schedule-scale` löschen.

```
[student@workstation ~]$ oc delete project schedule-scale
project.project.openshift.io "schedule-scale" deleted
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab schedule-scale finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Steuern der Pod-Zuordnung (Scheduling)

Performance-Checkliste

In dieser Übung konfigurieren Sie eine Anwendung so, dass sie auf einer Teilmenge der Cluster-Knoten ausgeführt wird und anhand von Last skaliert werden kann.

Ergebnisse

Sie sollten in der Lage sein, die OpenShift-Befehlszeilenschnittstelle zu verwenden, um:

- Knoten ein neues Label hinzuzufügen
- Pods auf Knoten bereitzustellen, die mit einem bestimmten Label übereinstimmen
- CPU- und Arbeitsspeicherressourcen für Pods anzufordern
- Eine Anwendung so zu konfigurieren, dass sie automatisch skaliert wird
- Ein Kontingent zur Beschränkung der Ressourcen, die ein Projekt verwenden kann, zu erstellen

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt ein Verzeichnis für die Übungsdateien.

```
[student@workstation ~]$ lab schedule-review start
```

Anweisungen

1. Kennzeichnen Sie als `admin`-Benutzer die beiden Knoten mit dem Label `tier`. Weisen Sie dem `master01`-Knoten das Label `tier=gold` und dem `master02`-Knoten das Label `tier=silver` zu.
2. Wechseln Sie zum Benutzer `developer`, und erstellen Sie ein neues Projekt mit dem Namen `schedule-review`.
3. Erstellen Sie mit dem Container unter `quay.io/redhattraining/loadtest:v1.0` eine neue Anwendung mit dem Namen `loadtest`. Die Anwendung `loadtest` sollte auf Knoten mit dem Label `tier=silver` bereitgestellt werden. Stellen Sie sicher, dass jeder Container `100 m` CPU und `20 Mi` Arbeitsspeicher anfordert.
4. Erstellen Sie mit dem standardmäßigen (automatisch generierten) Hostnamen eine Route zu Ihrer Anwendung mit dem Namen `loadtest`. Je nachdem, wie Sie Ihre Anwendung erstellt haben, müssen Sie möglicherweise einen Service erstellen, bevor Sie die Route erstellen. Ihre Anwendung funktioniert erwartungsgemäß, wenn die Ausführung von `curl`

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

`http://loadtest-schedule-review.apps.ocp4.example.com/api/loadtest/v1/healthz` die Meldung {"health": "ok"} zurückgibt.

5. Erstellen Sie einen horizontalen Pod-Autoscaler mit dem Namen `loadtest` für die Anwendung `loadtest`, die von 2 Pods auf maximal 40 Pods skaliert wird, wenn die CPU-Nutzung 70 % überschreitet. Sie können den horizontalen Pod-Autoscaler mit dem folgenden Befehl testen: `curl -X GET http://loadtest-schedule-review.apps.ocp4.example.com/api/loadtest/v1/cpu/3`



Anmerkung

Der horizontale Pod-Autoscaler skaliert die Anwendung `loadtest`, Ihr OpenShift-Cluster verfügt jedoch bereits über keine Ressourcen mehr, bevor das Maximum von 40 Pods erreicht ist.

6. Implementieren Sie als Benutzer `admin` für das Projekt `schedule-review` ein Kontingent mit dem Namen `review-quota`. Beschränken Sie das Projekt `schedule-review` auf maximal 1 vollständige CPU, 2 G Arbeitsspeicher und 20 Pods.

Bewertung

Führen Sie den folgenden `lab`-Befehl aus, um Ihre Arbeit zu überprüfen. Wenn der Befehl `lab` Fehler meldet, dann überprüfen Sie Ihre Änderungen, nehmen Sie Korrekturen vor, und führen Sie den Befehl `lab` so lange erneut aus, bis er erfolgreich ist.

```
[student@workstation ~]$ lab schedule-review grade
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab schedule-review finish
```

Hiermit ist die praktische Übung beendet.

► Lösung

Steuern der Pod-Zuordnung (Scheduling)

Performance-Checkliste

In dieser Übung konfigurieren Sie eine Anwendung so, dass sie auf einer Teilmenge der Cluster-Knoten ausgeführt wird und anhand von Last skaliert werden kann.

Ergebnisse

Sie sollten in der Lage sein, die OpenShift-Befehlszeilenschnittstelle zu verwenden, um:

- Knoten ein neues Label hinzuzufügen
- Pods auf Knoten bereitzustellen, die mit einem bestimmten Label übereinstimmen
- CPU- und Arbeitsspeicherressourcen für Pods anzufordern
- Eine Anwendung so zu konfigurieren, dass sie automatisch skaliert wird
- Ein Kontingent zur Beschränkung der Ressourcen, die ein Projekt verwenden kann, zu erstellen

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt ein Verzeichnis für die Übungsdateien.

```
[student@workstation ~]$ lab schedule-review start
```

Anweisungen

1. Kennzeichnen Sie als `admin`-Benutzer die beiden Knoten mit dem Label `tier`. Weisen Sie dem `master01`-Knoten das Label `tier=gold` und dem `master02`-Knoten das Label `tier=silver` zu.

- 1.1. Melden Sie sich als Benutzer `admin` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Ermitteln Sie, ob Knoten bereits das Label `tier` verwenden.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

```
[student@workstation ~]$ oc get nodes -L tier
NAME      STATUS    ROLES     AGE      VERSION      TIER
master01  Ready     master,worker  5d20h   v1.19.0+a5a0987
master02  Ready     master,worker  5d20h   v1.19.0+a5a0987
master03  Ready     master,worker  5d20h   v1.19.0+a5a0987
```

- 1.3. Kennzeichnen Sie den master01-Knoten mit dem Label tier=gold.

```
[student@workstation ~]$ oc label node master01 tier=gold
node/master01 labeled
```

- 1.4. Kennzeichnen Sie den master02-Knoten mit dem Label tier=silver.

```
[student@workstation ~]$ oc label node master02 tier=silver
node/master02 labeled
```

- 1.5. Vergewissern Sie sich, dass die Labels korrekt hinzugefügt wurden.

```
[student@workstation ~]$ oc get nodes -L tier
NAME      STATUS    ROLES     AGE      VERSION      TIER
master01  Ready     master,worker  5d20h   v1.19.0+a5a0987  gold
master02  Ready     master,worker  5d20h   v1.19.0+a5a0987  silver
master03  Ready     master,worker  5d20h   v1.19.0+a5a0987
```

2. Wechseln Sie zum Benutzer developer, und erstellen Sie ein neues Projekt mit dem Namen schedule-review.

- 2.1. Melden Sie sich als Benutzer developer bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u developer -p developer
Login successful.
...output omitted...
```

- 2.2. Erstellen Sie das Projekt schedule-review.

```
[student@workstation ~]$ oc new-project schedule-review
Now using project "schedule-review" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

3. Erstellen Sie mit dem Container unter quay.io/redhattraining/loadtest:v1.0 eine neue Anwendung mit dem Namen loadtest. Die Anwendung loadtest sollte auf Knoten mit dem Label tier=silver bereitgestellt werden. Stellen Sie sicher, dass jeder Container 100 m CPU und 20 Mi Arbeitsspeicher anfordert.

- 3.1. Um die anstehenden Anpassungen zu vereinfachen, erstellen Sie eine Bereitstellungsressourcendatei, ohne die Anwendung tatsächlich zu erstellen.

```
[student@workstation ~]$ oc create deployment loadtest --dry-run=client \
>   --image quay.io/redhattraining/loadtest:v1.0 \
>   -o yaml > ~/D0280/labs/schedule-review/loadtest.yaml
```

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

- 3.2. Bearbeiten Sie die Datei `~/D0280/labs/schedule-review/loadtest.yaml`, um einen Knoten-Selektor anzugeben. Fügen Sie die unten hervorgehobenen Zeilen hinzu, und achten Sie auf die korrekte Einrückung.

```
[student@workstation ~]$ vim ~/D0280/labs/schedule-review/loadtest.yaml

...output omitted...

spec:
  nodeSelector:
    tier: silver
  containers:
    - image: quay.io/redhattraining/loadtest:v1.0
      name: loadtest
      resources: {}
  status: {}
```

- 3.3. Fahren Sie mit der Bearbeitung von `~/D0280/labs/schedule-review/loadtest.yaml` fort. Ersetzen Sie die Zeile `resources: {}` durch die unten hervorgehobenen Zeilen. Vergewissern Sie sich vor dem Speichern der Datei, dass Sie die korrekte Einrückung verwendet haben.

```
...output omitted...

spec:
  nodeSelector:
    tier: silver
  containers:
    - image: quay.io/redhattraining/loadtest:v1.0
      name: loadtest
      resources:
        requests:
          cpu: "100m"
          memory: 20Mi
  status: {}
```

- 3.4. Erstellen Sie die Anwendung `loadtest`.

```
[student@workstation ~]$ oc create --save-config \
>   -f ~/D0280/labs/schedule-review/loadtest.yaml
deployment.apps/loadtest created
```

- 3.5. Überprüfen Sie, ob der Anwendungs-Pod ausgeführt wird. Möglicherweise müssen Sie den Befehl `oc get pods` mehrmals ausführen.

```
[student@workstation ~]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
loadtest-85f7669897-z4mq7   1/1     Running   0          53s
```

- 3.6. Stellen Sie sicher, dass der Anwendungs-Pod Ressourcenanforderungen angibt.

```
[student@workstation ~]$ oc describe pod/loadtest-85f7669897-z4mq7 \
>   | grep -A2 Requests
Requests:
  cpu:        100m
  memory:    20Mi
```

4. Erstellen Sie mit dem standardmäßigen (automatisch generierten) Hostnamen eine Route zu Ihrer Anwendung mit dem Namen `loadtest`. Je nachdem, wie Sie Ihre Anwendung erstellt haben, müssen Sie möglicherweise einen Service erstellen, bevor Sie die Route erstellen. Ihre Anwendung funktioniert erwartungsgemäß, wenn die Ausführung von `curl http://loadtest-schedule-review.apps.ocp4.example.com/api/loadtest/v1/healthz` die Meldung `{"health": "ok"}` zurückgibt.

- 4.1. Erstellen Sie einen Service, indem Sie die Verteilung für die Anwendung `loadtest` bereitstellen.

```
[student@workstation ~]$ oc expose deployment/loadtest \
>   --port 80 --target-port 8080
service/loadtest exposed
```

- 4.2. Erstellen Sie eine Route mit dem Namen `loadtest`, indem Sie den Service `loadtest` bereitstellen.

```
[student@workstation ~]$ oc expose service/loadtest --name loadtest
route.route.openshift.io/loadtest exposed
```

- 4.3. Ermitteln Sie den Hostnamen, der von der Route `loadtest` erstellt wurde.

```
[student@workstation ~]$ oc get route/loadtest
NAME      HOST/PORT          ...
loadtest  loadtest-schedule-review.apps.ocp4.example.com ...
```

- 4.4. Überprüfen Sie den Zugriff auf die Anwendung `loadtest` mit dem im vorherigen Schritt ermittelten Hostnamen.

```
[student@workstation ~]$ curl http://loadtest-schedule-review.\
> apps.ocp4.example.com/api/loadtest/v1/healthz
{"health": "ok"}
```

5. Erstellen Sie einen horizontalen Pod-Autoscaler mit dem Namen `loadtest` für die Anwendung `loadtest`, die von 2 Pods auf maximal 40 Pods skaliert wird, wenn die CPU-Nutzung 70 % überschreitet. Sie können den horizontalen Pod-Autoscaler mit dem folgenden Befehl testen: `curl -X GET http://loadtest-schedule-review.apps.ocp4.example.com/api/loadtest/v1/cpu/3`



Anmerkung

Der horizontale Pod-Autoscaler skaliert die Anwendung `loadtest`, Ihr OpenShift-Cluster verfügt jedoch bereits über keine Ressourcen mehr, bevor das Maximum von 40 Pods erreicht ist.

Kapitel 6 | Steuern der Pod-Zuordnung (Scheduling)

- 5.1. Erstellen Sie den horizontalen Pod-Autoscaler für die Anwendung `loadtest`.

```
[student@workstation ~]$ oc autoscale deployment/loadtest --name loadtest \
>   --min 2 --max 40 --cpu-percent 70
horizontalpodautoscaler.autoscaling/loadtest autoscaled
```

- 5.2. Warten Sie, bis der horizontale Pod-Autoscaler `loadtest` die standardmäßige Nutzung in der Spalte `TARGETS` angibt.

```
[student@workstation ~]$ watch oc get hpa/loadtest
```

Drücken Sie Strg+C, um den Befehl `watch` zu beenden, nachdem sich `<unknown>` in 0% geändert hat.

Every 2.0s: <code>oc get hpa/loadtest</code>						
NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	...
loadtest	Deployment/loadtest	0%/70%	2	40	2	...

**Anmerkung**

Es kann bis zu fünf Minuten dauern, bis der Eintrag `<unknown>` in der Spalte `TARGETS` in 0% geändert wird. Wenn der Eintrag `<unknown>` nicht geändert wird, verwenden Sie den Befehl `oc describe`, um zu überprüfen, ob die Container für die `loadtest`-Anwendung CPU-Ressourcen anfordern.

- 5.3. Testen Sie den horizontalen Pod-Autoscaler, indem Sie der CPU Last zuweisen. Verwenden Sie den zuvor ermittelten Hostnamen für die Route `loadtest`. Warten Sie, bis der Befehl `curl` abgeschlossen ist.

```
[student@workstation ~]$ curl -X GET http://loadtest-schedule-review.\
> apps.ocp4.example.com/api/loadtest/v1/cpu/3
```

- 5.4. Überprüfen Sie, ob zusätzliche Pods hinzugefügt wurden. Führen Sie den Befehl `oc get hpa/loadtest` mehrmals aus, bis die Änderungen berücksichtigt wurden. Ihre Ausgabe weicht wahrscheinlich ab, aber überprüfen Sie, ob die Replikanzahl größer als 2 ist.

[student@workstation ~]\$ oc get hpa/loadtest						
NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	...
loadtest	Deployment/loadtest	1043%/70%	2	40	21	...

6. Implementieren Sie als Benutzer `admin` für das Projekt `schedule-review` ein Kontingent mit dem Namen `review-quota`. Beschränken Sie das Projekt `schedule-review` auf maximal 1 vollständige CPU, 2 G Arbeitsspeicher und 20 Pods.

- 6.1. Melden Sie sich als Benutzer `admin` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat
Login successful.
...output omitted...
```

6.2. Erstellen Sie das Ressourcenkontingent.

```
[student@workstation ~]$ oc create quota review-quota \
>   --hard cpu="1",memory="2G",pods="20"
resourcequota/review-quota created
```



Anmerkung

Das Kontingent wirkt sich nicht auf vorhandene Pods aus, aber der Scheduler bewertet das Kontingent, wenn neue Ressourcen, wie Pods, angefordert werden.

Bewertung

Führen Sie den folgenden `lab`-Befehl aus, um Ihre Arbeit zu überprüfen. Wenn der Befehl `lab` Fehler meldet, dann überprüfen Sie Ihre Änderungen, nehmen Sie Korrekturen vor, und führen Sie den Befehl `lab` so lange erneut aus, bis er erfolgreich ist.

```
[student@workstation ~]$ lab schedule-review grade
```

Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab schedule-review finish
```

Hiermit ist die praktische Übung beendet.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Der Standard-Pod-Scheduler verwendet Regionen und Zonen, um Leistung und Redundanz zu erzielen.
- Das Kennzeichnen von Knoten und die Verwendung von Knoten-Selektoren beeinflussen die Platzierung von Pods.
- Ressourcenanforderungen definieren die minimale Menge an Ressourcen, die ein Pod benötigt, um zugeordnet zu werden.
- Kontingente begrenzen die Menge an Ressourcen, die ein Projekt nutzen darf.
- Angepasste Projektvorlagen können Kontingente und Einschränzungsbereiche für neue Projekte automatisch erstellen.
- Mit dem Befehl `oc scale` wird die Anzahl der Replikate eines Pods manuell skaliert.
- Horizontale Pod-Autoscaler skalieren Pod-Replikate dynamisch auf Grundlage der Last.

Kapitel 7

Beschreiben von Cluster-Updates

Ziel

Beschreiben der Durchführung eines Cluster-Updates

Ziele

Beschreiben des Prozesses für Cluster-Updates

Abschnitte

Beschreiben des Prozesses für Cluster-Updates (und Test)



Beschreiben des Prozesses für Cluster-Updates

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, den Prozess für das Update eines Clusters zu beschreiben.

Einführung in Cluster-Updates

Red Hat OpenShift Container Platform 4 bietet durch die Verwendung von Red Hat Enterprise Linux CoreOS viele neue Features. Red Hat hat ein neues Software-Distributionssystem veröffentlicht, das den besten Upgrade-Pfad für Updates Ihres Clusters und des zugrunde liegenden Betriebssystems bietet. Dieses neue Distributionssystem ist einer der wichtigsten Vorteile der OpenShift 4-Architekturänderungen und ermöglicht das Upgrade von Clustern Over-the-Air (OTA).

Dieses Software-Distributionssystem für OTA verwaltet die Controller-Manifeste, Cluster-Rollen und alle anderen Ressourcen, die für das Update eines Clusters auf eine bestimmte Version erforderlich sind.

Mit diesem Feature wird sichergestellt, dass ein Cluster die neueste verfügbare Version nahtlos ausführen kann. Darüber hinaus ermöglicht OTA, dass Cluster neue Features verwenden können, sobald sie verfügbar sind, einschließlich der letzten Fehlerkorrekturen und Sicherheits-Patches. OTA reduziert die Ausfallzeiten aufgrund von Upgrades erheblich.

Red Hat hostet und verwaltet diesen Service unter <https://cloud.redhat.com/openshift> und hostet Cluster-Images unter <https://quay.io>.



Wichtig

Ab OpenShift 4.6 erfordert das OTA-System eine persistente Verbindung mit dem Internet. Es ist nicht möglich, dieses Feature On-premises bereitzustellen.

Weitere Informationen zur Aktualisierung getrennter Cluster finden Sie in der Anleitung zum *Update* und im Kapitel *Installation und Konfiguration* im Abschnitt mit den Referenzen.

OTA ermöglicht schnellere Updates, indem Zwischenversionen übersprungen werden können. Sie können z. B. ein Update von 4.6.1 auf 4.6.3 durchführen und dabei 4.6.2 umgehen.

Sie verwenden zur Verwaltung des Lifecycle aller OpenShift-Cluster eine einzige Schnittstelle (<https://cloud.redhat.com/openshift>).

	Status	Type	Created	Version	Provider (...)
	Ready	OCP	10 Mar 2020	4.6.21 Update	AWS (us-west-1)
	Ready	OCP	Evaluation expired	4.6.19 Update	OpenStack
	Ready	OCP	60-day trial	4.6.24	OpenStack
	Ready	OCP	Evaluation expired	4.6.7 Update	AWS (us-east-2)

Abbildung 7.1: Verwalten von Clustern auf cloud.redhat.com

Der Service definiert *Upgrade-Pfade*, die der Cluster-Berechtigung für bestimmte Updates entsprechen.

Upgrade-Pfade sind Bestandteil von Upgrade-Kanälen. Sie können sich den Upgrade-Pfad wie einen Kanal vorstellen. Der Kanal steuert die Häufigkeit und Stabilität von Updates. Die OTA-Richtlinien-Engine stellt Kanäle als eine Reihe von Zeigern auf bestimmte Versionen innerhalb des Upgrade-Pfads dar.

Ein Kanalname besteht aus drei Teilen: der Ebene („release candidate“, „fast“ und „stable“), der Hauptversion (4) und der Nebenversion (.2). Beispiele für Kanalnamen: **stable-4.6**, **fast-4.6**, **eus-4.6** und **candidate-4.6**. Jeder Kanal liefert Patches für eine bestimmte Cluster-Version.

Beschreiben des candidate-Kanals

Der *candidate*-Kanal liefert Updates, um die Akzeptanz von Funktionen in der nächsten Version von OpenShift Container Platform zu testen. Die Release Candidate-Versionen unterliegen weiteren Prüfungen und werden zu den fast- oder stable-Kanälen heraufgestuft, wenn sie die Qualitätsstandards erfüllen.



Anmerkung

Die im *candidate*-Kanal aufgeführten Updates werden von Red Hat nicht unterstützt.

Beschreiben des fast-Kanals

Der *fast*-Kanal liefert Updates, sobald sie verfügbar sind. Dieser Kanal eignet sich am besten für Produktions- und QS-Umgebungen. Mit dem Kanal **fast-4.6** können Sie ein Upgrade von einer früheren Nebenversion von OpenShift Container Platform durchführen.



Anmerkung

Kunden können bei der Verbesserung von OpenShift mithelfen, indem sie dem Red Hat-Programm für vernetzte Kunden beitreten. Wenn Sie diesem Programm beitreten, wird Ihr Cluster für den fast-Kanal registriert.

Beschreiben des stable-Kanals

Der stable-Kanal enthält verzögerte Updates, liefert nur kleinere Updates für eine bestimmte Cluster-Version und ist besser für Produktionsumgebungen geeignet.

Die Teams von Red Hat für Support und Site Reliability Engineering (SRE) überwachen betriebsbereite Cluster mit neuen schnellen Updates. Wenn betriebsbereite Cluster zusätzliche Tests und Validierungen bestehen, werden Updates im fast-Kanal im stable-Kanal aktiviert.

Wenn Red Hat bei einem Update aus dem stable-Kanal Betriebsprobleme feststellt, wird dieses Update im stable-Kanal übersprungen. Die Verzögerung des stable-Kanals bietet Zeit, um unvorhergesehene Probleme in tatsächlichen OpenShift-Clustern festzustellen, die von Tests nicht aufgedeckt wurden.

Beschreiben des Extended Update Support-Kanals

Der Extended Update Support (EUS)-Kanal wird Kunden mit Premium-Abonnements in bestimmten Nebenversionen von OpenShift Container Platform angeboten. Die EUS-Versionen verlängern die Wartungsphase auf 14 Monate. Es gibt keinen Unterschied zwischen den Kanälen **stable-4.6** und **eus-4.6**. Sie können zum EUS-Kanal wechseln, sobald er verfügbar ist.



Anmerkung

Nach dem Upgrade auf eine Version, die exklusiv für den EUS-Kanal zur Verfügung steht, kann dieser Cluster erst dann Upgrades der Nebenversion durchführen, wenn Upgrades auf die nächste EUS-Version verfügbar sind.

Beschreiben von Upgrade-Pfaden

Im Folgenden wird beschrieben, wie diese Upgradepfade auf Red Hat OpenShift Container Platform Version 4.6 angewendet werden:

- Wenn Sie den Kanal **stable-4.6** verwenden, können Sie Ihren Cluster von 4.6.0 auf 4.6.1 oder 4.6.2 aktualisieren. Wenn in Version 4.6.3 ein Problem festgestellt wird, können Sie nicht auf diese Version upgraden. Wenn ein Patch in der Version 4.6.4 verfügbar ist, können Sie Ihren Cluster auf diese Version aktualisieren.

Dieser Kanal eignet sich für Produktionsumgebungen, da die Releases in diesem Kanal von Red Hat SREs und Support-Services getestet wurden.

- Der Kanal **fast-4.5** kann Updates für 4.5.1 und 4.5.2 ausliefern, aber nicht für 4.6.1. Dieser Kanal wird ebenfalls von Red Hat unterstützt und kann in Produktionsumgebungen eingesetzt werden.

Administratoren müssen einen anderen Kanal für Nebenversionen auswählen, z. B. **fast-4.6**, um auf eine neue Version in einer neuen Nebenversion zu aktualisieren.

- Mit dem Kanal **candidate-4.6** können Sie die neuesten Funktionen von OpenShift installieren. Mit diesem Kanal können Sie auf alle *z-stream*-Releases wie 4.6.1, 4.6.2, 4.6.3 usw. aktualisieren.

Mit diesem Kanal haben Sie Zugriff auf die neuesten Funktionen des Produkts, sobald diese veröffentlicht werden. Dieser Kanal eignet sich für Entwicklungs- und Vorproduktionsumgebungen.

- Beim Wechsel zum Kanal **eus-4.6** erhält der Kanal **stable-4.6** keine *z-stream*-Updates, bis die nächste EUS-Version verfügbar ist. Die nächste geplante EUS-Version ist 4.10. Ein Upgrade auf diese Version erfordert eine Reihe kleiner Upgrades, z. B. von 4.6 auf 4.7 usw., bis 4.10 erreicht ist.

In der folgenden Grafik werden die Update-Diagramme für stable - und candidate-Kanäle beschrieben:

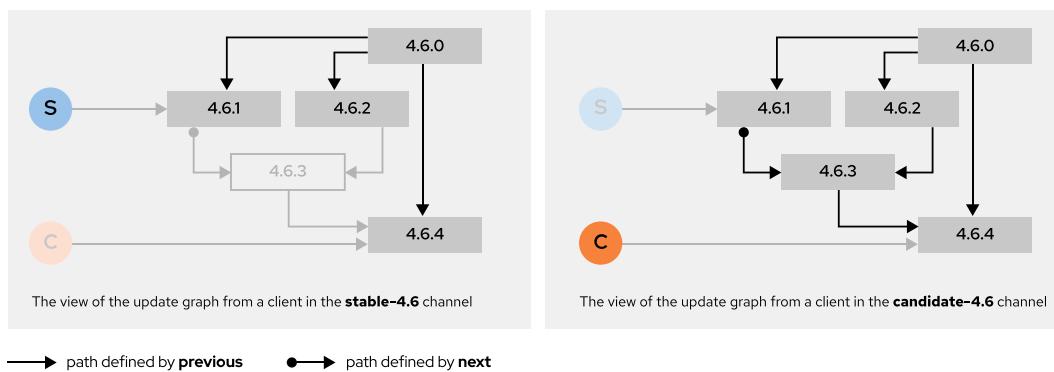


Abbildung 7.2: Aktualisieren von Diagrammen für stable- und candidate-Kanäle

Die Kanäle **stable** (stabil) und **fast** (schnell) werden als allgemein verfügbar (General Availability, GA) klassifiziert, während der Kanal **candidate** (Kandidat) nicht von Red Hat unterstützt wird.

Um die Stabilität des Clusters und den richtigen Support zu gewährleisten, sollten Sie nur von einem stable-Kanal zu einem fast-Kanal oder umgekehrt wechseln. Es ist zwar möglich, von einem stable-Kanal oder einem fast-Kanal zu einem candidate-Kanal zu wechseln, allerdings wird davon abgeraten. Der candidate-Kanal eignet sich am besten, um die Akzeptanz von Funktionen zu testen und Unterstützung bei der Qualifizierung der nächsten Version von OpenShift Container Platform zu bieten.



Anmerkung

Die Veröffentlichung von Updates für Patch- und CVE-Korrekturen reicht von mehreren Stunden bis zu einem Tag. Diese Verzögerung bietet Zeit für die Bewertung etwaiger betrieblicher Auswirkungen auf OpenShift-Cluster.

Ändern des Update-Kanals

Sie können den Update-Kanal mit der Web Console oder dem OpenShift-CLI-Client zu **stable-4.6**, **fast-4.6** oder **candidate-4.6** ändern:

- Navigieren Sie in der Web Console zur Seite **Administration** → **Cluster Settings** auf der Registerkarte „Details“, und klicken Sie dann auf das Stift-Symbol.

Abbildung 7.3: Aktueller Update-Kanal in der Web Console

In einem Fenster werden Optionen zum Auswählen eines Update-Kanals angezeigt.

Abbildung 7.4: Ändern des Update-Kanals in der Web Console

- Führen Sie den folgenden Befehl aus, um mit dem oc-Client zu einem anderen Update-Kanal zu wechseln. Sie können auch zu einem anderen Update-Kanal wechseln, z. B. stable-4.6, um auf die nächste Nebenversion von OpenShift Container Platform zu aktualisieren.

```
[user@host ~]$ oc patch clusterversion version --type="merge" --patch \
>   '{"spec":{"channel":"fast-4.6"}}'
clusterversion.config.openshift.io/version patched
```

Beschreiben von OTA

OTA folgt einem Client-Server-Ansatz. Red Hat hostet die Cluster-Images und die Update-Infrastruktur. Ein Feature von OTA ist die Generierung aller möglichen Update-Pfade für Ihren Cluster. OTA sammelt Informationen über den Cluster und Ihre Berechtigung, um verfügbare Upgrade-Pfade zu bestimmen. Die Web Console sendet eine Benachrichtigung, wenn ein neues Update verfügbar ist.

Das folgende Diagramm zeigt die Architektur von Updates: Red Hat hostet sowohl die Cluster-Images als auch einen „Watcher“, der automatisch neue Images erkennt, die an Quay weitergeleitet werden. Der *Cluster Version Operator* (CVO) erhält den Update-Status von diesem Watcher. Der CVO beginnt mit der Aktualisierung der Cluster-Komponenten über ihre Operatoren und aktualisiert dann alle zusätzlichen Komponenten, die vom *Operator Lifecycle Manager* (OLM) verwaltet werden.

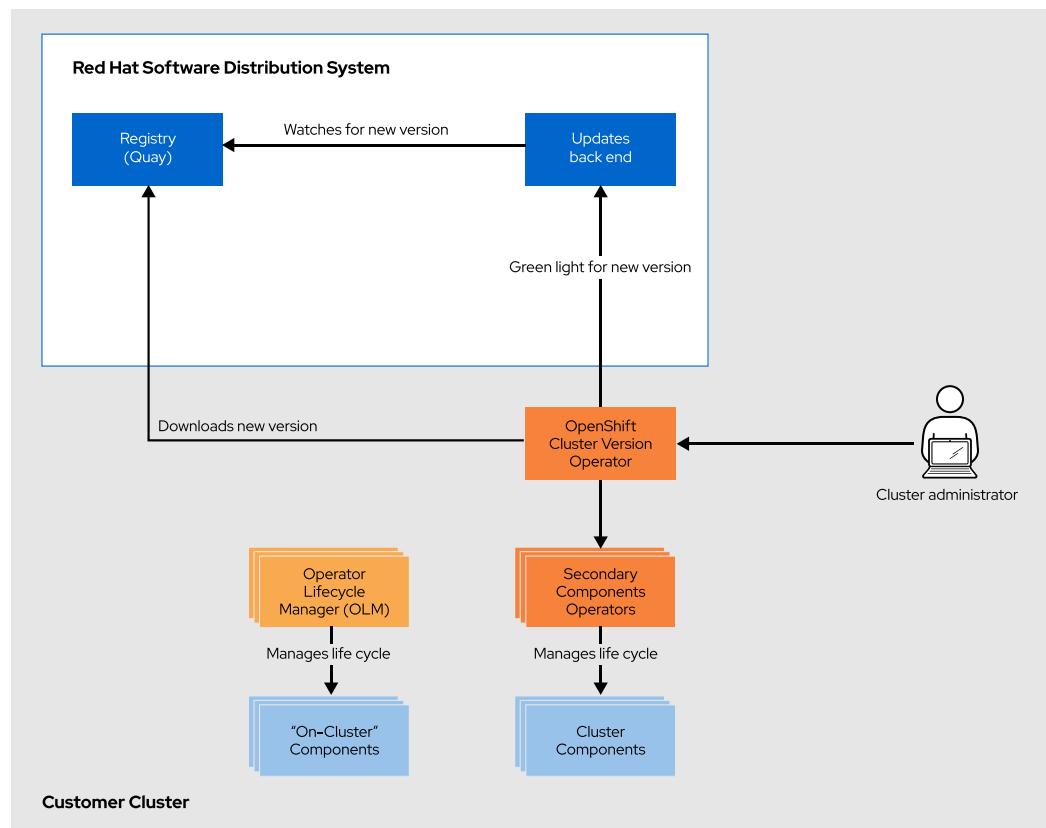


Abbildung 7.5: Architektur der Updates für OpenShift Container Platform

Red Hat bestimmt anhand von Telemetrie den Update-Pfad. Der Cluster verwendet die auf Prometheus basierende Telemetrie, um den Status jedes Cluster-Operators zu melden. Die Daten werden anonymisiert und an Red Hat-Server zurückgesendet, die Cluster-Administratoren über potenzielle neue Releases informieren.

Anmerkung

Red Hat ist der Schutz von Kundendaten sehr wichtig. Eine vollständige Liste der Daten, die von Telemeter gesammelt werden, finden Sie im Dokument *Sample Metrics*, das im Abschnitt „References“ aufgeführt ist.

Red Hat beabsichtigt in Zukunft, die Liste der aktualisierten, im Upgrade-Pfad enthaltenen Operatoren zu erweitern, um unabhängige Softwareanbieter (ISV) einzubeziehen.

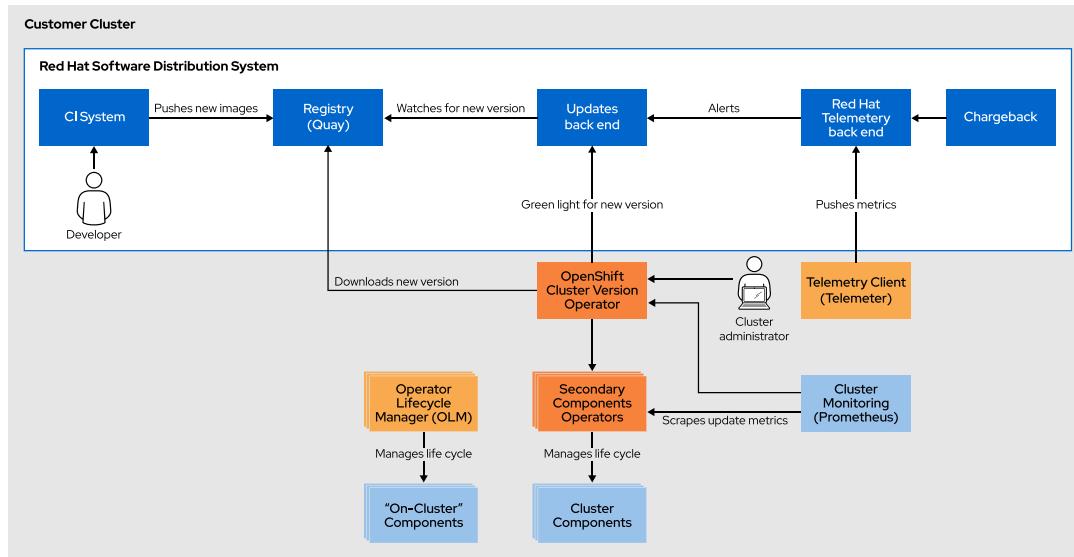


Abbildung 7.6: Verwalten von Cluster-Updates mit Telemetrie

Erläutern des Update-Prozesses

Am Cluster-Update-Prozess sind zwei Komponenten beteiligt:

Machine Config-Operator

Der Machine Config Operator wendet den gewünschten Rechnerstatus auf die einzelnen Knoten an. Diese Komponente verarbeitet auch das Rolling Upgrade von Knoten im Cluster und verwendet CoreOS Ignition als Konfigurationsformat.

Operator Lifecycle Manager (OLM)

Der Operator Lifecycle Manager (OLM) orchestriert Updates für alle Operatoren, die im Cluster ausgeführt werden.

Aktualisieren des Clusters

Sie können den Cluster über die Web Console oder über die Befehlszeile aktualisieren. Die Aktualisierung über die Web Console ist einfacher als die Verwendung der Befehlszeile. Auf der Seite **Administration → Cluster Settings** wird der **Update Status** des Typs **Update available** angezeigt, wenn ein neues Update verfügbar ist. Klicken Sie auf dieser Seite auf **Update now**, um den Prozess zu starten:

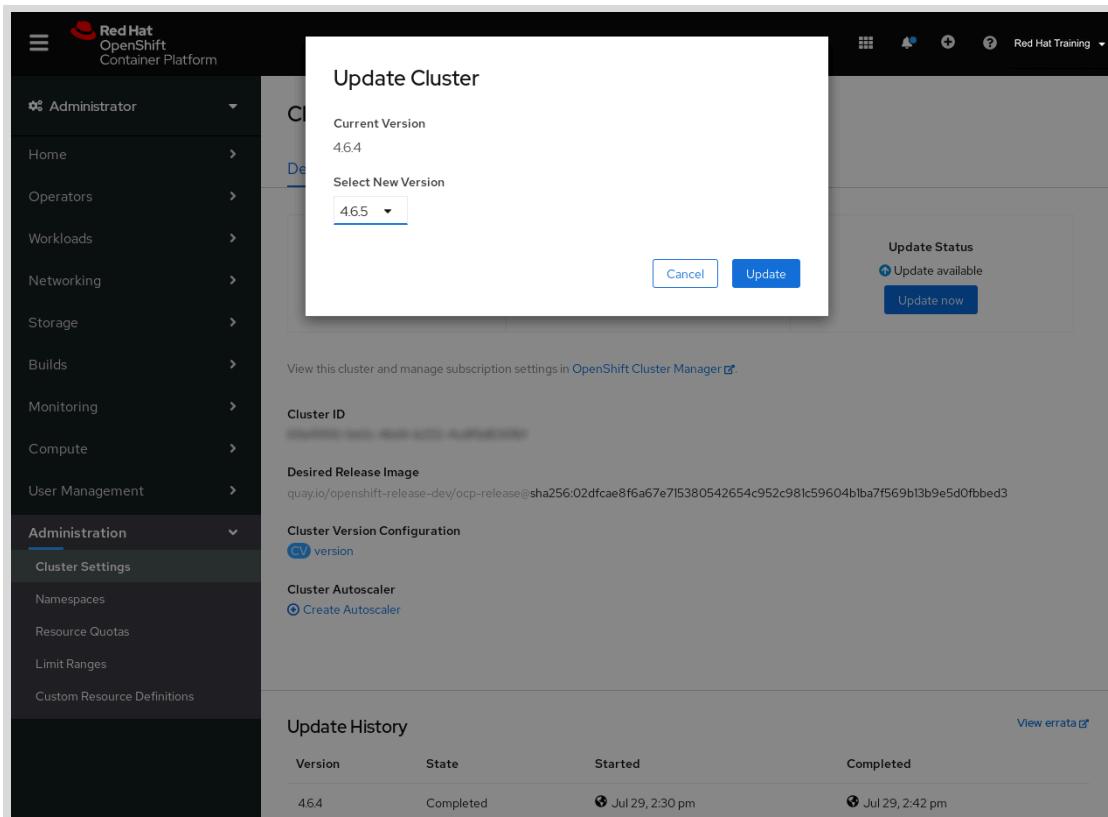


Abbildung 7.7: Aktualisieren des Clusters mit der Web Console



Wichtig

Red Hat unterstützt keine Rollbacks oder das Zurücksetzen des Clusters auf eine frühere Version. Red Hat unterstützt nur Upgrades auf eine neuere Version.

Der Update-Prozess aktualisiert auch das zugrunde liegende Betriebssystem, wenn Updates dafür verfügbar sind. Der Prozess verwendet die Technologie `rpm-ostree` für die Verwaltung von Transaktions-Upgrades. Updates werden über Container-Images bereitgestellt und sind Teil des OpenShift-Update-Prozesses. Wenn das Update bereitgestellt wird, rufen die Knoten das neue Image ab, extrahieren es, schreiben die Pakete auf die Disk und ändern dann den Bootloader so, dass er in die neue Version bootet. Der Rechner wird neu gebootet und implementiert ein Rolling Update, um sicherzustellen, dass die Cluster-Kapazität nur minimal wirkt.

In den folgenden Schritten wird das Verfahren beschrieben, das ein Cluster-Administrator zum Aktualisieren eines Clusters über die Befehlszeilenschnittstelle ausführt:

1. Aktualisieren Sie alle über den Operator Lifecycle Manager (OLM) installierten Operatoren auf die neueste Version, bevor Sie den OpenShift-Cluster aktualisieren.
2. Rufen Sie die Cluster-Version ab, überprüfen Sie die aktuellen Informationen zum Update-Kanal, und bestätigen Sie den Kanal. Wenn Sie den Cluster in der Produktion ausführen, muss der Kanal `stable` lauten.

Kapitel 7 | Beschreiben von Cluster-Updates

```
[user@host ~]$ oc get clusterversion
NAME      VERSION  AVAILABLE  PROGRESSING  SINCE    STATUS
version   4.6.4     True       False        5d       Cluster version is 4.6.4

[user@host ~]$ oc get clusterversion -o json | jq ".items[0].spec.channel"
"stable-4.6"
```

3. Zeigen Sie die verfügbaren Updates an, und notieren Sie sich die Versionsnummer des Updates, das Sie anwenden möchten.

```
[user@host ~]$ oc adm upgrade
Cluster version is 4.6.4

Updates:

VERSION IMAGE
4.6.5 quay.io/openshift-release-dev/ocp-release@sha256:...
...output omitted...
```

4. Wenden Sie das neueste Update auf Ihren Cluster an, oder aktualisieren Sie auf eine bestimmte Version:
 - Führen Sie den folgenden Befehl aus, um das neueste verfügbare Update für Ihren Cluster zu installieren.

```
[user@host ~]$ oc adm upgrade --to-latest=true
```

- Führen Sie den folgenden Befehl aus, um eine bestimmte Version zu installieren.
VERSION entspricht einer der verfügbaren Versionen, die vom Befehl `oc adm upgrade` zurückgegeben werden.

```
[user@host ~]$ oc adm upgrade --to=VERSION
```

5. Mit dem vorherigen Befehl wird der Update-Prozess initialisiert. Führen Sie den folgenden Befehl aus, um den Status des Cluster Version Operator (CVO) und der installierten Cluster-Operatoren zu überprüfen.

```
[user@host ~]$ oc get clusterversion
NAME      VERSION  AVAILABLE  PROGRESSING  SINCE    STATUS
version   4.6.4     True       True        32m     Working towards 4.6.5 ...

[user@host ~]$ oc get clusteroperators
NAME                  VERSION  AVAILABLE  PROGRESSING  DEGRADED
authentication        4.6.4     True       False        False
cloud-credential      4.6.5     False      True        False
openshift-apiserver  4.6.5     True       False        True
...output omitted...
```

6. Mit dem folgenden Befehl können Sie den Verlauf des Cluster-Versionsstatus überprüfen, um den Status des Updates zu überwachen. Es kann einige Zeit dauern, bis die Aktualisierung aller Objekte abgeschlossen ist.

Kapitel 7 | Beschreiben von Cluster-Updates

Der Verlauf enthält eine Liste der neuesten Versionen, die auf den Cluster angewendet wurden. Dieser Wert wird aktualisiert, wenn der CVO ein Update anwendet. Die Liste ist nach Datum sortiert, wobei das neueste Update oben in der Liste aufgeführt ist.

Wenn der Rollout erfolgreich war, haben die Updates im Verlauf den Status **Completed**. Das Update weist den Status **Partial** auf, wenn das Update fehlgeschlagen ist oder nicht abgeschlossen wurde.

```
[user@host ~]$ oc describe clusterversion
...output omitted...
History:
  Completion Time: 2020-09-28T16:02:18Z
  Image:          quay.io/openshift-release-dev/ocp-release@sha256:...
  Started Time:   2020-09-28T15:31:13Z
  State:          Completed
  Verified:       true
  Version:        4.6.5
  Completion Time: 2020-08-05T18:35:08Z
  Image:          quay.io/openshift-release-dev/ocp-release@sha256:...
  Started Time:   2020-08-05T18:22:42Z
  State:          Completed
  Verified:       true
  Version:        4.6.4
  Observed Generation: 5
  Version Hash:   AF5-oev9wI=
  Events:         none
```



Wichtig

Wenn ein Update fehlschlägt, hält der Operator an und meldet den Status der fehlerhaften Komponente. Das Zurücksetzen des Clusters auf eine frühere Version wird nicht unterstützt. Wenn Ihr Update fehlschlägt, wenden Sie sich an den Support von Red Hat.

- Nach Abschluss des Updates können Sie überprüfen, ob die Cluster-Version auf die neue Version aktualisiert wurde.

```
[user@host ~]$ oc get clusterversion
NAME      VERSION      AVAILABLE      PROGRESSING      SINCE      STATUS
version   4.6.5      True          False           11m       Cluster version is 4.6.5
```



Literaturhinweise

Weitere Informationen zur Installation von Red Hat OpenShift Container Platform in einer getrennten Umgebung finden Sie im Kapitel *Installation configuration* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Installing* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/installing/index#installation-configuration

Weitere Informationen zu Update-Kanälen, Voraussetzungen für Updates und zum Aktualisieren von Clustern in getrennten Umgebungen finden Sie in den Kapiteln *Updating a restricted network cluster* und *Updating a cluster between minor versions* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Updating clusters* unter

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/updating_clusters/index#updating-restricted-network-cluster

Weitere Informationen zur Aktualisierung der über den Operator Lifecycle Manager (OLM) installierten Operatoren finden Sie im Abschnitt *Upgrade installed Operators* im Kapitel *Administrator tasks* in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Working with Operators* unter https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/operators/index#olm-upgrading-operators

Weitere Informationen zu den Upgrade-Pfaden für OpenShift Container Platform finden Sie auf der folgenden Seite im Customer Portal:
<https://access.redhat.com/solutions/4583231/>

Beispielmetriken

<https://github.com/openshift/cluster-monitoring-operator/blob/master/Documentation/sample-metrics.md>

Cincinnati

<https://github.com/openshift/cincinnati/blob/master/docs/design/cincinnati.md#cincinnati>

► Quiz

Beschreiben des Prozesses für Cluster-Updates

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Welche zwei der folgenden Updates wären im Update-Kanal fast - 4 . 6 verfügbar?
(Wählen Sie zwei Antworten aus.)
 - a. 4.5.2
 - b. 4.6.1
 - c. 4.7.1
 - d. 4.6.5

- ▶ 2. Welche der folgenden Komponenten ruft die aktualisierten Cluster-Images von Quay.io ab?
 - a. Cluster Monitoring (Prometheus)
 - b. Operator Lifecycle Manager (OLM)
 - c. Cluster Version Operator (CVO)
 - d. Telemetrie-Client (Telemeter)

- ▶ 3. Welche der folgenden Komponenten verwaltet die Updates von Operatoren, die keine Cluster-Operatoren sind?
 - a. Operator Lifecycle Manager (OLM)
 - b. Telemetrie-Client (Telemeter)
 - c. Cluster Version Operator (CVO)

- ▶ 4. Mit welchen zwei der folgenden Befehle können Sie die derzeit ausgeführte Version des Clusters abrufen? **(Wählen Sie zwei Antworten aus.)**
 - a. oc adm upgrade
 - b. oc get clusterchannel
 - c. oc get clusterversion

► **5. Welche der folgenden Aussagen ist in Bezug auf das OTA-Feature richtig?**

- a. Der *stable*-Kanal ist als allgemein verfügbar (General Availability, GA) klassifiziert, während der *fast*-Kanal als Release Candidate (RC) klassifiziert ist.
- b. Bei der Verwendung des *stable*-Kanals können keine Zwischenversionen übersprungen werden. Beispielsweise installiert OpenShift bei einer Aktualisierung von 4.3.27 auf 4.3.29 zuerst die Version 4.3.28.
- c. Es wird nicht empfohlen, von einem *stable*-Kanal oder einem *fast*-Kanal zu einem *candidate*-Kanal zu wechseln. Sie können jedoch von einem *fast*- zu einem *stable*-Kanal wechseln und umgekehrt.
- d. Das Rollback eines fehlgeschlagenen Updates wird nur von Red Hat unterstützt, wenn Sie versuchen, von einer z-stream-Version auf eine andere zu aktualisieren (z. B. von 4.5.2 auf 4.5.3, jedoch nicht von 4.5.3 auf 4.6).

► **6. Welche zwei der folgenden Kanäle werden als allgemein verfügbar klassifiziert?**

(Wählen Sie zwei Antworten aus.)

- a. candidate-4.6.stable
- b. stable-4.6
- c. candidate-stable-4.6
- d. fast-4.6
- e. fast-4.6.1

► Lösung

Beschreiben des Prozesses für Cluster-Updates

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- ▶ 1. Welche zwei der folgenden Updates wären im Update-Kanal fast - 4 . 6 verfügbar?
(Wählen Sie zwei Antworten aus.)
 - a. 4.5.2
 - b. 4.6.1
 - c. 4.7.1
 - d. 4.6.5
- ▶ 2. Welche der folgenden Komponenten ruft die aktualisierten Cluster-Images von Quay.io ab?
 - a. Cluster Monitoring (Prometheus)
 - b. Operator Lifecycle Manager (OLM)
 - c. Cluster Version Operator (CVO)
 - d. Telemetrie-Client (Telemeter)
- ▶ 3. Welche der folgenden Komponenten verwaltet die Updates von Operatoren, die keine Cluster-Operatoren sind?
 - a. Operator Lifecycle Manager (OLM)
 - b. Telemetrie-Client (Telemeter)
 - c. Cluster Version Operator (CVO)
- ▶ 4. Mit welchen zwei der folgenden Befehle können Sie die derzeit ausgeführte Version des Clusters abrufen? **(Wählen Sie zwei Antworten aus.)**
 - a. oc adm upgrade
 - b. oc get clusterchannel
 - c. oc get clusterversion

► **5. Welche der folgenden Aussagen ist in Bezug auf das OTA-Feature richtig?**

- a. Der *stable*-Kanal ist als allgemein verfügbar (General Availability, GA) klassifiziert, während der *fast*-Kanal als Release Candidate (RC) klassifiziert ist.
- b. Bei der Verwendung des *stable*-Kanals können keine Zwischenversionen übersprungen werden. Beispielsweise installiert OpenShift bei einer Aktualisierung von 4.3.27 auf 4.3.29 zuerst die Version 4.3.28.
- c. Es wird nicht empfohlen, von einem *stable*-Kanal oder einem *fast*-Kanal zu einem *candidate*-Kanal zu wechseln. Sie können jedoch von einem *fast*- zu einem *stable*-Kanal wechseln und umgekehrt.
- d. Das Rollback eines fehlgeschlagenen Updates wird nur von Red Hat unterstützt, wenn Sie versuchen, von einer z-stream-Version auf eine andere zu aktualisieren (z. B. von 4.5.2 auf 4.5.3, jedoch nicht von 4.5.3 auf 4.6).

► **6. Welche zwei der folgenden Kanäle werden als allgemein verfügbar klassifiziert?**

(Wählen Sie zwei Antworten aus.)

- a. candidate-4.6.stable
- b. stable-4.6
- c. candidate-stable-4.6
- d. fast-4.6
- e. fast-4.6.1

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Einer der wichtigsten Vorteile von OpenShift 4-Architekturänderungen besteht darin, dass Sie Ihre Cluster Over-the-Air (OTA) aktualisieren können.
- Red Hat bietet ein neues Software-Distributionssystem, das den besten Pfad für Updates Ihres Clusters und des zugrunde liegenden Betriebssystems sicherstellt.
- Es gibt verschiedene Distributionskanäle:
 - Der *stable*-Kanal liefert verzögerte Updates.
 - Der *fast*-Kanal liefert Updates, sobald sie verfügbar sind.
 - Der *candidate*-Kanal liefert Updates, um die Akzeptanz von Funktionen in der nächsten Version von OpenShift Container Platform zu testen.
 - Der Kanal *eus* (nur bei Ausführung von 4.6 verfügbar) erweitert die Wartungsphase für Kunden mit Premium-Abonnements.
- Red Hat unterstützt das Zurücksetzen des Clusters auf eine frühere Version nicht. Red Hat unterstützt nur Upgrades auf eine neuere Version.

Kapitel 8

Verwalten eines Clusters mit der Web Console

Ziel

Verwalten eines Red Hat OpenShift-Clusters mit der Web Console

Zielsetzungen

- Durchführen der Cluster-Verwaltung mit der Web Console
- Verwalten von Anwendungen und Kubernetes-Operatoren mit der Web Console
- Untersuchen von Performance- und Integritätsmetriken für Cluster-Knoten und Anwendungen

Abschnitte

- Durchführen der Cluster-Verwaltung (und angeleitete Übung)
- Verwalten von Workloads und Operatoren (und angeleitete Übung)
- Untersuchen von Cluster-Metriken (und angeleitete Übung)

Praktische Übung

Verwalten eines Clusters mit der Web Console

Durchführen der Cluster-Verwaltung

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, die Cluster-Verwaltung mit der Web Console durchzuführen.

Beschreiben der Web Console

Die Web Console von Red Hat OpenShift bietet eine grafische Benutzeroberfläche für administrative, Management- und Fehlerbehebungsaufgaben. Sie unterstützt die Perspektiven **Administrator** und **Developer**. In diesem Kurs wird die Perspektive **Administrator** erläutert.

In der folgenden Liste werden einige der wichtigsten Bestandteile der Web Console beschrieben, die entsprechend den Hauptelementen des Navigationsmenüs gruppiert sind. Die Sichtbarkeit der einzelnen Elemente hängt von den Rollen und Rollenbindungen des jeweiligen Benutzers ab. Benutzer mit der Cluster-Rolle `cluster-admin` können alles anzeigen und bearbeiten. Benutzer mit der Cluster-Rolle `view` können die meisten Elemente anzeigen, können jedoch keine Änderungen vornehmen. Zusätzliche Rollen können den Zugriff auf bestimmte Elemente ermöglichen.

Home

Die Seite **Home** → **Overview** bietet eine schnelle Übersicht über den Cluster, einschließlich der Integritätsmetriken, der Ressourcenanzahl und einer Streaming-Liste von Ereignissen, z. B. Rechner-Updates oder Pod-Fehler.

Sie können zur Seite **Home** → **Search** navigieren, um Ressourcen eines beliebigen Typs zu suchen oder zu erstellen. Diese Seite ist auch ein nützlicher Ausgangspunkt, um zu Ressourcen zu navigieren, für die keine dedizierte Navigation im Menü verfügbar ist.

Auf der Seite **Home** → **Events** wird ein filterbarer Stream von Ereignissen angezeigt, die im Cluster auftreten. Diese Seite ist zudem ein guter Ausgangspunkt für die Fehlerbehebung.

Operatoren

Untersuchen und installieren Sie mit **OperatorHub** die von Red Hat zusammengestellten Operatoren, und navigieren Sie dann zur Seite **Installed Operators**, um die Operatoren zu verwalten.

Workloads, Networking und Storage

Verwalten Sie gängige Ressourcen wie Bereitstellungen, Services und persistente Volumes. Besonders hilfreich für die Fehlerbehebung ist die Möglichkeit, Pod-Protokolle anzuzeigen und eine Verbindung mit einem Terminal herzustellen.

Builds

Verwalten Sie Build-Konfigurationen, Builds und Image-Streams.

Überwachung

Zeigen Sie Alarme an und führen Sie Ad-hoc-Abfragen mit Prometheus durch.

Compute

Zeigen Sie Rechenressourcen wie Knoten, Rechner und MachineAutoscaler an, und verwalten Sie diese.

Benutzerverwaltung

Anzeigen und Verwalten von Benutzern, Gruppen, Servicekonten, Rollen und Rollenbindungen.

Administration

Zeigen Sie eine Vielzahl von Einstellungen an, die besonders für Cluster-Administratoren interessant sind, z. B. Cluster-Updates, Cluster-Operatoren, CRDs sowie Ressourcenkontingente, und verwalten Sie diese.

Zugriff auf die OpenShift Web Console

Die OpenShift Web Console wird als Pods im Projekt `openshift-console` ausgeführt und durch einen Operator verwaltet, der im Projekt `openshift-console-operator` ausgeführt wird. Sie können die URL auflisten, indem Sie die Route auflisten:

```
[user@host ~]$ oc get routes -n openshift-console
NAME      HOST/PORT          ... PORT ...
console   console-openshift-console.apps.cluster.example.com  ... https ...
downloads downloads-openshift-console.apps.cluster.example.com ... http  ...
```

In Nichtproduktionssystemen werden in der Regel selbstsignierte Zertifikate für den HTTPS-Endpunkt verwendet. Webbrowser warnen Sie vor dem Zertifikat, und Sie müssen eine Sicherheitsausnahme hinzufügen, wenn Sie zum ersten Mal zur Web Console navigieren.

Suchen nach Ressourcen

Die Web-Benutzeroberfläche bietet mehrere Möglichkeiten, Ressourcen zu suchen. Viele gängige Ressourcen, z. B. Deployments und Services, stehen im Hauptmenü auf der linken Seite zur Verfügung. Sie können auf der Seite **Home → Search** nach anderen Ressourcentypen suchen. Diese Seite bietet ein vollständiges Menü mit Ressourcentypen und einem Label-Selektor-Feld.

Verwenden Sie den Namensfilter, um Ressourcen auf Seiten mit langen Listen wie der Seite **Projects** schnell zu finden:

Name	Display ...	Status	Requester	Memory	CPU	Created
openshift-apiserver	No display name	Active	No requester	506.3 MiB	0.033 cores	May 25, 12:57 pm
openshift-apiserver-operator	No display name	Active	No requester	91.8 MiB	0.011 cores	May 25, 12:55 pm

Es kann nützlich sein, Pods nach Status zu filtern, um potenzielle Probleme oder problematische Bereitstellungen zu identifizieren:

Status	Ready	Restarts	Owner	Memory	CPU
Running 3	2/2	0	RS apiserver-5ff856f954	-	-
Pending 0					
Terminating 0	2/2	0	RS apiserver-5ff856f954	-	-
CrashLoopBackOff 0	2/2	0	RS apiserver-5ff856f954	-	-
Completed 0					
Failed 0					
Unknown 0					

Auf der Detailseite einer Ressource werden allgemeine, hilfreiche Informationen angezeigt. Der Inhalt dieser Seite ist je nach Ressourcentyp unterschiedlich. Auf der Seite **Pod Details** werden beispielsweise Metriken und Statusinformationen angezeigt, und auf der Seite **Secret Details** können Sie Daten, die im Secret gespeichert sind, anzeigen oder kopieren. Detail-Seiten bieten einen YAML-Editor zum Anzeigen und Ändern der Ressourcenspezifikation über die Web Console. Für einige Ressourcentypen, z. B. `secrets` und `role bindings`, sind erweiterte, auf den Ressourcentyp zugeschnittene Benutzeroberflächen vorhanden.

Erstellen von Benutzern und Gruppen

Auf der Seite **Users**, auf die Sie über **User Management → Users** zugreifen, werden Benutzer angezeigt, die sich zuvor bereits bei OpenShift angemeldet haben. Wie in *Kapitel 3, Konfigurieren von Autorisierung und Authentifizierung* beschrieben, unterstützt OpenShift verschiedene Identitätsanbieter (IdPs), einschließlich HTPasswd, LDAP und OpenID Connect.

Bei Verwendung des HTPasswd-Identitätsanbieters kann der `secrets`-Editor das Hinzufügen, Aktualisieren und Entfernen von Einträgen im HTPasswd-Secret vereinfachen. Nachdem Sie in einem Terminal einen neuen HTPasswd-Eintrag generiert oder diesen aktualisiert haben, wechseln Sie zur Web Console, um das Secret zu ändern.

Suchen Sie in der Web-Benutzeroberfläche im Projekt `openshift-config` das Secret, und klicken Sie dann auf **Actions → Edit Secret**. Das Tool **Edit Key/Value Secret** nimmt die base64-Codierung für Sie vor. Fügen Sie eine Zeile hinzu, damit sich ein neuer Benutzer bei OpenShift anmelden kann. Aktualisieren Sie eine Zeile, um das Passwort für einen Benutzer zu ändern. Löschen Sie eine Zeile, sodass sich ein Benutzer nicht bei OpenShift anmelden kann.

Auf der Seite **Groups**, auf die Sie über **User Management → Groups** zugreifen, werden vorhandene Gruppen angezeigt, und es können neue Gruppen erstellt werden.

Erstellen eines Projekts

Die Web-Benutzeroberfläche bietet eine Vielzahl von Seiten und Formularen für die Konfiguration von Projekten. So erstellen Sie ein Projekt:

1. Navigieren Sie zur Seite **Home → Projects**, um die vollständige Liste der Projekte anzuzeigen. Klicken Sie auf **Create Project**, und füllen Sie das Formular aus, um ein neues Projekt zu erstellen.

Kapitel 8 | Verwalten eines Clusters mit der Web Console

2. Nach dem Erstellen Ihres neuen Projekts können Sie auf der Seite mit den Projektdetails zur Registerkarte **Role Bindings** navigieren.
3. Red Hat empfiehlt, dass Administratoren, die für Multitenant-Cluster verantwortlich sind, **Resource Quotas** und **Limit Ranges** konfigurieren, die die Beschränkungen für das gesamte Projekt bzw. für Container erzwingen. Navigieren Sie zu **Administration → Resource Quotas** oder **Administration → Limit Ranges**, um auf den entsprechenden YAML-Editor zuzugreifen, in dem Sie diese Beschränkungen konfigurieren können.

Erläutern von Beschränkungen

Die OpenShift Web Console ist ein leistungsstarkes Tool zur grafischen Verwaltung von OpenShift-Clustern, einige administrative Aufgaben sind jedoch derzeit nicht in der Web Console verfügbar. Zum Beispiel ist für das Anzeigen von Knotenprotokollen und das Ausführen von Knoten-Debug-Sitzungen das Befehlszeilentool oc erforderlich.



Literaturhinweise

Weitere Informationen finden Sie in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Web console* unter
https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/web_console/index

► Angeleitete Übung

Durchführen der Cluster-Verwaltung

In dieser Übung führen Sie die Cluster-Verwaltung mit der Web Console durch.

Ergebnisse

Sie sollten in der Lage sein, die OpenShift Web Console für Folgendes zu verwenden:

- Suchen Sie nach Ressourcen, die einem Operator zugeordnet sind.
- Überprüfen Sie den Pod-Status, die YAML-Definition und die Protokolle.
- Anzeigen und Bearbeiten von Cluster-Konfigurationsressourcen
- Erstellen eines neuen Projekts und Konfigurieren von Ressourcenkontingenten, Beschränkungsbereichen und Role-based Access Control (RBAC)

Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt die für diese Übung benötigten Ressourcen.

```
[student@workstation ~]$ lab console-admin start
```

Anweisungen

► 1. Navigieren Sie als Benutzer `admin` zur OpenShift Web Console.

- 1.1. Melden Sie sich als Benutzer `admin` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat \
>   https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Ermitteln Sie die URL für die Web Console.

```
[student@workstation ~]$ oc whoami --show-console
https://console-openshift-console.apps.ocp4.example.com
```

- 1.3. Öffnen Sie einen Webbrower und navigieren Sie zu `https://console-openshift-console.apps.ocp4.example.com`.
- 1.4. Klicken Sie auf `localusers`, und melden Sie sich als `admin`-Benutzer mit dem Passwort `redhat` an.

Kapitel 8 | Verwalten eines Clusters mit der Web Console

- 2. Überprüfen Sie die Pod-Protokolle `openshift-console-operator` und `openshift-console`.
- 2.1. Klicken Sie in der Web-Benutzeroberfläche von Red Hat OpenShift Container Platform auf **Home** → **Projects**, um die Seite **Projects** anzuzeigen.
 - 2.2. Geben Sie `console` in das Feld **Search by name** ein, und klicken Sie dann auf den Link `openshift-console-operator`.

The screenshot shows the 'Projects' page in the Red Hat OpenShift Container Platform web console. A search bar at the top has 'console' typed into it. Below the search bar, there is a filter section with 'Name' selected and 'console' entered. A link 'Clear all filters' is also present. The main table lists two projects:

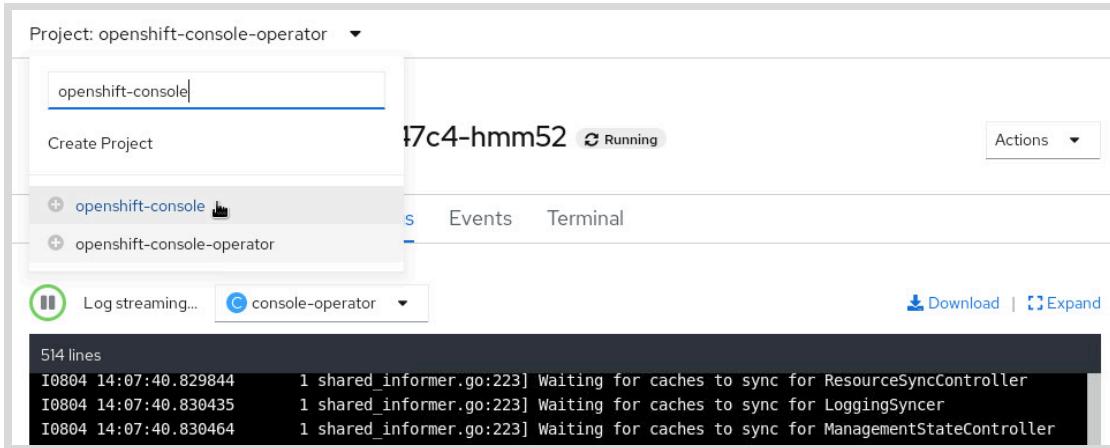
Name	Display Name	Status	Requester	Created	...
openshift-console	No display name	Active	No requester	May 25, 1:00 pm	...
openshift-console-operator	No display name	Active	No requester	May 25, 1:00 pm	...

- 2.3. Klicken Sie auf **Workloads** und dann auf **1 of 1 pods**, um zum Replikatsatz `console-operator` zu navigieren. Klicken Sie auf den Pod-Namen, der durch das Symbol P gekennzeichnet ist, um zum Pod `console-operator` zu navigieren.

The screenshot shows the 'Replica Sets' details page for the 'console-operator' replica set. The title is 'RS console-operator-6d89b76984'. The 'Pods' tab is selected. A table lists the pods:

Name	Status	Ready	Restarts	Node	...
console-operator-6d89b76984-wd5t8	Running	1/1	0	master01	...

- 2.4. Überprüfen Sie die Seite **Pod Details**, und beachten Sie die Pod-Metriken, den Ausführungsstatus und die Volumes.
- 2.5. Klicken Sie auf **YAML**, um zum Pod-Ressourcen-Editor zu navigieren.
- 2.6. Klicken Sie auf **Logs**, um die Konsolen-Operator-Protokolle anzuzeigen.
- 2.7. Öffnen Sie die Liste **Project**, und geben Sie `openshift-console` ein, um zum Projekt `openshift-console` zu wechseln.



- 2.8. Klicken Sie auf den ersten Pod in der Tabelle und dann auf **Logs**, um die Konsolen-Pod-Protokolle anzuzeigen.
- ▶ 3. Überprüfen Sie die Cluster-Einstellungen Console, Image und OAuth.
 - 3.1. Klicken Sie auf **Administration** → **Cluster Settings**, um die Seite **Cluster Settings** anzuzeigen. Die Informationen über den Update-Kanal und die aktuelle Version des Clusters sind oben aufgeführt, und ein Abschnitt für den Update-Verlauf des Clusters ist unten aufgeführt.
 - 3.2. Klicken Sie auf **Global Configuration**, um zur Liste der Cluster-Konfigurationsressourcen zu navigieren.
 - 3.3. Klicken Sie auf **Console** und dann auf **YAML**, um die Ressource **Console** zu überprüfen.
 - 3.4. Kehren Sie zur globalen Konfigurationsseite **Cluster Settings** zurück. Klicken Sie auf **Image** und dann auf **YAML**. Beachten Sie, dass der **internalRegistryHostname** für die Verwendung der internen Image-Registry konfiguriert ist.
 - 3.5. Kehren Sie zur globalen Konfigurationsseite für **Cluster Settings** zurück, und klicken Sie auf **OAuth**. Die Seite **OAuth Details** verfügt über einen speziellen Abschnitt zum Auflisten und Hinzufügen von Identitätsanbietern. Navigieren Sie zur YAML-Seite, um zusätzliche Konfigurationsdetails anzuzeigen.
- ▶ 4. Überprüfen Sie die Cluster-Rollen **admin**, **edit** und **view**.
 - 4.1. Klicken Sie im linken Menü auf **User Management** → **Roles**, um die Seite **Roles** anzuzeigen.
 - 4.2. Klicken Sie neben dem Symbol CR auf **admin**. Überprüfen Sie die Tabelle **Rules**, in der die zulässigen Aktionen für verschiedene Ressourcen beschrieben sind.

Name	Namespace	...
CR admin	All Namespaces	...
CR aggregate-olm-edit	All Namespaces	...
CR aggregate-olm-view	All Namespaces	...

- 4.3. Kehren Sie zur Seite **Cluster Roles** zurück, und klicken Sie auf die Cluster-Rolle mit dem Namen **edit**, um die Details der Cluster-Rolle **edit** anzuzeigen.
- 4.4. Kehren Sie zur Seite **Cluster Roles** zurück, und geben Sie **view** in das Feld **Search by name** ein. Klicken Sie auf die Cluster-Rolle mit dem Namen **view**, um zu den Details der Cluster-Rolle **view** zu navigieren. Beachten Sie, dass diese Rolle nur die Aktionen **get**, **list** und **watch** für die aufgelisteten Ressourcen zulässt.
5. Fügen Sie dem Secret **localusers** den Benutzereintrag **tester** hinzu.
 - 5.1. Klicken Sie in der Web-Benutzeroberfläche von OpenShift Container Platform auf **Workloads** → **Secrets**, und wählen Sie dann in der Filterliste **Project** den Eintrag **openshift-config** aus, um die Secrets für das Projekt **openshift-config** anzuzeigen.
 - 5.2. Verwenden Sie den Filter, oder scrollen Sie zum Ende der Seite, um den Link **localusers** zu suchen, und klicken Sie dann darauf.
 - 5.3. Klicken Sie auf **Actions** → **Edit Secret**, um zum Tool **Edit Key/Value Secret** zu navigieren.
 - 5.4. Generieren Sie im Terminal **workstation** einen **htpasswd**-Eintrag mit dem Passwort **redhat**.

```
[student@workstation ~]$ htpasswd -n -b tester redhat
tester:$apr1$oQ3BtW0p.HtW97.$wVbJBofBNsNd4sd
```

- 5.5. Ordnen Sie im Secret-Editor der OpenShift Web Console die Terminalausgabe des Befehls **htpasswd** dem Wert **htpasswd** zu, und klicken Sie dann auf **Save**.
- ```
admin:$apr1$Au9.fFr$0k5wvUBd3eeBt0baa77.dae
leader:$apr1$/abo4Hybn7a.tG5ZoOBn.QwefXckiy1
developer:$apr1$RjqTY4cv$xqlz3.BQfg42moSxwnTNkh.
tester:$apr1$oQ3BtW0p.HtW97.$wVbJBofBNsNd4sd
```
6. Erstellen und konfigurieren Sie ein neues Projekt mit dem Namen **console-apps**.
    - 6.1. Klicken Sie auf **Home** → **Projects**, um die Seite **Projects** anzuzeigen, und klicken Sie dann auf **Create Project**.
    - 6.2. Verwenden Sie die folgenden Informationen für das neue Projekt, und klicken Sie dann auf **Create**.

**Projektformular erstellen**

| Feld         | Wert                         |
|--------------|------------------------------|
| Name         | console-apps                 |
| Anzeigename  | Console chapter applications |
| Beschreibung | Beispielprojekt              |

- 6.3. Klicken Sie auf **Administration** → **Resource Quotas** und dann auf **Create Resource Quota**. Ändern Sie das YAML-Dokument wie folgt:

```
apiVersion: v1
kind: ResourceQuota
metadata:
 name: quota
 namespace: console-apps
spec:
 hard:
 pods: '10'
 requests.cpu: '2'
 requests.memory: 8Gi
 limits.cpu: '4'
 limits.memory: 12Gi
```

Klicken Sie auf **Create**.

- 6.4. Klicken Sie auf **Administration** → **Limit Ranges** und dann auf **Create Limit Range**. Bearbeiten Sie das YAML-Dokument, und geben Sie einen Namen für den Begrenzungsbereich an. Legen Sie die standardmäßigen Container-Beschränkungen und -Anforderungen für Arbeitsspeicher und CPU fest:

```
apiVersion: v1
kind: LimitRange
metadata:
 name: limit-range
 namespace: console-apps
spec:
 limits:
 - default:
 cpu: 500m
 memory: 5Gi
 defaultRequest:
 cpu: 10m
 memory: 100Mi
 type: Container
```

Klicken Sie auf **Create**.

- 6.5. Klicken Sie auf **User Management** → **Groups** und dann auf **Create Group**. Definieren Sie im Editor wie folgt eine Gruppenressource:

```
apiVersion: user.openshift.io/v1
kind: Group
metadata:
 name: project-team
users:
 - developer
 - tester
```

Klicken Sie auf **Create**, um die neue Gruppe `project-team` zu erstellen.

- 6.6. Klicken Sie auf **User Management** → **Role Bindings** und dann auf **Create Binding**. Füllen Sie das Formular wie folgt aus, um eine Rollenbindung für die Gruppe `project-team` zu erstellen.

#### Formular für die Team-Rollenbindung

| Feld         | Wert                                 |
|--------------|--------------------------------------|
| Binding Type | Namespace Role Binding (RoleBinding) |
| Name         | team                                 |
| Namespace    | console-apps                         |
| Role Name    | edit                                 |
| Subject      | Gruppe                               |
| Subject Name | project-team                         |

Klicken Sie auf **Create**, um die Namespace-bezogene RoleBinding zu erstellen.

- 6.7. Kehren Sie zur Seite **Role Bindings** zurück, und klicken Sie auf **Create Binding**, um eine Rollenbindung für den Benutzer `leader` zu erstellen. Füllen Sie das Formular wie folgt aus:

#### Formular für die Leader-Rollenbindung

| Feld         | Wert                                 |
|--------------|--------------------------------------|
| Binding Type | Namespace Role Binding (RoleBinding) |
| Name         | leader                               |
| Namespace    | console-apps                         |
| Role Name    | admin                                |
| Subject      | User                                 |
| Subject Name | leader                               |

Klicken Sie auf **Create**, um die Namespace-bezogene RoleBinding zu erstellen.

- 6.8. Klicken Sie auf **admin** → **Log out**, und melden Sie sich dann wieder als Benutzer **developer** mit dem Passwort **developer** an.  
Stellen Sie sicher, dass das Benutzerkonto **developer** nur auf das Projekt **console-apps** zugreifen kann.



### Anmerkung

Frühere Projekte aus angeleiteten Übungen, die nach Abschluss nicht gelöscht wurden, können auch in der Liste angezeigt werden.

- 6.9. Im nächsten Abschnitt verwenden Sie weiterhin das neue Projekt **console-apps**, sodass Sie es nicht löschen müssen.

## Beenden

Führen Sie auf dem Rechner **workstation** den Befehl **lab** aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab console-admin finish
```



### Wichtig

Löschen Sie das Projekt **console-apps** nicht. Es wird in den nächsten Abschnitten verwendet.

Hiermit ist die angeleitete Übung beendet.

# Verwalten von Workloads und Operatoren

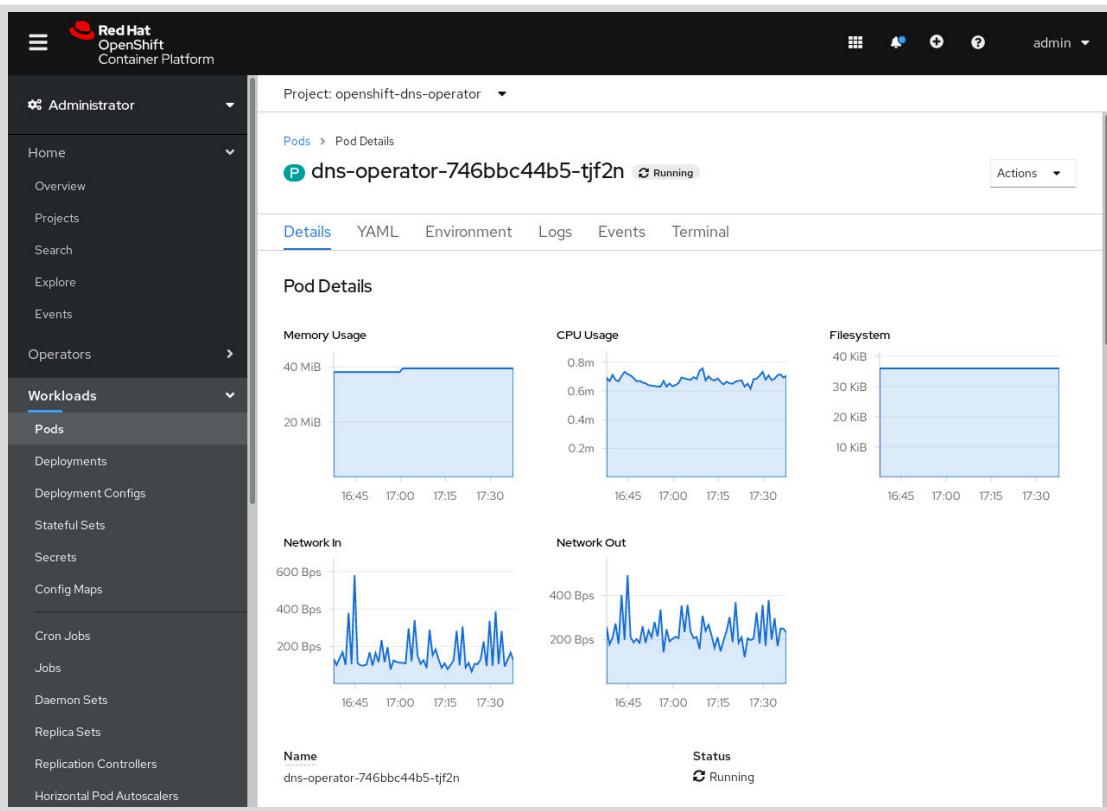
## Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, Anwendungen und Kubernetes-Operatoren mit der Web Console zu verwalten.

## Untersuchen von Workload-Ressourcen

Workload-Ressourcen wie Pods, Bereitstellungen, StatefulSets und Konfigurations-Maps werden im Menü **Workloads** aufgelistet. Klicken Sie auf einen Ressourcentyp, um eine Liste der Ressourcen anzuzeigen, und klicken Sie dann auf den Namen der Ressource, um zur Detailseite für diese Ressource zu navigieren.

Um beispielsweise zum DNS-Operator-Pod von OpenShift zu navigieren, klicken Sie auf **Workloads** → **Pods**, wählen oben auf der Seite in der Liste „Project“ `openshift-dns-operator` aus und klicken auf den Namen des Pods, der in der Tabelle aufgeführt ist:



Es gibt häufig mehrere Möglichkeiten, zu allgemeinen Ressourcen zu navigieren. In der gesamten Web-Benutzeroberfläche sind zugehörige Ressourcen häufig miteinander verknüpft. Auf der Seite „Deployment Details“ wird eine Liste der Pods angezeigt. Klicken Sie auf einen beliebigen Pod-Namen in der Liste, um die Pod-Detailseite für diesen Pod anzuzeigen.

## Verwalten von Workloads

Die Web Console bietet spezielle Editor-Seiten für viele Workload-Ressourcen. Verwenden Sie das Menü **Actions** auf der Detailseite der Ressource, um zu den spezialisierten Editor-Seiten zu navigieren:

The screenshot shows the 'Deployment Details' page for the 'openshift-apiserver-operator'. At the top, there are tabs for 'Details', 'YAML', 'Replica Sets', 'Pods', 'Environment', and 'Events'. The 'Details' tab is selected. On the left, there's a summary section with a large blue circle containing the number '1' and the word 'pod'. Below this are fields for 'Name' (set to 'openshift-apiserver-operator'), 'Namespace' (set to 'NS openshift-apiserver-operator'), 'Update Strategy' (set to 'Recreate'), and 'Progress Deadline Seconds' (set to '600 seconds'). To the right of these fields is a vertical 'Actions' dropdown menu, which is highlighted with a red box. The menu contains the following options: Edit Pod Count, Pause Rollouts, Add Health Checks, Add Storage, Edit Update Strategy, Edit Labels, Edit Annotations, Edit Deployment, and Delete Deployment.

Abbildung 8.8: Verwenden des Menüs „Actions“ zum Ändern einer Bereitstellung

Einige nützliche Aktionsseiten werden im Folgenden beschrieben:

- Alle Ressourcen verfügen über die Editoren **Edit Labels** und **Edit Annotations**.
- Klicken Sie auf **Actions** → **Add Storage**, um einer Bereitstellung eine Anforderung für ein persistentes Volume (PVC) hinzuzufügen.
- Navigieren Sie zum Bearbeiten der Replikatanzahl zur Seite **Deployment Details**, und klicken Sie auf **Actions** → **Edit Pod Count**.
- Um die Update-Strategie für eine Bereitstellung zu ändern (wie Parameter für Rolling Updates), navigieren Sie zur Seite **Deployment Details** und klicken auf **Actions** → **Edit Update Strategy**. Navigieren Sie zum Ändern der Update-Strategie für eine Bereitstellungskonfiguration zur Seite **Deployment Config Details**, und klicken Sie auf **Actions** → **Edit Deployment Config**.
- Navigieren Sie zur Seite **Secret Details**, und klicken Sie auf **Actions** → **Edit Secret**, um das Tool **Edit Key/Value Secret** anzuzeigen, das automatisch Werte mit Base64 codiert und decodiert.

Sie können auch den eingebetteten YAML-Editor verwenden, um Workload-Ressourcen zu erstellen oder zu ändern. Ziehen Sie eine JSON- oder YAML-Datei per Drag-and-Drop in den browserbasierten Editor, um die Ressource aus einer Datei zu aktualisieren, ohne den Befehl `oc` zu verwenden:

```

1 kind: Deployment
2 apiVersion: apps/v1
3 metadata:
4 annotations:
5 deployment.kubernetes.io/revision: '1'
6 exclude.release.openshift.io/internal-openshift-hosted: 'true'
7 selfLink: >-
8 /apis/apps/v1/namespaces/openshift-apiserver-operator/deployments/openshift-apiserver-operator
9 resourceVersion: '35597'
10 name: openshift-apiserver-operator
11 uid: 93b3f536-65c3-4b47-baa1-67511a26aad4
12 creationTimestamp: '2020-07-29T18:30:38Z'
13 generation: 1
14 managedFields:
15 - manager: cluster-version-operator
16 operation: Update
17 apiVersion: apps/v1
18 time: '2020-07-29T18:30:38Z'
19 fieldsType: FieldsV1
20 fieldsV1:
21 'f:metadata':
22 'f:annotations':
23 .: {}
24 'f:exclude.release.openshift.io/internal-openshift-hosted': {}
25 'f:labels':
26 .: {}
27 'f:app': {}

```

Save   Reload   Cancel   Download

**Abbildung 8.9: Bearbeiten einer Ressource mit dem eingebetteten YAML-Editor**

Neben der Möglichkeit, Ressourcen auf einer dedizierten Seite oder im eingebetteten YAML-Editor zu bearbeiten, können Sie viele andere häufige Vorgänge direkt über die OpenShift Web Console ausführen. Um beispielsweise eine Ressource zu löschen, navigieren Sie zur Detailseite der Ressource und klicken auf **Actions → Delete Resource Type**.

Es gibt häufig mehr als eine Möglichkeit, eine bestimmte Aufgabe auszuführen. Um beispielsweise eine Bereitstellung manuell zu skalieren, können Sie zur Seite **Deployment Details** navigieren und dann auf **Actions → Edit Pod Count** klicken, oder Sie können auf die Pfeile neben der Pod-Anzahl klicken, ohne die Seite zu verlassen.

## Bereitstellen von Anwendungen

Sie können Bereitstellungsressourcen auf der Seite **Workloads → Deployments** erstellen. Dieser Abschnitt enthält einen YAML-Editor mit einer vorausgefüllten Spezifikation zur Definition der YAML-Ressource.

Der Abschnitt **Builds** enthält Tools zum:

- Erstellen von Build-Konfigurationen für Source-to-Image (S2I), Dockerfile oder benutzerdefinierte Builds
- Auflisten und Überprüfen von Builds
- Verwalten von Streams

## Kapitel 8 | Verwalten eines Clusters mit der Web Console

Nachdem Sie eine Bereitstellung oder einen Build initiiert haben, verwenden Sie die Details- und Ereignis-Seiten der Ressource, um den Erfolg zu überprüfen oder um mit der Untersuchung der Ursache eines Bereitstellungsfehlers zu beginnen.

## Installieren und Verwenden von Operatoren

Untersuchen Sie Community- und Partner-Operatoren auf der Seite **Operators → OperatorHub** der OpenShift Web Console. Für die Installation über die Web-Benutzeroberfläche stehen über 360 Operatoren zur Verfügung. Darin sind auch Community-Operatoren enthalten, die von Red Hat nicht unterstützt werden.

Operatoren fügen Ihrem Cluster zusammen mit der Automatisierung Features und Services hinzu, die bisher von menschlichen Operatoren ausgeführt wurde, wie z. B. Bereitstellungskoordination oder automatische Backups. Die Operatoren umfassen eine breite Palette von Kategorien, darunter:

- Herkömmliche Datenbanken wie PostgreSQL und MySQL
- Gängige Big Data-Frameworks wie Apache Spark
- Auf Kafka basierende Streaming-Plattformen wie Red Hat AMQ-Streams
- Das serverlose Knative-Framework OpenShift Serverless Operator

Klicken Sie auf die Liste „Operator“, um Details zum Operator anzuzeigen, z. B. die Version und den Speicherort der Dokumentation. Wenn Sie zur Installation eines Operators bereit sind, klicken Sie auf **install**, um die Installation zu starten. Füllen Sie das Formular **Operator Installation** aus, um die Genehmigungsstrategie für den Ziel-Namespace und den Operator auszuwählen. Sie können Operatoren für alle oder nur bestimmte Namespaces installieren. Beachten Sie jedoch, dass nicht alle Operatoren alle Optionen für das Installationsziel unterstützen.

Nach der Installation eines Operators wird er auf der Seite **Operators → Installed Operators** angezeigt. Wenn ein Operator für einen bestimmten Namespace installiert ist, müssen Sie das richtige Projekt mit dem Projektfilter oben auf der Seite auswählen:

The screenshot shows the Red Hat OpenShift Container Platform web console interface. The left sidebar is dark-themed with white text, showing navigation links for Home, Operators (which is currently selected), OperatorHub, and Installed Operators. Under Installed Operators, there are links for Workloads, Networking, Storage, Builds, and Monitoring. The main content area has a light background. At the top, it says "Project: example-project". Below that, it says "Installed Operators > Operator Details" for the "etcd" operator. The "etcd" operator is version 0.9.4 provided by CNCF. There are tabs for Details (which is selected), YAML, Subscription, Events, All Instances, etcd Cluster, etcd Backup, and etcd Restore. Under "Details", there's a section titled "Provided APIs" with two items: "etcd Cluster" and "etcd Backup". Each item has a brief description, a "Create Instance" button, and a "Provider" section (CNCF). The "etcd Cluster" provider is listed as "Created At" a minute ago, with links to "Blog" and "Documentation". The "etcd Backup" provider is listed with "Links" and "Documentation".

Auf der Seite „Operator Details“ werden die vom Operator bereitgestellten APIs aufgeführt, sodass Sie Instanzen dieser Ressourcen erstellen können. Auf der Seite für den etcd-Operator können Sie beispielsweise Instanzen eines Etcd-Clusters, eine Backup-Anforderung oder eine Wiederherstellungsanforderung erstellen.



### Literaturhinweise

Weitere Informationen finden Sie in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Web console* unter  
[https://access.redhat.com/documentation/en-us/openshift\\_container\\_platform/4.6/html-single/web\\_console/index](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/web_console/index)

Weitere Informationen finden Sie in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Operators* unter  
[https://access.redhat.com/documentation/en-us/openshift\\_container\\_platform/4.6/html-single/operators/index](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/operators/index)

## ► Angeleitete Übung

# Verwalten von Workloads und Operatoren

In dieser Übung verwalten Sie Cluster-Workloads mit der Web Console.

### Ergebnisse

Sie sollten in der Lage sein, die OpenShift Web Console für Folgendes zu verwenden:

- Installieren eines Operators von OperatorHub
- Verwenden einer benutzerdefinierte Ressourcen zum Erstellen einer Datenbank
- Bereitstellen einer Anwendung, die die vom Operator verwalteten Ressourcen verwendet, und Beheben von Fehlern in dieser Anwendung

### Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt die für diese Aktivität benötigten Ressourcen.

```
[student@workstation ~]$ lab console-workloads start
```

### Anweisungen

- 1. Navigieren Sie als Benutzer `admin` zur OpenShift Web Console.

- 1.1. Melden Sie sich als Benutzer `admin` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Ermitteln Sie die URL für die Web Console.

```
[student@workstation ~]$ oc whoami --show-console
https://console-openshift-console.apps.ocp4.example.com
```

- 1.3. Öffnen Sie einen Webbrower und navigieren Sie zu `https://console-openshift-console.apps.ocp4.example.com`.
- 1.4. Klicken Sie auf `localusers`, und melden Sie sich als `admin`-Benutzer mit dem Passwort `redhat` an.

- 2. Überprüfen Sie die Bereitstellungen, Replikatsätze und Pods **openshift-console-operator** und **openshift-console**.
- 2.1. Klicken Sie im linken Fensterbereich auf **Workloads** → **Deployments**, und wählen Sie oben in der Liste „Project“ **all projects** aus. Geben Sie **console** in das Feld **Search by name** ein.  
Beachten Sie, dass OpenShift im Namespace **openshift-console-operator** über eine Bereitstellung mit dem Namen **console-operator** mit einem einzelnen Pod verfügt, der eine Bereitstellung mit dem Namen **console** im Namespace **openshift-console** verwaltet.

| Name             | Namespace                       | Status      | Labels                      | Pod Selector                 |
|------------------|---------------------------------|-------------|-----------------------------|------------------------------|
| console          | (NS) openshift-console          | 2 of 2 pods | app=console<br>component=ui | app=console,<br>component=ui |
| console-operator | (NS) openshift-console-operator | 1 of 1 pods | No labels                   | name=console-operator        |

- 2.2. Klicken Sie im linken Fensterbereich auf **Workloads** → **Replica Sets**, und geben Sie **console** in das Feld **Search by name** ein.  
Bereitstellungen deklarieren ein **ReplicaSet**, um sicherzustellen, dass immer eine bestimmte Anzahl von Pods ausgeführt wird.
  - 2.3. Klicken Sie in der Spalte „Status“ auf **2 of 2 pods**, um die Konsolen-Pod-Liste **ReplicaSet** anzuzeigen.
- 3. Installieren Sie den von Dev4Devs.com bereitgestellten Community-PostgreSQL-Operator über die Seite **OperatorHub**.
- 3.1. Klicken Sie im linken Fensterbereich auf **Operators** → **OperatorHub** und dann auf **Database**, um die Liste der Datenbankoperatoren anzuzeigen, die über OperatorHub verfügbar sind.
  - 3.2. Geben Sie **postgres** in das Feld **Filter by keyword** ein, und klicken Sie dann auf **PostgreSQL Operator by Dev4Ddevs.com**. Klicken Sie auf **Continue**, um die Seite „Community Operator“ anzuzeigen, und klicken Sie dann auf **Install**.

**OperatorHub**

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software through [Red Hat Marketplace](#). You can install Operators on your clusters to provide optional add-ons and shared services to your developers. After installation, the Operator capabilities will appear in the [Developer Catalog](#) providing a self-service experience.

The screenshot shows the OperatorHub interface. On the left, there's a sidebar with categories like All Items, Database, AI/Machine Learning, Application Runtime, Big Data, Cloud Provider, Database (which is selected and highlighted in blue), Developer Tools, Integration & Delivery, Logging & Tracing, Monitoring, Networking, OpenShift Optional, Security, Storage, Streaming & Messaging, Install State (with options for Installed (0) and Not Installed (4)), and Provider Type (with option for Red Hat (0)). The main area has a search bar with 'postgres' typed in. Below it, there are four items listed under 'Database': two from 'Marketplace' (Crunchy PostgreSQL for OpenShift) and two from 'Community' (PostgreSQL Operator by Dev4Ddevs.com). The item from Dev4Ddevs is highlighted with a red border.

| Category    | Name                                 | Provider                 | Description                                                                                 |
|-------------|--------------------------------------|--------------------------|---------------------------------------------------------------------------------------------|
| Marketplace | Crunchy PostgreSQL for OpenShift     | provided by Crunchy Data | Enterprise open source PostgreSQL-as-a-Service                                              |
|             | Crunchy PostgreSQL for OpenShift     | provided by Crunchy Data | Enterprise open source PostgreSQL-as-a-Service                                              |
| Community   | Crunchy PostgreSQL for OpenShift     | provided by Crunchy Data | Enterprise open source PostgreSQL-as-a-Service                                              |
|             | PostgreSQL Operator by Dev4Ddevs.com | provided by Dev4Devs.com | Operator in Go developed using the Operator Framework to package, install, configure and... |

- 3.3. Wählen Sie den Namespace `console-apps` aus, und klicken Sie dann auf **Install**, um den Operator für die Verwendung im Projekt `console-apps` zu installieren. Belassen Sie die anderen Formularfelder unverändert.
  
- ▶ 4. Melden Sie sich als Benutzer `admin` ab und als Benutzer `developer` an.
  - 4.1. Klicken Sie in der rechten oberen Ecke auf `admin` → **Log out**.
  - 4.2. Klicken Sie auf `localusers`, und melden Sie sich als Benutzer `developer` mit dem Passwort `developer` an.
  
- ▶ 5. Stellen Sie eine PostgreSQL-Datenbank unter Verwendung des installierten Operators und der benutzerdefinierten Ressourcendefinition (CRD) `Database` bereit.
  - 5.1. Klicken Sie auf der Seite `Projects` auf den Link `console-apps`, um die dem Projekt `console-apps` zugeordneten Ressourcen anzuzeigen.
  - 5.2. Klicken Sie im linken Fensterbereich auf `Operators` → `Installed Operators` und dann auf den Link `PostgreSQL Operator by Dev4Ddevs.com`, um die Seite `Operator Details` anzuzeigen.

**Anmerkung**

Wenn die Liste **Installed Operators** nicht geladen wird, überprüfen Sie, ob das Projekt **console-apps** oben auf der Seite ausgewählt ist.

| Name                                | Managed Namespaces | Status    | Provided APIs                        |
|-------------------------------------|--------------------|-----------|--------------------------------------|
| PostgreSQL Operator by Dev4Devs.com | NS console-apps    | Succeeded | Database Backup<br>Database Database |

- 5.3. Klicken Sie auf den Link **PostGreSQL Operator by Dev4Devs.com**.
- 5.4. Klicken Sie auf **Database** und dann auf **Create Database**.
- 5.5. Wechseln Sie von **Form View** zu **YAML View**, und aktualisieren Sie dann die **Database-YAML**, um das von Red Hat bereitgestellte PostgresSQL-Image anzugeben. Ändern Sie die anderen Standardwerte nicht.

```
apiVersion: postgresql.dev4devs.com/v1alpha1
kind: Database
metadata:
 name: database
 namespace: console-apps
spec:
 ...
 databaseUserKeyEnvVar: POSTGRESQL_USER
 image: registry.redhat.io/rhel8/postgresql-13:1
 size: 1
```

- 5.6. Klicken Sie auf **Create**, um die Ressource **Database** hinzuzufügen. Der PostgreSQL-Operator liest die Spezifikation und erstellt automatisch das Workload, das Netzwerk und den Storage für die neue Datenbank.
- 6. Überprüfen Sie die vom Operator erstellten Ressourcen.
- 6.1. Klicken Sie im linken Fensterbereich auf **Workloads → Deployments**, und überprüfen Sie die Liste der Bereitstellungen. Sie werden feststellen, dass eine **database**-Bereitstellung und eine **postgresql-operator**-Bereitstellung vorhanden sind.
  - 6.2. Klicken Sie auf die **database**-Bereitstellung und dann auf die Registerkarte **Pods**, um den von der **database**-Bereitstellung verteilten Pod anzuzeigen. Klicken Sie auf den Pod-Namen, um die Seite **Pod Details** anzuzeigen.

- 6.3. Klicken Sie im linken Fensterbereich auf **Networking** → **Services** und dann auf den Servicenamen **database**, um die Details des vom PostgreSQL-Operator erstellten Service anzuzeigen
- 6.4. Klicken Sie im linken Fensterbereich auf **Storage** → **Persistent Volume Claims** und dann auf die **database-PVC**, um die Details der vom PostgreSQL-Operator erstellten Anforderung für ein persistentes Volume anzuzeigen.
- 7. Erstellen Sie folgende Ressourcen für eine einfache Webanwendung: **deployment**, **service** und **route**. Die Anwendung zeigt eine Liste der Bücher an, die in der Datenbank gespeichert sind.
- 7.1. Klicken Sie im linken Fensterbereich auf **Workloads** → **Deployments** und dann auf **Create Deployment**, um den YAML-Editor der Web Console anzuzeigen. Aktualisieren Sie die YAML wie folgt, und klicken Sie dann auf **Create**.



### Anmerkung

Sie können die YAML aus der Datei `~/D0280/labs/console-workloads/deployment.yaml` auf den Rechner `workstation` kopieren.

```
kind: Deployment
apiVersion: apps/v1
metadata:
 name: books
 namespace: console-apps
spec:
 selector:
 matchLabels:
 app: books
 replicas: 1
 template:
 metadata:
 labels:
 app: books
 spec:
 containers:
 - name: books
 image: 'quay.io/redhattraining/books:v0.9'
 ports:
 - containerPort: 8080
 protocol: TCP
 readinessProbe:
 httpGet:
 path: /healthz
 port: 8080
 env:
 - name: DB_HOST
 value: database.console-apps.svc.cluster.local
 - name: DB_PORT
 value: '5432'
 - name: DB_USER
 value: postgres
 - name: DB_PASSWORD
```

```

 value: postgres
 - name: DB_NAME
 value: postgres

```

**Wichtig**

Erwarten Sie nicht, dass die Pods nach Abschluss dieses Schritts erfolgreich ausgeführt werden. Sie werden das Bereitstellungsproblem später in dieser Übung beheben.

- 7.2. Klicken Sie im linken Fensterbereich auf **Networking** → **Services** und dann auf **Create Service**, um den YAML-Editor der Web Console anzuzeigen. Aktualisieren Sie die YAML wie folgt, und klicken Sie dann auf **Create**.

**Anmerkung**

Sie können die YAML aus der Datei `~/D0280/labs/console-workloads/service.yaml` auf den Rechner `workstation` kopieren.

```

kind: Service
apiVersion: v1
metadata:
 name: books
 namespace: console-apps
spec:
 selector:
 app: books
 ports:
 - protocol: TCP
 port: 8080
 targetPort: 8080

```

- 7.3. Klicken Sie im linken Fensterbereich auf **Networking** → **Routes** und dann auf **Create Route**. Füllen Sie die Seite wie folgt aus, belassen Sie die anderen Felder unverändert, und klicken Sie dann auf **Create**.

**Erstellen des Routenformulars**

| Feld        | Wert              |
|-------------|-------------------|
| Name        | books             |
| Service     | books             |
| Target Port | 8080 → 8080 (TCP) |

- 8. Suchen und beheben Sie Bereitstellungsprobleme.

- 8.1. Klicken Sie im linken Fensterbereich auf **Home** → **Events**, und beachten Sie die Fehlerereignisse. Meldungen wie `Failed to pull image "quay.io/redhat/training/books:v0.9"` und `Error: ImagePullBackOff` zeigen ein Problem mit dem Image-Namen oder Image-Tag an.

**Events**

Resources All ▾ All Types ▾ Filter Events by name or message...

Resource A All ×

Streaming events... Showing 39 events

| Pod                    | Namespace    | Message                                                                                                                                                                                    | Timestamp         | Count                         |
|------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-------------------------------|
| books-5f7fbffdb7-ngbk5 | console-apps | Generated from kubelet on master01                                                                                                                                                         | a few seconds ago | 4 times in the last 2 minutes |
|                        |              | Pulling image "quay.io/redhattraining/books:v0.9"                                                                                                                                          |                   |                               |
| books-5f7fbffdb7-ngbk5 | console-apps | Generated from kubelet on master01                                                                                                                                                         | a few seconds ago | 4 times in the last 2 minutes |
|                        |              | Failed to pull image "quay.io/redhattraining/books:v0.9": rpc error: code = Unknown desc = Error reading manifest v0.9 in quay.io/redhattraining/books: manifest unknown: manifest unknown |                   |                               |
| books-5f7fbffdb7-ngbk5 | console-apps | Generated from kubelet on master01                                                                                                                                                         | a few seconds ago | 4 times in the last 2 minutes |
|                        |              | Error: ErrImagePull                                                                                                                                                                        |                   |                               |
| books-5f7fbffdb7-ngbk5 | console-apps | Generated from kubelet on master01                                                                                                                                                         | a few seconds ago | 6 times in the last 2 minutes |
|                        |              | Back-off pulling image "quay.io/redhattraining/books:v0.9"                                                                                                                                 |                   |                               |
| books-5f7fbffdb7-ngbk5 | console-apps | Generated from kubelet on master01                                                                                                                                                         | a few seconds ago | 6 times in the last 2 minutes |
|                        |              | Error: ImagePullBackOff                                                                                                                                                                    |                   |                               |

- 8.2. Klicken Sie im linken Fensterbereich auf **Workloads** → **Deployments** und dann auf die Bereitstellung **books**. Scrollen Sie zum Ende der Seite, um die Tabelle **Conditions** zu untersuchen. Beachten Sie, dass der Bedingungstyp **Available** den Status **False** aufweist.

| Conditions  |        |               |                            |                                                |
|-------------|--------|---------------|----------------------------|------------------------------------------------|
| Type        | Status | Updated       | Reason                     | Message                                        |
| Available   | False  | 4 minutes ago | MinimumReplicasUnavailable | Deployment does not have minimum availability. |
| Progressing | True   | 4 minutes ago | ReplicaSetUpdated          | ReplicaSet "books-695647ff54" is progressing.  |

- 8.3. Klicken Sie oben auf dem Bildschirm **Deployment Details** auf die Registerkarte **Pods**, und suchen Sie den Pod-Status. Der Status lautet **ImagePullBackOff**.
- 8.4. Klicken Sie oben auf der Seite **Deployment Details** auf die Registerkarte **YAML**, um zum YAML-Editor zu navigieren und das Problem zu beheben. Aktualisieren Sie den Image-Wert **spec** auf '**quay.io/redhattraining/books:v1.4**', und klicken Sie dann auf **Save**.



### Anmerkung

Wenn OpenShift eine Bereitstellungsressource aktualisiert, während Sie versuchen, sie zu aktualisieren, können Sie mit dem YAML-Editor Ihre Änderungen nicht speichern, ohne zuerst die neueste Version abzurufen. Klicken Sie in diesem Fall auf **Reload**, führen Sie die Bearbeitung erneut aus, und klicken Sie dann auf **Save**.

- 8.5. Klicken Sie oben auf der Seite **Deployment Details** auf die Registerkarte **Details**, und überwachen Sie die Pod-Bereitstellung. Leider kann der Pod immer noch nicht gestartet werden.
- 8.6. Klicken Sie im linken Fensterbereich auf **Home → Events**, und suchen Sie nach Hinweisen auf zusätzliche Probleme. Eine neue Ereignismeldung weist auf ein Kontingentproblem hin.

```
Error creating: pods "books-5c65dc95-z9bss" is forbidden: exceeded quota: quota, requested: limits.memory=5Gi, used: limit.memory=10752Mi, limited: limits.memory=12Gi
```

Bei der Aktualisierung der books-Bereitstellung wurde ein neuer Replikatsatz erstellt. Die Planung eines Pods aus dem neuen Replikatsatz würde jedoch die Projektquote für Speicherbeschränkungen überschreiten.

- 8.7. Um dieses Problem zu beheben, identifizieren Sie die Replikatgruppe books mit einem vorhandenen Pod, und löschen Sie sie. Durch das Löschen des Replikatsatzes mit dem fehlschlagenden Pod wird die Kontingentauslastung reduziert und die Planung des Pods aus dem neuen Replikatsatz ermöglicht. Klicken Sie im linken Fensterbereich auf **Workloads → Replica Sets**.  
Es wird erwartet, dass für die Bereitstellung books zwei Replikatsätze vorhanden sind. Der Replikatsatz books mit dem Status **1 of 1 pods** gibt die falsche Container-Image-Version an. Löschen Sie den Replikatsatz über das vertikale Ellipsenmenü für die Zeile, und wählen Sie **Delete Replica Set** aus. Bestätigen Sie den Löschvorgang, indem Sie auf **Delete**klicken.

| Name                           | Namespace    | Status      | Labels                                  | Owner               | Created       |
|--------------------------------|--------------|-------------|-----------------------------------------|---------------------|---------------|
| books-5c65dc95                 | console-apps | 0 of 1 pods | app=books<br>pod-template...=5c65d...   | books               | 2 minutes ago |
| books-5f7fbffdb7               | console-apps | 1 of 1 pods | app=books<br>pod-template...=5f7fb...   | books               | 8 minutes ago |
| database-599f5f4f8b            | console-apps | 1 of 1 pods | cr=database<br>own...=postgresqloper... | database            | 8 minutes     |
| postgresql-operator-67df97f444 | console-apps | 1 of 1 pods | na... *postgresql-oper...               | postgresql-operator | 9 minutes     |

- 8.8. Klicken Sie im linken Fensterbereich auf **Workloads → Deployments** und dann auf den Link für die Bereitstellung books. Warten Sie, bis der Ring anzeigt, dass ein Pod ausgeführt wird.

- 8.9. Klicken Sie im linken Fensterbereich auf **Networking** → **Routes** und dann auf den Link in der Spalte **Location**. Firefox öffnet eine neue Registerkarte, auf der eine Liste der aus der Datenbank abgerufenen Bücher gerendert wird.
- 8.10. Im nächsten Abschnitt verwenden Sie weiterhin das neue Projekt **console-apps** und die Bereitstellung **books**, sodass Sie sie nicht löschen müssen.

## Beenden

Führen Sie auf dem Rechner **workstation** den Befehl **lab** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab console-workloads finish
```



### Wichtig

Löschen Sie das Projekt **console-apps** oder alle Arbeiten, die Sie in diesem Abschnitt durchgeführt haben, nicht. Sie werden sie im nächsten Abschnitt verwenden.

Hiermit ist die angeleitete Übung beendet.

# Untersuchen von Cluster-Metriken

## Ziele

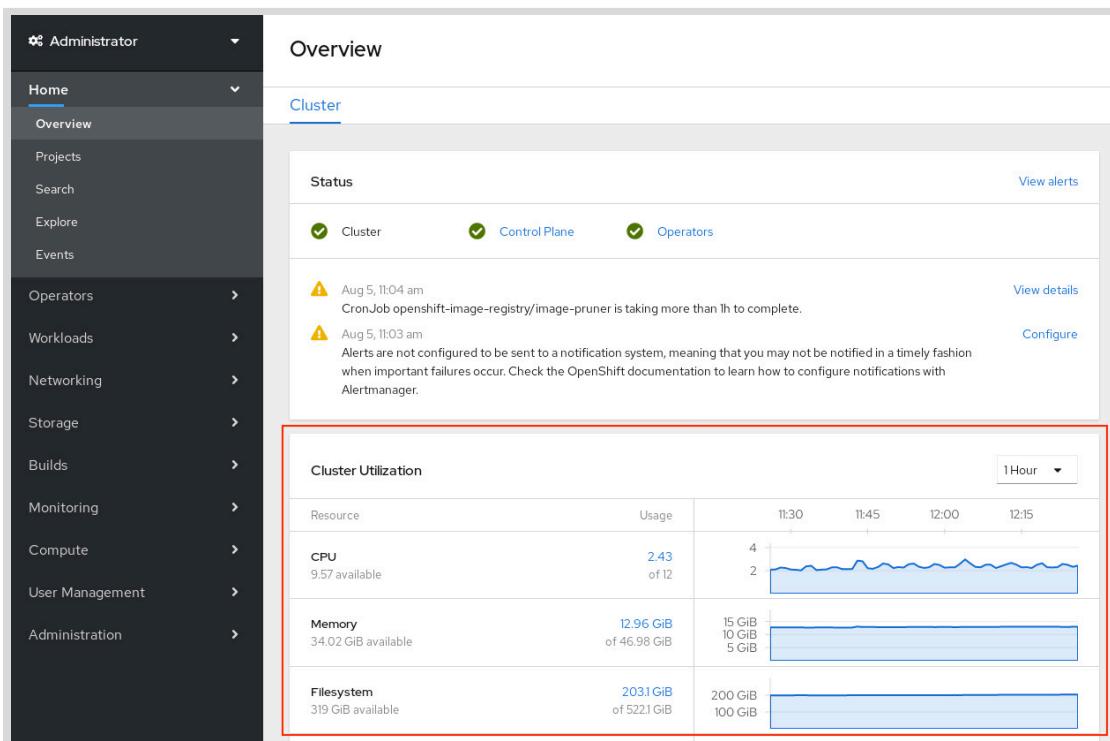
Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, die Performance- und Integritätsmetriken für Cluster-Knoten und Anwendungen zu untersuchen.

## Anzeigen von Cluster-Metriken

Die OpenShift Web Console enthält nützliche Diagramme zur Visualisierung der Cluster- und Ressourcenanalysen. Cluster-Administratoren und Benutzer mit der Cluster-Rolle `view` oder `cluster-monitoring-view` können auf die Seite `Home → Overview` zugreifen. Die Seite `Overview` zeigt verschiedene clusterweite Metriken an und bietet eine allgemeine Übersicht über den Gesamtzustand des Clusters.

Die Übersicht beinhaltet:

- Aktuelle Cluster-Kapazität auf Grundlage von CPU-, Arbeitsspeicher-, Storage- und Netzwerknutzung
- Ein Zeitreichendiagramm der Gesamtnutzung der CPU, des Arbeitsspeichers und der Disk
- Die Möglichkeit, die Top-Verbraucher von CPU, Speicher und Storage anzuzeigen.



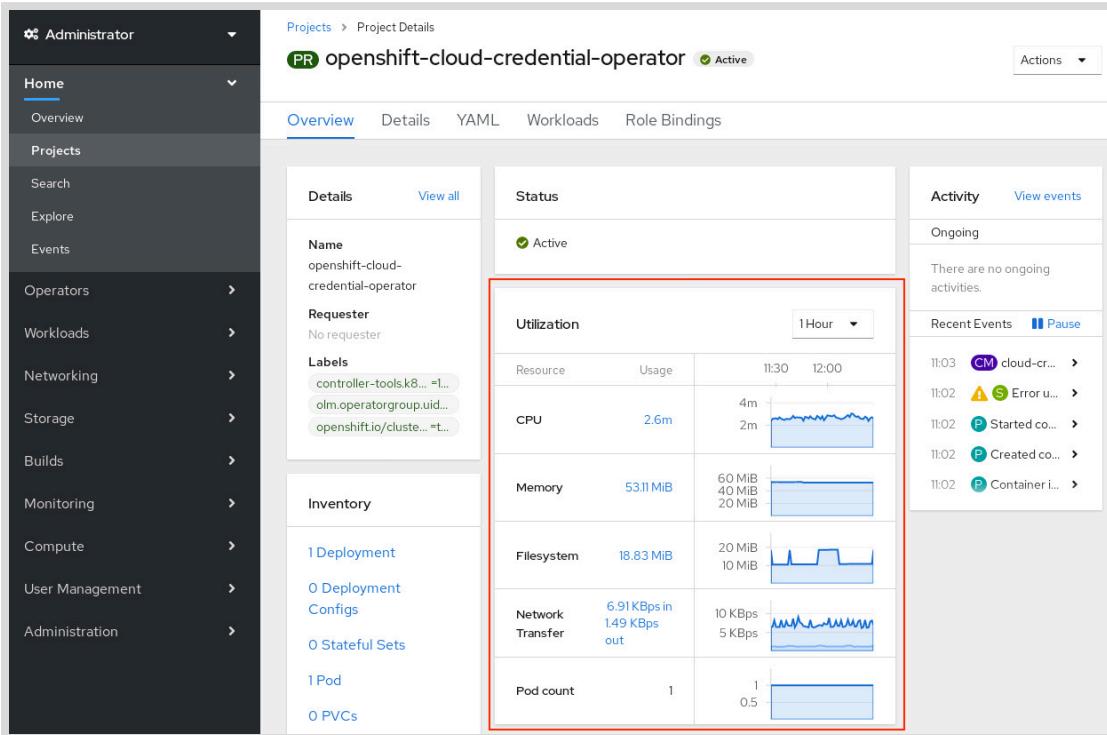
Für alle Ressourcen, die im Abschnitt zur **Cluster-Auslastung** aufgeführt sind, können Administratoren auf den Link für die aktuelle Ressourcennutzung klicken. Der Link zeigt ein Fenster mit einer Aufschlüsselung der wichtigsten Verbraucher für diese Ressource an. Die wichtigsten Verbraucher können nach Projekt, nach Pod oder nach Knoten sortiert werden. Die

## Kapitel 8 | Verwalten eines Clusters mit der Web Console

Liste der wichtigsten Verbraucher kann nützlich sein, um problematische Pods oder Knoten zu identifizieren. Beispielsweise kann ein Pod mit einem unerwartetem Arbeitsspeicherleck oben in der Liste angezeigt werden.

## Anzeigen von Projektmetriken

Auf der Seite **Project Details** werden Metriken angezeigt, die eine Übersicht über die im Rahmen eines bestimmten Projekts verwendeten Ressourcen liefern. Im Abschnitt **Utilization** werden Nutzungsinformationen zu Ressourcen wie CPU und Speicher angezeigt. Außerdem ist es möglich, die wichtigsten Verbraucher für jede Ressource anzuseigen:



Alle Metriken werden von Prometheus abgerufen. Klicken Sie auf ein beliebiges Diagramm, um zur Seite **Metrics** zu navigieren. Zeigen Sie die ausgeführte Abfrage an, und überprüfen Sie die Daten eingehender.

Wenn ein Ressourcenkontingent für das Projekt erstellt wird, werden die aktuellen Projektanforderungen und -beschränkungen auf der Seite **Project Details** angezeigt.

## Anzeigen von Ressourcenmetriken

Bei der Fehlerbehebung ist es häufig nützlich, Metriken mit einer kleineren Granularität als der gesamte Cluster oder das gesamte Projekt anzeigen. Auf der Seite **Pod Details** werden Zeitreihendiagramme der CPU-, Arbeitsspeicher- und Dateisystemnutzung für einen bestimmten Pod angezeigt. Eine plötzliche Änderung dieser wichtigen Metriken, wie z. B. eine durch eine hohe Last verursachte CPU-Spitze, wird auf dieser Seite angezeigt:

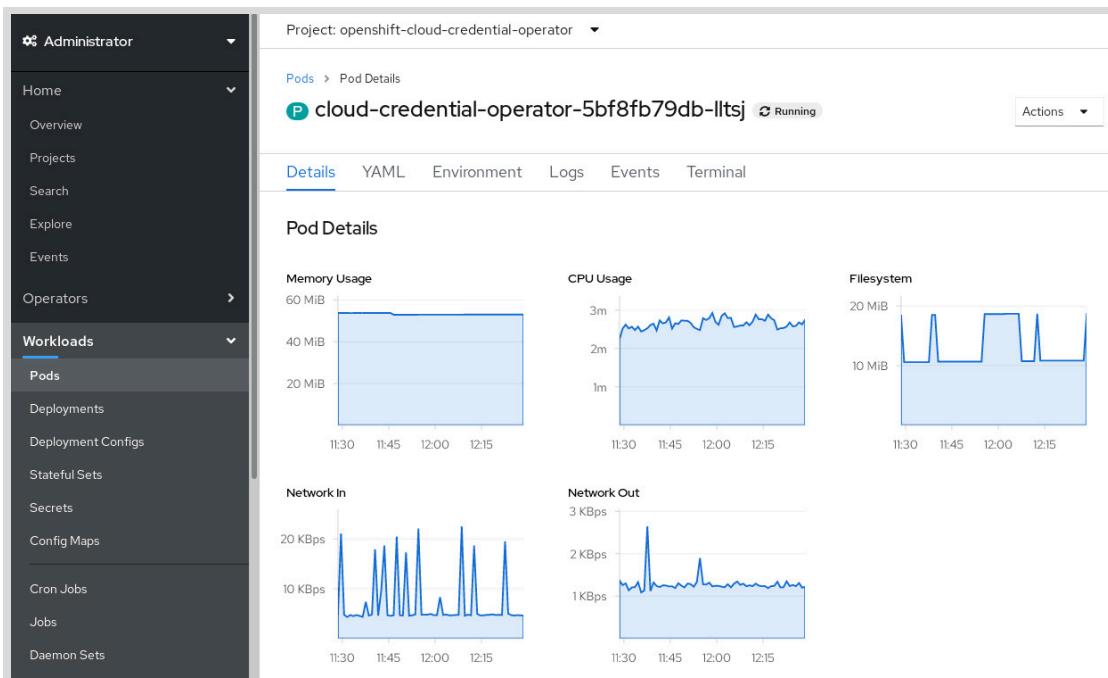
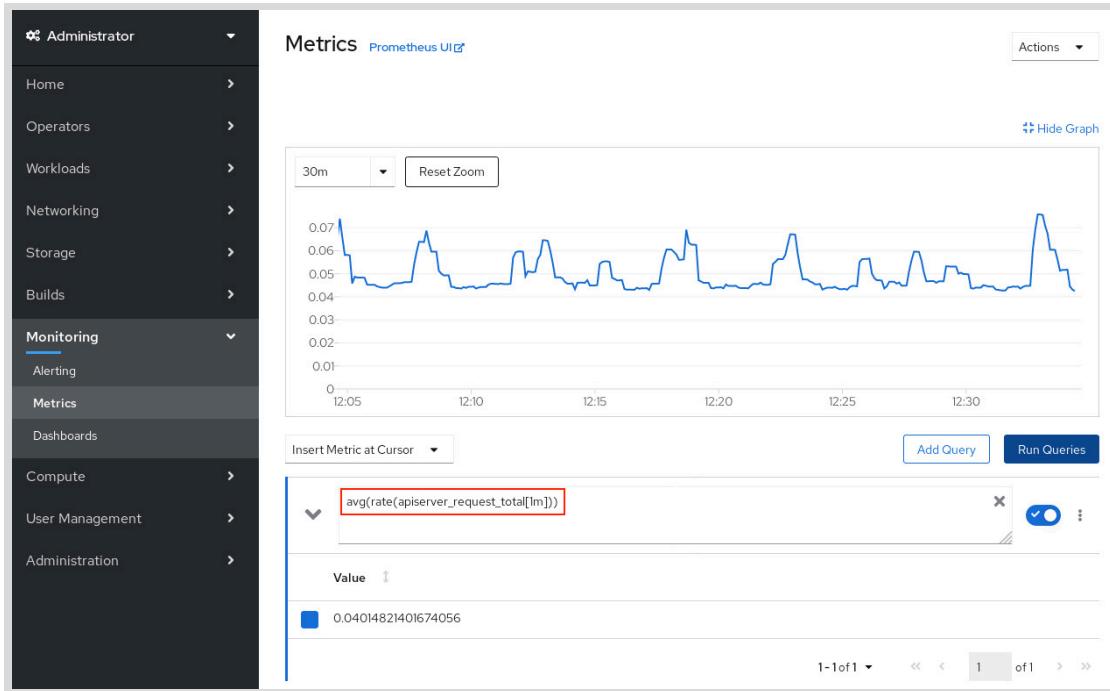


Abbildung 8.19: Zeitreihendiagramme, in denen verschiedene Metriken für einen Pod angezeigt werden

## Ausführen von Prometheus-Abfragen in der Web Console

Die Prometheus-Benutzeroberfläche ist ein funktionsreiches Tool zur Visualisierung von Metriken und zur Konfiguration von Alarmen. Die OpenShift Web Console bietet eine Schnittstelle für die Ausführung von Prometheus-Abfragen direkt aus der Web Console.

Um eine Abfrage auszuführen, navigieren Sie zu **Monitoring → Metrics**, geben in das Textfeld einen Ausdruck in der Prometheus-Abfragesprache ein und klicken auf **Run Queries**. Die Ergebnisse der Abfrage werden als Zeitreihendiagramm angezeigt:



**Abbildung 8.20:** Verwenden einer Prometheus-Abfrage zum Anzeigen eines Zeitreihendiagramms



### Anmerkung

Die Prometheus-Abfragesprache wird in diesem Kurs nicht ausführlich behandelt. In den Referenzen unten finden Sie einen Link zur offiziellen Dokumentation.



### Literaturhinweise

Weitere Informationen finden Sie in der Dokumentation zu Red Hat OpenShift Container Platform 4.6 *Monitoring* unter  
[https://access.redhat.com/documentation/en-us/openshift\\_container\\_platform/4.6/html-single/monitoring/index](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html-single/monitoring/index)

### Querying Prometheus

<https://prometheus.io/docs/prometheus/latest/querying/basics/>

## ► Angeleitete Übung

# Untersuchen von Cluster-Metriken

In dieser Übung untersuchen Sie die Metrik-Seite und das Dashboard in der Web Console.

## Ergebnisse

Sie sollten in der Lage sein, die Red Hat OpenShift Web Console für Folgendes zu verwenden:

- Anzeigen der Metriken für Cluster, Projekt, Pod und Knoten
- Identifizieren eines Pods, der große Mengen an Arbeitsspeicher oder CPU beansprucht

## Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt die für diese Übung benötigten Ressourcen.

```
[student@workstation ~]$ lab console-metrics start
```

## Anweisungen

- 1. Navigieren Sie als Benutzer `admin` zur OpenShift Web Console.

- 1.1. Melden Sie sich als Benutzer `admin` bei Ihrem OpenShift-Cluster an.

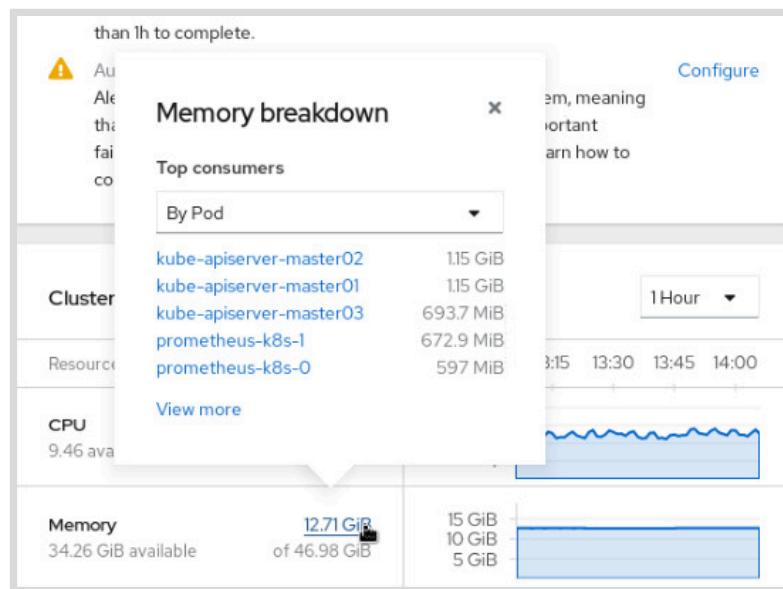
```
[student@workstation ~]$ oc login -u admin -p redhat \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Ermitteln Sie die URL für die Web Console.

```
[student@workstation ~]$ oc whoami --show-console
https://console-openshift-console.apps.ocp4.example.com
```

- 1.3. Öffnen Sie einen Webbrowser und navigieren Sie zu `https://console-openshift-console.apps.ocp4.example.com`.
- 1.4. Klicken Sie auf `localusers`, und melden Sie sich als `admin`-Benutzer mit dem Passwort `redhat` an.
- 2. In dieser angeleiteten Übung wird gezeigt, wie Laständerungen in der Web Console angezeigt werden. Beginnen Sie mit der Beobachtung der grundlegenden Integritätsmetriken auf den Seiten „Overview“, „Pod Details“ und „Project Details“.

- 2.1. Klicken Sie auf **Home** → **Overview**, um die Seite **Overview** anzuzeigen. Scrollen Sie nach unten zum Abschnitt **Cluster Utilization**, in dem ein Zeitverlaufsdiagramm der CPU, des Arbeitsspeichers und der Datenträgerverwendung des Clusters angezeigt wird.
- 2.2. Klicken Sie für jede Ressource in der Tabelle, z. B. **CPU**, **Memory** oder **Filesystem**, auf den Verwendungslink rechts, um die **Top Consumers** anzuzeigen, also die wichtigsten Verbraucher dieser Ressource. Standardmäßig filtert das Fenster die wichtigsten Verbraucher nach Projekt. Sie können aber auch nach Pod oder nach Knoten filtern.
- 2.3. Klicken Sie auf den Link für **Memory**-Nutzung, filtern Sie nach den wichtigsten Verbrauchern je Pod, und klicken Sie dann auf den Namen des Pods, der die meisten Speicherressourcen nutzt.



**Abbildung 8.21: Aufschlüsselung des Arbeitsspeichers: Wichtigste Verbraucher nach Pod**

- 2.4. Die Seite **Pod Details** zeigt Zeitverlaufsdiagramme für **Memory Usage**, **CPU Usage** und **Filesystem** oben auf der Seite an.
- 2.5. Klicken Sie auf **Home** → **Projects** und dann auf **console-apps**, um die Seite **Project Details** für **console-apps** anzuzeigen.

Beachten Sie den Abschnitt **Utilization**, in dem die Metriken für die Workloads angezeigt werden, die im Projekt **console-apps** ausgeführt werden. Die Links in der Spalte **Usage** öffnen Fenster, in denen die Pods angezeigt werden, die die meisten Ressourcen verbrauchen. Die Workloads werden sicher innerhalb der Grenzwerte ausgeführt.

- 2.6. Scrollen Sie zum Abschnitt **Resource Quotas**, in dem die aktuelle CPU- und die Speicherauslastung im Vergleich zum zugewiesenen Kontingent angezeigt werden.

► **3.** Suchen und überprüfen Sie die grundlegenden Integritätsmetriken eines Server-Knotens.

- 3.1. Klicken Sie auf **Compute** → **Nodes** und dann auf einen der Knoten in der Liste.
- 3.2. Beachten Sie auf der Seite **Utilization** die Zeitreihendiagramme, in denen die Metriken für den Knoten angezeigt werden, den Sie ausgewählt haben.

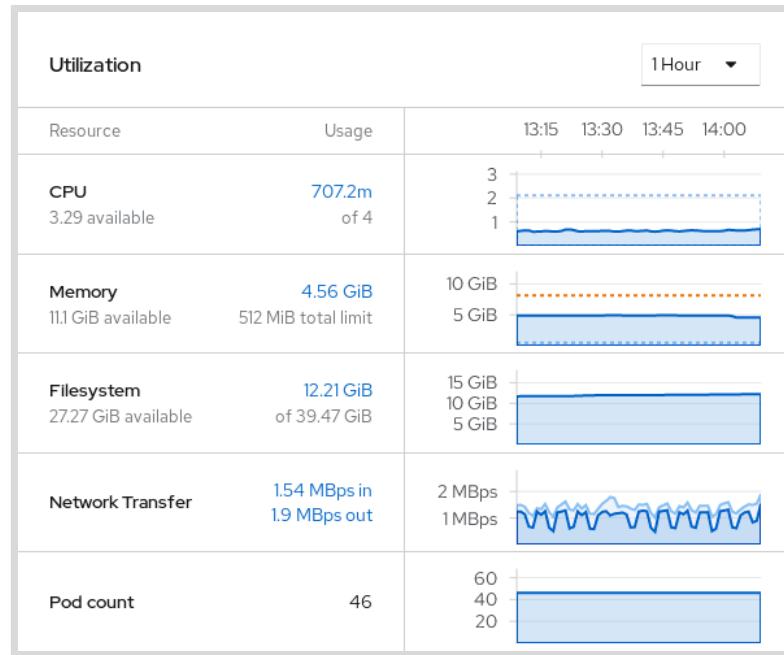


Abbildung 8.22: Zeitreihendiagramme, die verschiedene Metriken für einen Knoten anzeigen

- ▶ 4. Führen Sie auf dem Rechner `workstation` das Skript `load.sh` aus, um Last für die Beispielbereitstellung `books` zu generieren. Die Anwendung enthält absichtlich ein Arbeitsspeicherleck, das mit jeder Anforderung an den `/leak`-Pfad mehrere Megabyte RAM beansprucht.
  - 4.1. Öffnen Sie auf dem Rechner `workstation` ein Terminalfenster, und führen Sie den folgenden Befehl aus.

```
[student@workstation ~]$ ~/D0280/labs/console-metrics/load.sh
```

- ▶ 5. Beobachten Sie in der OpenShift Web Console die Änderung der Metriken, und identifizieren Sie den problematischen Pod. Die in der Web Console angezeigten Daten werden automatisch aktualisiert, sodass die Seite nicht neu geladen werden muss.
  - 5.1. Klicken Sie auf `Home → Projects` und dann auf `console-apps`, um die Seite `Project Details` für `console-apps` anzuzeigen. Sehen Sie sich das `Memory Usage`-Zeitreihendiagramm an, um Änderungen zu überwachen.  
Es kann ein oder zwei Minuten dauern, bis das Arbeitsspeicherleck signifikant genug ist, um sichtbar zu werden. Obwohl sowohl CPU als auch der Arbeitsspeicher zunehmen, bleibt die CPU-Auslastung insgesamt gering.

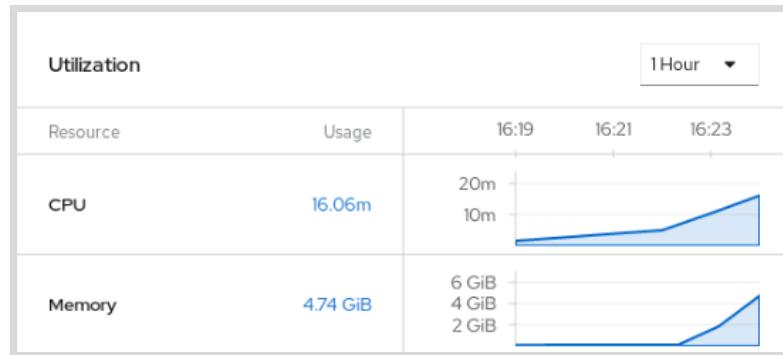


Abbildung 8.23: Nutzungsdiagramme, die auf ein mögliches Arbeitsspeicherleck hindeuten

- Klicken Sie auf Home → Overview, um die Seite Overview anzuzeigen. Der vom Lasttest verbrauchte Arbeitsspeicher ist möglicherweise zu klein, um in einem großen Cluster festgestellt zu werden, aber das Fenster **Memory breakdown** (nach Pod sortiert) bietet eine praktische Liste der Pods, die den meisten Arbeitsspeicher verbrauchen. Zeigen Sie das Fenster **Memory Breakdown** an, indem Sie auf den Nutzungslink für **Memory**klicken. Sortieren Sie die wichtigsten Verbraucher nach Pod.

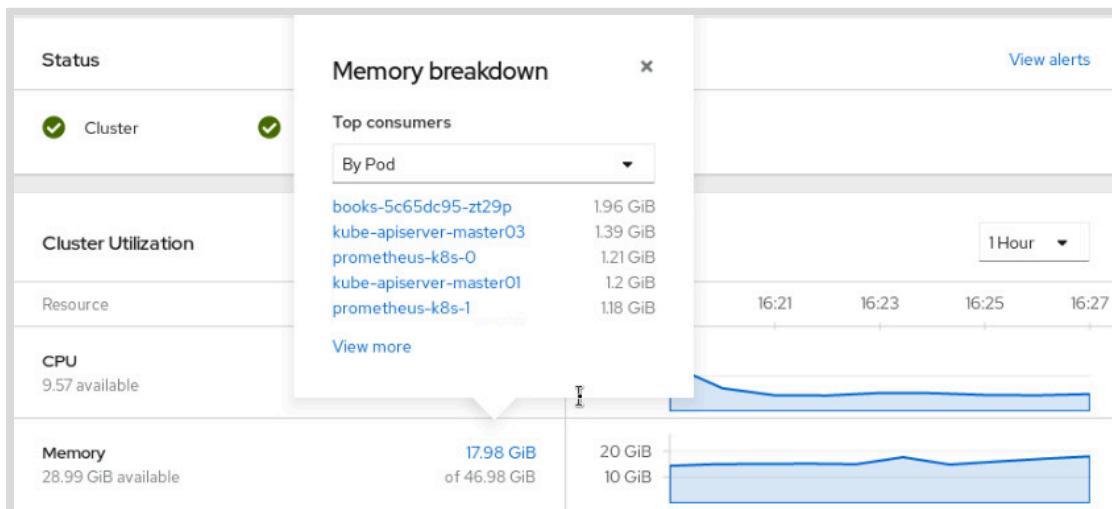
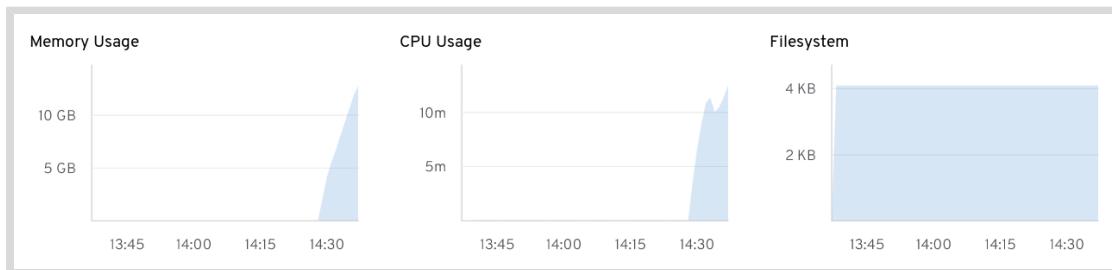


Abbildung 8.24: Der Pod „books“ ist einer der wichtigsten Verbraucher.

Der Pod books wird oben oder fast oben in der Liste angezeigt. Wenn er nicht in der Liste aufgeführt wird, müssen Sie möglicherweise etwas warten, bis das Load-Skript abgeschlossen ist.

- Klicken Sie im Fenster **Memory breakdown** auf den Pod-Link **books**, um zur Seite **Pod Details** zu navigieren. Beachten Sie das ansteigende Speicherleck, das im Zeitreihendiagramm **Memory Usage** angezeigt wird.

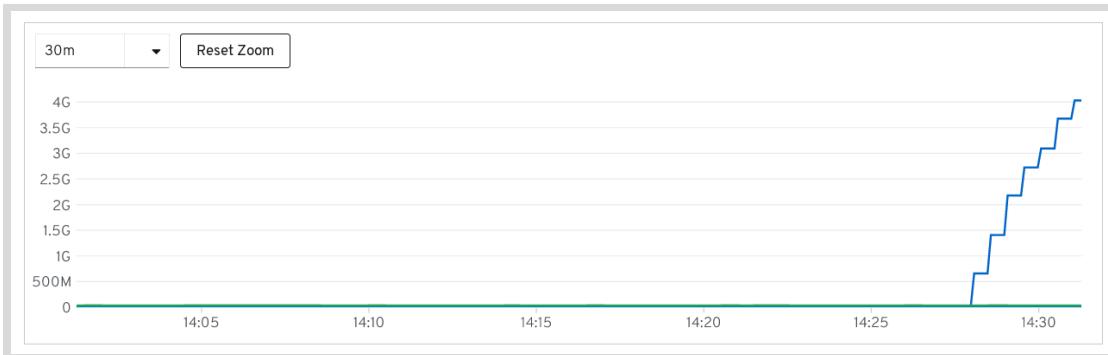


**Kapitel 8 |** Verwalten eines Clusters mit der Web Console

- 5.4. Klicken Sie auf **Monitoring → Metrics**, um die Seite **Metrics** der Web Console anzuzeigen. Geben Sie die folgende Prometheus-Abfrage in das Eingabefeld für den Ausdruck ein:

```
avg(container_memory_working_set_bytes{namespace='console-apps'}) BY (pod)
```

Klicken Sie auf **Run Queries**, um die Ergebnisse in der OpenShift Web Console anzuzeigen.



- 6. Löschen Sie das Projekt `console-apps`, und beenden Sie den Lasttest.

- 6.1. Klicken Sie auf **Home → Projects** und dann im Menü am Ende der Zeile `console-apps` auf **Delete Project**.

|                              |        |              |           |  |
|------------------------------|--------|--------------|-----------|--|
| <code>console-apps</code>    | Active | admin        | No labels |  |
| <code>default</code>         | Active | No requester | No labels |  |
| <code>kube-node-lease</code> | Active | No requester | No labels |  |
| <code>kube-public</code>     | Active | No requester | No labels |  |

- 6.2. Geben Sie `console-apps` in das Dialogfeld **Delete Project** ein, und klicken Sie dann auf **Delete**.
- 6.3. Wenn `load.sh` weiterhin auf dem Terminal „workstation“ ausgeführt wird, drücken Sie im Terminal **Strg+C**, um den Lasttest zu stoppen.

## Beenden

Führen Sie auf dem Rechner `workstation` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab console-metrics finish
```

Hiermit ist die angeleitete Übung beendet.

## ► Praktische Übung

# Verwalten eines Clusters mit der Web Console

In dieser praktischen Übung verwalten Sie den OpenShift-Cluster mit der Web Console.

### Ergebnisse

Sie sollten in der Lage sein, die OpenShift Web Console für Folgendes zu verwenden:

- Ändern eines Secret zum Hinzufügen von htpasswd-Einträgen für neue Benutzer
- Konfigurieren eines neuen Projekts mit Role-based Access Control und Ressourcenkontingenten
- Verwenden eines OperatorHub-Operators für die Bereitstellung einer Datenbank
- Erstellen einer Bereitstellung, eines Service und einer Route für eine Webanwendung
- Beheben von Fehlern in einer Anwendung anhand von Protokollen und Ereignissen

### Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt ein Verzeichnis für die Übungsdateien.

```
[student@workstation ~]$ lab console-review start
```

### Anweisungen

1. Melden Sie sich als Benutzer `admin` bei der OpenShift Web Console an.
2. Fügen Sie dem Secret `localusers` mit dem Passwort `redhat` htpasswd-Einträge für die Benutzer `dba` und `tester` hinzu.
3. Erstellen Sie die neue Gruppe `app-team`, die die Benutzer `developer` und `dba` enthält.
4. Erstellen Sie das neue Projekt `console-review` mit der Rollenbindung `view` für den Benutzer `tester` und der Rollenbindung `edit` für die Gruppe `app-team`. Legen Sie ein Ressourcenkontingent fest, das das Projekt auf zwei Pods beschränkt.
5. Installieren Sie den von Dev4Devs.com bereitgestellten Community-PostgreSQL-Operator zur Verwendung im Namespace `console-review`.
6. Erstellen Sie eine RoleBinding, die es dem Benutzer `dba` ermöglicht, Ressourcen im Projekt `openshift-operators` anzuzeigen.
7. Stellen Sie als Benutzer `dba` mit der OpenShift Web Console eine PostgreSQL-Datenbankinstanz im Projekt `console-review` bereit. Legen Sie `database` als Datenbanknamen und `registry.redhat.io/rhel8/postgresql-13:1` als Image-Namen fest.

**Kapitel 8 |** Verwalten eines Clusters mit der Web Console

- Erstellen Sie als Benutzer `developer` eine Bereitstellung, einen Service und eine Route im Projekt `console-review` mit Problemen, die Sie im nächsten Schritt beheben. Verwenden Sie das Image `quay.io/redhattraining/exoplanets:v1.0` (Replikat), und benennen Sie alle neuen Ressourcen `exoplanets`. Wenn die Anwendung `exoplanets` ordnungsgemäß konfiguriert ist, stellt sie eine Verbindung zur PostgreSQL-Datenbank her und zeigt eine Liste der Planeten an, die sich außerhalb unseres Sonnensystems befinden.

**Anmerkung**

Sie können die YAML-Ressourcen „Deployment“ und „Service“ von `~/D0280/labs/console-review` auf dem Rechner `workstation` kopieren.

Geben Sie die folgenden Umgebungsvariablen in der Bereitstellung an:

**Umgebungsvariablen für die Bereitstellung**

| Name        | Wert      |
|-------------|-----------|
| DB_HOST     | Datenbank |
| DB_PORT     | '5432'    |
| DB_USER     | postgres  |
| DB_NAME     | postgres  |
| DB_PASSWORD | postgres  |

**Wichtig**

Im nächsten Schritt beheben Sie die Probleme in der Bereitstellung.

- Suchen und beheben Sie die Bereitstellungsprobleme.
- Navigieren Sie in einem Browser zur Website „`exoplanets`“, und beobachten Sie die Ausführung der Anwendung.

**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.

```
[student@workstation ~]$ lab console-review grade
```

**Beenden**

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab console-review finish
```

Hiermit ist die praktische Übung beendet.

## ► Lösung

# Verwalten eines Clusters mit der Web Console

In dieser praktischen Übung verwalten Sie den OpenShift-Cluster mit der Web Console.

## Ergebnisse

Sie sollten in der Lage sein, die OpenShift Web Console für Folgendes zu verwenden:

- Ändern eines Secret zum Hinzufügen von htpasswd-Einträgen für neue Benutzer
- Konfigurieren eines neuen Projekts mit Role-based Access Control und Ressourcenkontingenzen
- Verwenden eines OperatorHub-Operators für die Bereitstellung einer Datenbank
- Erstellen einer Bereitstellung, eines Service und einer Route für eine Webanwendung
- Beheben von Fehlern in einer Anwendung anhand von Protokollen und Ereignissen

## Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Dieser Befehl stellt sicher, dass die Cluster-API erreichbar ist, und erstellt ein Verzeichnis für die Übungsdateien.

```
[student@workstation ~]$ lab console-review start
```

## Anweisungen

1. Melden Sie sich als Benutzer `admin` bei der OpenShift Web Console an.
  - 1.1. Melden Sie sich als Benutzer `admin` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Ermitteln Sie die URL für die Web Console.

```
[student@workstation ~]$ oc whoami --show-console
https://console-openshift-console.apps.ocp4.example.com
```

- 1.3. Öffnen Sie einen Webbrowser und navigieren Sie zu `https://console-openshift-console.apps.ocp4.example.com`.

**Kapitel 8 |** Verwalten eines Clusters mit der Web Console

- 1.4. Klicken Sie auf **localusers**, und melden Sie sich als **admin**-Benutzer mit dem Passwort **redhat** an.
2. Fügen Sie dem Secret **localusers** mit dem Passwort **redhat** htpasswd-Einträge für die Benutzer **dba** und **tester** hinzu.
  - 2.1. Klicken Sie in der Web-Benutzeroberfläche von Red Hat OpenShift Container Platform auf **Workloads** → **Secrets**, und wählen Sie dann in der Filterliste **Project** den Eintrag **openshift-config** aus, um die Secrets für das Projekt **openshift-config** anzuzeigen.
  - 2.2. Scrollen Sie an das Ende der Seite, und klicken Sie auf den Link **localusers**, um die **Secret Details** für **localusers** anzuzeigen.
  - 2.3. Klicken Sie oben auf der Seite auf **Actions** → **Edit Secret**, um zum Tool **Edit Key/Value Secret** zu navigieren.
  - 2.4. Verwenden Sie ein Terminal auf dem **workstation**-Rechner einen verschlüsselten htpasswd-Eintrag für beide Benutzer.

```
[student@workstation ~]$ htpasswd -n -b dba redhat
dba:$apr1$YF4ack.9$qho0THlWTC.cLByNEHDaV
[student@workstation ~]$ htpasswd -n -b tester redhat
tester:$apr1$XdTSqET7$i0hkC5bIs7PhYUm2KhiI.0
```

- 2.5. Ordnen Sie im Secret-Editor der OpenShift Web Console die Terminalausgabe der Befehle **htpasswd** dem Wert **htpasswd** zu, und klicken Sie dann auf **Save**.

```
admin:$apr1$Au9.fFr$0k5wvUBd3eeBt0baa77.dae
leader:$apr1$/abo4Hybn7a.tG5Zo0Bn.QwefXckiy1
developer:$apr1$RjqTY4cv$xql3.BQfg42moSxwnTNkh.
dba:$apr1$YF4ack.9$qho0THlWTC.cLByNEHDaV
tester:$apr1$XdTSqET7$i0hkC5bIs7PhYUm2KhiI.0
```

3. Erstellen Sie die neue Gruppe **app-team**, die die Benutzer **developer** und **dba** enthält.
  - 3.1. Klicken Sie auf **User Management** → **Groups** und dann auf **Create Group**. Verwenden Sie den YAML-Editor, um eine Gruppenressource wie folgt zu definieren:

```
apiVersion: user.openshift.io/v1
kind: Group
metadata:
 name: app-team
users:
 - developer
 - dba
```

Klicken Sie auf **Create**, um die neue Gruppe **app-team** hinzuzufügen.

4. Erstellen Sie das neue Projekt **console-review** mit der Rollenbindung **view** für den Benutzer **tester** und der Rollenbindung **edit** für die Gruppe **app-team**. Legen Sie ein Ressourcenkontingent fest, das das Projekt auf zwei Pods beschränkt.
  - 4.1. Click **Home** → **Projects** to view the Projects page, and then click **Create Project**. Geben Sie **console-review** in das Feld **Name** ein, und geben Sie dann einen optionalen **Display Name** und eine **Description** ein. Klicken Sie auf **Create**.

- 4.2. Klicken Sie auf **User Management** → **Role Bindings** und dann auf **Create Binding**. Füllen Sie das Formular wie folgt aus, um eine Namespace-bezogene Rollenbindung für die Gruppe `app-team` zu erstellen.

#### Formular für die App-Team-Rollenbindung

| Feld         | Wert                                 |
|--------------|--------------------------------------|
| Binding Type | Namespace Role Binding (RoleBinding) |
| Name         | app-team                             |
| Namespace    | console-review                       |
| Role Name    | edit                                 |
| Subject      | Gruppe                               |
| Subject Name | app-team                             |

Klicken Sie auf **Create**, um die Namespace-bezogene RoleBinding zu erstellen.

- 4.3. Klicken Sie auf den Link **Role Bindings**, um zur Seite **Role Bindings** zurückzukehren, und klicken Sie dann auf **Create Binding**. Füllen Sie das Formular wie folgt aus, um eine Namespace-bezogene Rollenbindung für den Benutzer `tester` zu erstellen.

#### Formular für die Tester-Rollenbindung

| Feld         | Wert                                 |
|--------------|--------------------------------------|
| Binding Type | Namespace Role Binding (RoleBinding) |
| Name         | tester                               |
| Namespace    | console-review                       |
| Role Name    | view                                 |
| Subject      | User                                 |
| Subject Name | tester                               |

Klicken Sie auf **Create**, um die Namespace-bezogene RoleBinding zu erstellen.

- 4.4. Klicken Sie auf **Administration** → **Resource Quotas** und dann auf **Create Resource Quota**. Ändern Sie das YAML-Dokument wie folgt, um eine Beschränkung auf vier Pods festzulegen:

**Kapitel 8 |** Verwalten eines Clusters mit der Web Console

```
apiVersion: v1
kind: ResourceQuota
metadata:
 name: quota
 namespace: console-review
spec:
 hard:
 pods: '2'
```

Entfernen Sie die CPU- und Arbeitsspeicheranforderungen und -beschränkungen, und klicken Sie dann auf **Create**.

5. Installieren Sie den von Dev4Devs.com bereitgestellten Community-PostgreSQL-Operator zur Verwendung im Namespace `console-review`.
  - 5.1. Klicken Sie auf **Operators** → **OperatorHub** und dann auf **Database**, um die Liste der Datenbankoperatoren anzuzeigen, die über OperatorHub verfügbar sind.
  - 5.2. Geben Sie `postgres` in das Feld **Filter by keyword** ein, und klicken Sie dann auf **PostgreSQL Operator by Dev4Ddevs.com**. Klicken Sie auf **Continue**, um die Seite „Community Operator“ anzuzeigen, und klicken Sie dann auf **Install**.

The screenshot shows the OperatorHub interface with the following details:

- Header:** OperatorHub
- Search Bar:** Database, Filter by keyword: postgres
- Sidebar:**
  - All Items
  - AI/Machine Learning
  - Application Runtime
  - Big Data
  - Cloud Provider
  - Database** (selected)
  - Developer Tools
  - Integration & Delivery
  - Logging & Tracing
  - Monitoring
  - Networking
  - OpenShift Optional
  - Security
  - Storage
  - Streaming & Messaging
- Filter Options:**
  - Install State:  Installed (0),  Not Installed (4)
  - Provider Type:  Red Hat (0)
- Results:**
  - Marketplace:** Crunchy PostgreSQL for OpenShift (provided by Crunchy Data) - Enterprise open source PostgreSQL-as-a-Service
  - Community:** PostgreSQL Operator by Dev4Ddevs.com (provided by Dev4Devs.com) - Operator in Go developed using the Operator Framework to package, install, configure and...
- Total Items:** 4 items

- 5.3. Wählen Sie den Namespace `console-review` aus, und klicken Sie dann auf **Install**, um den Operator für die Verwendung im Projekt `console-review` zu installieren. Belassen Sie die anderen Formularfelder unverändert, und klicken Sie auf **Install**.
6. Erstellen Sie eine RoleBinding, die es dem Benutzer `dba` ermöglicht, Ressourcen im Projekt `openshift-operators` anzuzeigen.
  - 6.1. Klicken Sie auf **User Management** → **Role Bindings** und dann auf **Create Binding**. Füllen Sie das Formular wie folgt aus.

#### Formular für die Rollenbindung von DBA-OpenShift-Operators

| Feld         | Wert                                 |
|--------------|--------------------------------------|
| Binding Type | Namespace Role Binding (RoleBinding) |
| Name         | <code>dba</code>                     |
| Namespace    | <code>openshift-operators</code>     |
| Role Name    | <code>view</code>                    |
| Subject      | User                                 |
| Subject Name | <code>dba</code>                     |

Klicken Sie auf **Create**, um die Namespace-bezogene RoleBinding hinzuzufügen.

7. Stellen Sie als Benutzer `dba` mit der OpenShift Web Console eine PostgreSQL-Datenbankinstanz im Projekt `console-review` bereit. Legen Sie `database` als Datenbanknamen und `registry.redhat.io/rhel8/postgresql-13:1` als Image-Namen fest.
  - 7.1. Klicken Sie auf **admin** → **Log out**, und melden Sie sich dann als `dba`-Benutzer mit dem Passwort `redhat` an.
  - 7.2. Klicken Sie auf **Home** → **Projects** und dann auf den Projekt-Link `console-review`, um zum Projekt `console-review` zu wechseln.
  - 7.3. Klicken Sie auf **Operators** → **Installed Operators** und dann auf den Link `PostgreSQL Operator by Dev4Ddevs.com`.



#### Anmerkung

Wenn die Liste **Installed Operators** nicht geladen wird, überprüfen Sie, ob das Projekt `console-review` oben auf der Seite ausgewählt ist.

- 7.4. Klicken Sie auf **Database** und dann auf **Create Database**.

**Kapitel 8 |** Verwalten eines Clusters mit der Web Console

The screenshot shows the 'Operator Details' page for the 'PostgreSQL Operator by Dev4Ddevs.com'. The 'Database' tab is active. A blue button labeled 'Create Database' is highlighted with a red box. Below it, a message says 'No Operands Found' and 'Operands are declarative components used to define the behavior of the application.'

- 7.5. Wechseln Sie von **Form View** zu **YAML View**, und aktualisieren Sie dann die **Database-YAML**, um das PostgreSQL-Image anzugeben. Ändern Sie die anderen Standardwerte nicht.

```
apiVersion: postgresql.dev4devs.com/v1alpha1
kind: Database
 name: database
 ...
 databaseUserKeyEnvVar: POSTGRESQL_USER
 image: registry.redhat.io/rhel8/postgresql-13:1
 size: 1
```

- 7.6. Klicken Sie auf **Create**, um die Ressource **Database** hinzuzufügen. Der PostgreSQL-Operator liest die Spezifikation und erstellt automatisch das Workload, das Netzwerk und den Storage für die neue Datenbank.
8. Erstellen Sie als Benutzer **developer** eine Bereitstellung, einen Service und eine Route im Projekt **console-review** mit Problemen, die Sie im nächsten Schritt beheben. Verwenden Sie das Image `quay.io/redhattraining/exoplanets:v1.0` (Replikat), und benennen Sie alle neuen Ressourcen **exoplanets**. Wenn die Anwendung **exoplanets** ordnungsgemäß konfiguriert ist, stellt sie eine Verbindung zur PostgreSQL-Datenbank her und zeigt eine Liste der Planeten an, die sich außerhalb unseres Sonnensystems befinden.

**Anmerkung**

Sie können die YAML-Ressourcen „Deployment“ und „Service“ von `~/DO280/labs/console-review` auf dem Rechner `workstation` kopieren.

Geben Sie die folgenden Umgebungsvariablen in der Bereitstellung an:

## Umgebungsvariablen für die Bereitstellung

| Name        | Wert      |
|-------------|-----------|
| DB_HOST     | Datenbank |
| DB_PORT     | '5432'    |
| DB_USER     | postgres  |
| DB_NAME     | postgres  |
| DB_PASSWORD | postgres  |



### Wichtig

Im nächsten Schritt beheben Sie die Probleme in der Bereitstellung.

- 8.1. Klicken Sie auf **dba** → **Log out**, und melden Sie sich dann als Benutzer **developer** mit dem Passwort **developer** an.
- 8.2. Klicken Sie auf **Home** → **Projects** und dann auf das Projekt **console-review**, um zum Projekt **console-review** zu wechseln.
- 8.3. Klicken Sie auf **Workloads** → **Deployments** und dann auf **Create Deployment**, um den YAML-Editor der Web Console anzuzeigen. Aktualisieren Sie die YAML wie folgt, und klicken Sie dann auf **Create**:

```
kind: Deployment
apiVersion: apps/v1
metadata:
 name: exoplanets
 namespace: console-review
spec:
 selector:
 matchLabels:
 app: exoplanets
 replicas: 1
 template:
 metadata:
 labels:
 app: exoplanets
 spec:
 containers:
 - name: exoplanets
 image: 'quay.io/redhattraining/exoplanets:v1.0'
 ports:
 - containerPort: 8080
 protocol: TCP
 readinessProbe:
 httpGet:
 path: /healthz
 port: 8080
```

```
env:
 - name: DB_HOST
 value: database
 - name: DB_PORT
 value: '5432'
 - name: DB_USER
 value: postgres
 - name: DB_NAME
 value: postgres
 - name: DB_PASSWORD
 value: postgres
```

- 8.4. Klicken Sie auf **Networking → Services** und dann auf **Create Service**, um den YAML-Editor der Web Console anzuzeigen. Aktualisieren Sie die YAML wie folgt, und klicken Sie dann auf **Create**:

```
kind: Service
apiVersion: v1
metadata:
 name: exoplanets
 namespace: console-review
spec:
 selector:
 app: exoplanets
 ports:
 - protocol: TCP
 port: 8080
 targetPort: 8080
```

- 8.5. Klicken Sie auf **Networking → Routes** und dann auf **Create Route**. Füllen Sie das Formular wie folgt aus, belassen Sie die anderen Felder unverändert, und klicken Sie dann auf **Create**.

### Erstellen des Routenformulars

| Feld        | Wert              |
|-------------|-------------------|
| Name        | exoplanets        |
| Service     | exoplanets        |
| Target Port | 8080 → 8080 (TCP) |

9. Suchen und beheben Sie die Bereitstellungsprobleme.

- 9.1. Klicken Sie auf **developer → Log out**, und melden Sie sich dann als **admin**-Benutzer mit dem Passwort **redhat** an.
- 9.2. Klicken Sie auf **Home → Events**, und wählen Sie dann oben im Projektlisten-Filter **console-review** aus. Für das Kontingent „exoplanets“ wird ein Fehler angezeigt:

```
(combined from similar events): Error creating: pods "exoplanets-5f88574546-lsnmx"
is forbidden: exceeded quota: quota, requested: pods=1, used: pods=2, limited:
pods=2
```

- 9.3. Klicken Sie auf **Administration** → **Resource Quotas**, und wählen Sie dann **console-review** in der Filterliste **Project** aus.
- 9.4. Klicken Sie in der Liste mit dem Ressourcenkontingenten auf den Link **quota** und dann auf die Registerkarte **YAML**. Ändern Sie die **spec** wie folgt, um eine Beschränkung von vier Pods festzulegen, und klicken Sie dann auf **Save**.

```
kind: ResourceQuota
apiVersion: v1
metadata:
 name: quota
 namespace: console-review
 ...output omitted...
spec:
 hard:
 pods: '4'
 ...output omitted...
```



### Anmerkung

Für das Projekt sind ein Pod für das angegebene Replikat von „exoplanet“ und ein zusätzlicher Pod erforderlich, um eine Änderung zu implementieren.

- 9.5. Klicken Sie auf **Workloads** → **Pods**, und überprüfen Sie die Pod-Liste. Es kann ein bis zwei Minuten dauern, bis der Pod **exoplanets** in der Liste angezeigt wird.
10. Navigieren Sie in einem Browser zur Website „exoplanets“, und beobachten Sie die Ausführung der Anwendung.
  - 10.1. Klicken Sie auf **Networking** → **Routes**, dann auf den Routennamen **exoplanets** und schließlich auf den Link in der Spalte **Location**. Firefox öffnet eine neue Registerkarte, auf der eine Tabelle mit Exoplaneten angezeigt wird.

## 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.

```
[student@workstation ~]$ lab console-review grade
```

## Beenden

Führen Sie auf dem Rechner **workstation** als Benutzer **student** den Befehl **lab** aus, um diese Übung zu beenden. Dies ist wichtig, um sicherzustellen, dass Ressourcen aus vorherigen Übungen sich nicht auf zukünftige Übungen auswirken.

```
[student@workstation ~]$ lab console-review finish
```

Hiermit ist die praktische Übung beendet.

# Zusammenfassung

---

In diesem Kapitel wurden die folgenden Themen behandelt:

- Die OpenShift Web Console bietet eine grafische Benutzeroberfläche zum Anzeigen und Verwalten von OpenShift-Ressourcen.
- Einige Ressourcen verfügen über eine spezielle Seite, auf der Ressourcen bequemer erstellt und bearbeitet werden können als mit YAML, wie z. B. den Editor **Edit Key/Value Secret**, der die Base64-Kodierung und -Dekodierung automatisch vornimmt.
- Sie können Operatoren von Partnern und der Community über die eingebettete Seite **OperatorHub** installieren.
- Auf der Seite **Dashboards** werden clusterweite Metriken wie CPU-, Arbeitsspeicher- und Speichernutzung angezeigt.
- Auf **Project Details**-Seiten werden für das Projekt spezifische Metriken angezeigt, z. B. die Top-Ten-Verbraucher von Arbeitsspeicher nach Pod und die aktuelle Nutzung des Ressourcenkontingents.

## Kapitel 9

# Ausführliche Wiederholung

### Ziel

Wiederholen von Aufgaben aus *Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster*

### Ziele

- Wiederholen von Aufgaben aus *Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster*

### Abschnitte

- Ausführliche Wiederholung

### Praktische Übungen

- Fehlerbehebung in OpenShift-Clustern und -Anwendungen
- Konfigurieren einer Projektvorlage mit Ressourcen- und Netzwerkbeschränkungen

# Ausführliche Wiederholung

## Ziele

Nach Abschluss dieses Abschnitts sollten Sie die in *Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster* erworbenen Kenntnisse demonstrieren können.

## Wiederholung Red Hat OpenShift Administration II: Operating a Production Kubernetes Cluster

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 von Red Hat OpenShift Container Platform**

Beschreiben der Features und Architektur von OpenShift Container Platform

- Beschreiben der typischen Nutzung des Produkts und seiner Funktionen
- Beschreiben der Architektur der Red Hat OpenShift Container Platform
- Beschreiben von Cluster-Operatoren und deren Funktionsweise sowie Benennen der wichtigsten Cluster-Operatoren

### **Kapitel 2, Überprüfen der Integrität eines Clusters**

Beschreiben der OpenShift-Installationsmethoden und Überprüfen der Integrität eines neu installierten Clusters

- Beschreiben des OpenShift-Installationsvorgangs, der Full-Stack-Automatisierung und sonstiger vorhandener Infrastruktur-Installationsmethode
- Ausführen von Befehlen, die bei der Fehlerbehebung helfen, Überprüfen, ob die OpenShift-Knoten fehlerfrei sind, und Beheben häufiger Probleme bei OpenShift- und Kubernetes-Bereitstellungen.
- Identifizieren der Komponenten und Ressourcen des persistenten Storage und Bereitstellen einer Anwendung mit persistenter Volume-Anforderung

### **Kapitel 3, Konfigurieren von Autorisierung und Authentifizierung**

Konfigurieren der Authentifizierung mit dem HTPasswd-Identitätsanbieter und Zuweisen von Rollen zu Benutzern und Gruppen

- Konfigurieren des HTPasswd-Identitätsanbieters für die OpenShift-Authentifizierung
- Rollenbasierte Zugriffskontrollen definieren und Berechtigungen für Benutzer anwenden.

## **Kapitel 4, Konfigurieren der Anwendungssicherheit**

Beschränken der Berechtigungen von Anwendungen mithilfe von Sicherheitskontextbeschränkungen und Schützen der Zugangsdaten mit Secrets

- Erstellen und Anwenden von Geheimnissen zum Verwalten von vertraulichen Informationen und Teilen von Geheimnissen zwischen Anwendungen
- Erstellen von Servicekonten und Anwenden von Berechtigungen sowie Verwalten von Sicherheitskontextbeschränkungen

## **Kapitel 5, Konfigurieren des OpenShift-Netzwerks für Anwendungen**

Beheben von Fehlern in OpenShift Software-defined Networking (SDN) und Konfigurieren von Netzwerkrichtlinien

- Beheben von Fehlern des SDN von OpenShift über die Befehlszeilenschnittstelle
- Zulassen und Absichern von Netzwerkverbindungen zu Anwendungen in einem OpenShift-Cluster
- Beschränken des Netzwerkdatenverkehrs zwischen Projekten und Pods

## **Kapitel 6, Steuern der Pod-Zuordnung (Scheduling)**

Steuern der Knoten, auf denen ein Pod ausgeführt wird

- Beschreiben der Algorithmen für die Pod-Zuordnung, der Methoden zur Steuerung der Zuordnung und Anwenden dieser Methoden
- Einschränken der Ressourcen, die von Containern, Pods und Projekten beansprucht werden
- Steuern der Anzahl von Replikaten eines Pods, Angeben der Anzahl von Replikaten in einer Bereitstellung, manuelles Skalieren der Anzahl der Replikate und Erstellen einer Horizontal Pod Autoscaler (HPA)-Ressource.

## **Kapitel 7, Beschreiben von Cluster-Updates**

Beschreiben der Durchführung eines Cluster-Updates

Beschreiben des Prozesses für Cluster-Updates

## **Kapitel 8, Verwalten eines Clusters mit der Web Console**

Verwalten eines Red Hat OpenShift-Clusters mit der Web Console

- Durchführen der Cluster-Verwaltung mit der Web Console
- Verwalten von Anwendungen und Kubernetes-Operatoren mit der Web Console
- Untersuchen von Performance- und Integritätsmetriken für Cluster-Knoten und Anwendungen

## ► Praktische Übung

# Fehlerbehebung in OpenShift-Clustern und -Anwendungen

In dieser Übung erlauben Sie Entwicklern den Zugriff auf einen Cluster und die Fehlerbehebung bei Anwendungsbereitstellungen.

### Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen Sie ein neues Projekt
- Führen Sie einen Smoke-Test des OpenShift-Clusters durch, indem Sie eine Anwendung mithilfe des Source-to-Image-Prozesses erstellen.
- Erstellen Sie Anwendungen mithilfe der deployment-Ressource.
- Verwalten von Benutzern mit dem HTPasswd-Identitätsanbieter
- Erstellen und Verwalten von Gruppen.
- Verwalten von RBAC und SCC für Benutzer und Gruppen
- Verwalten von Secrets für Datenbanken und Anwendungen
- Beheben häufiger Probleme

### Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl:

- Stellt sicher, dass die Cluster-API erreichbar ist.
- Entfernt vorhandene Benutzer und Gruppen.
- Entfernt vorhandene Identitätsanbieter.
- Entfernt die Cluster-Rollenbindung `cluster-admin` vom `admin`-Benutzer.
- Stellt sicher, dass authentifizierte Benutzer neue Projekte erstellen können.

```
[student@workstation ~]$ lab review-troubleshoot start
```

### Anweisungen

Führen Sie die folgenden Aufgaben aus:

1. Erstellen Sie als Benutzer `kubeadmin` das Projekt `review-troubleshoot`. Das Passwort für den Benutzer `kubeadmin` befindet sich in der Datei `/usr/local/etc/ocp4.config`

in der Zeile RHT\_OCP4\_KUBEADM\_PASSWD. Führen Sie alle nachfolgenden Aufgaben im Projekt review-troubleshoot aus.

2. Führen Sie einen Smoke-Test des Clusters durch, um die grundlegende Funktionalität des Clusters zu überprüfen. Verwenden Sie eine deployment-Ressource, um eine Anwendung mit dem Namen hello-world-nginx zu erstellen. Der Anwendungsquellcode befindet sich im Unterverzeichnis hello-world-nginx des Repositorys <https://github.com/RedHatTraining/D0280-apps>.  
Erstellen Sie eine Route für die Anwendung mit einem beliebigen verfügbaren Hostnamen in der Subdomain apps.ocp4.example.com, und überprüfen Sie anschließend, ob die Anwendung auf externe Anforderungen reagiert.
3. Konfigurieren Sie den Cluster für die Verwendung eines HTPasswd-Identitätsanbieters. Der Name des Identitätsanbieters lautet cluster-users. Der Identitätsanbieter liest htpasswd-Anmeldedaten, die im Secret compreview-users gespeichert sind.  
Stellen Sie sicher, dass vier Benutzerkonten vorhanden sind: admin, leader, developer und qa-engineer. Alle Benutzerkonten müssen review als Passwort verwenden.  
Fügen Sie dem Benutzer admin die Rolle cluster-admin hinzu.
4. Erstellen Sie als admin-Benutzer drei Benutzergruppen: leaders, developers und qa.  
Weisen Sie der Gruppe leaders den Benutzer leader, der Gruppe developers den Benutzer developer und der Gruppe qa den Benutzer qa-engineer zu.  
Weisen Sie jeder Gruppe Rollen zu:
  - Weisen Sie der Gruppe leaders die Rolle self-provisioner zu. Mitglieder dieser Gruppe können dann Projekte erstellen. Damit diese Rolle wirksam ist, müssen Sie auch die Berechtigung für alle authentifizierten Benutzer, neue Projekte zu erstellen, aufheben.
  - Weisen Sie der Gruppe developers die Rolle edit nur für das Projekt review-troubleshoot zu. Mit dieser Rolle können die Mitglieder Projektressourcen erstellen und löschen.
  - Weisen Sie der Gruppe qa die Rolle view nur für das Projekt review-troubleshoot zu. Diese Rolle gewährt Mitgliedern der Gruppe Lesezugriff auf Projektressourcen.
5. Erstellen Sie als developer-Benutzer mit deployment eine Anwendung mit dem Namen mysql im Projekt review-troubleshoot. Verwenden Sie das unter registry.redhat.io/rhel8/mysql-80:1-139 verfügbare Image. Diese Anwendung bietet einen gemeinsamen Datenbankservice für andere Projektanwendungen.  
Erstellen Sie ein generisches Secret mit dem Namen mysql, und verwenden Sie dabei password als Schlüssel und r3dh4t123 als Wert.  
Legen Sie die Umgebungsvariable MYSQL\_ROOT\_ aus den Werten im mysql-Geheimnis fest.  
Konfigurieren Sie die MySQL-Datenbankanwendung so, dass eine Anforderung für ein persistentes Volume (PVC) im Verzeichnis /var/lib/mysql/data im Pod gemountet wird. Die PVC muss 2 GB groß sein und darf nur den ReadWriteOnce-Zugriffsmodus anfordern.
6. Erstellen Sie als developer-Benutzer mit deployment eine Anwendung mit dem Namen wordpress. Erstellen Sie die Anwendung im Projekt review-troubleshoot. Verwenden Sie das unter quay.io/redhattraining/wordpress:5.7-php7.4-apache verfügbare Image.  
Die Wordpress-Anwendung benötigt mehrere festgelegte Umgebungsvariablen. Die erforderlichen Umgebungsvariablen sind: WORDPRESS\_DB\_HOST mit dem Wert mysql, WORDPRESS\_DB\_NAME mit dem Wert wordpress, WORDPRESS\_USER mit dem Wert wpuser, WORDPRESS\_PASSWORD mit dem Wert wppass, WORDPRESS\_TITLE mit dem Wert review-troubleshoot, WORDPRESS\_URL mit dem Wert wordpress.

## Kapitel 9 | Ausführliche Wiederholung

`${RHT_OCP4_WILDCARD_DOMAIN}` und `WORDPRESS_EMAIL` mit dem Wert `student@redhat.com`.

Legen Sie die `WORDPRESS_DB_*`-Umgebungsvariablen so fest, dass ihre Werte aus dem Geheimnis `mysql` abgerufen werden.

Für die Anwendung `wordpress` ist die Sicherheitskontextbeschränkung `anyuid` erforderlich. Erstellen Sie ein Servicekonto mit dem Namen `wordpress-sa`, und weisen Sie ihm dann die Sicherheitskontextbeschränkung `anyuid` zu. Konfigurieren Sie die `wordpress`-Bereitstellung so, dass sie das Servicekonto `wordpress-sa` verwendet.

Die Anwendung `wordpress` erfordert auch, dass die Datenbank `WORDPRESS_DB_NAME` auf dem Datenbankserver vorhanden ist. Erstellen Sie eine leere Datenbank mit dem Namen `wordpress`.

Erstellen Sie eine Route für die Anwendung mit einem beliebigen verfügbaren Hostnamen in der Subdomain `apps.ocp4.example.com`. Wenn Sie die Anwendung ordnungsgemäß bereitgestellt haben und über einen Browser auf die Anwendung zugreifen, wird ein Installations-Assistent angezeigt.

7. Stellen Sie als `developer`-Benutzer die Anwendung `famous-quotes` im Projekt `review-troubleshoot` bereit, indem Sie das Skript `~/D0280/labs/review-troubleshoot/deploy_famous-quotes.sh` ausführen. Dieses Skript erstellt die Datenbank `defaultdb` und die in der Datei `~/D0280/labs/review-troubleshoot/famous-quotes.yaml` definierten Ressourcen.

Verwenden Sie das Geheimnis `mysql`, um Umgebungsvariablen in der Bereitstellung `famous-quotes` mit dem Präfix `QUOTES_` zu initialisieren.

Die Anwendungs-Pods werden erst nach der Ausführung des Skripts bereitgestellt. In der Bereitstellung `famous-quotes` ist ein Knoten-Selektor angegeben, aber es sind keine Cluster-Knoten mit einem übereinstimmenden Knoten-Label vorhanden.

Entfernen Sie den Knoten-Selektor aus der Bereitstellung, damit OpenShift die Anwendungs-Pods einem verfügbaren Knoten zuordnen kann.

Erstellen Sie eine Route für die Anwendung `famous-quotes` mit einem beliebigen verfügbaren Hostnamen in der Subdomain `apps.ocp4.example.com`, und überprüfen Sie anschließend, ob die Anwendung auf externe Anforderungen reagiert.

## 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.

```
[student@workstation ~]$ lab review-troubleshoot grade
```

## Beenden

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab review-troubleshoot finish
```

Hiermit ist die praktische Übung beendet.

## ► Lösung

# Fehlerbehebung in OpenShift-Clustern und -Anwendungen

In dieser Übung erlauben Sie Entwicklern den Zugriff auf einen Cluster und die Fehlerbehebung bei Anwendungsbereitstellungen.

## Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen Sie ein neues Projekt
- Führen Sie einen Smoke-Test des OpenShift-Clusters durch, indem Sie eine Anwendung mithilfe des Source-to-Image-Prozesses erstellen.
- Erstellen Sie Anwendungen mithilfe der `deployment`-Ressource.
- Verwalten von Benutzern mit dem HTPasswd-Identitätsanbieter
- Erstellen und Verwalten von Gruppen.
- Verwalten von RBAC und SCC für Benutzer und Gruppen
- Verwalten von Secrets für Datenbanken und Anwendungen
- Beheben häufiger Probleme

## Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl:

- Stellt sicher, dass die Cluster-API erreichbar ist.
- Entfernt vorhandene Benutzer und Gruppen.
- Entfernt vorhandene Identitätsanbieter.
- Entfernt die Cluster-Rollenbindung `cluster-admin` vom `admin`-Benutzer.
- Stellt sicher, dass authentifizierte Benutzer neue Projekte erstellen können.

```
[student@workstation ~]$ lab review-troubleshoot start
```

## Anweisungen

Führen Sie die folgenden Aufgaben aus:

1. Erstellen Sie als Benutzer `kubeadmin` das Projekt `review-troubleshoot`. Das Passwort für den Benutzer `kubeadmin` befindet sich in der Datei `/usr/local/etc/ocp4.config`

## Kapitel 9 | Ausführliche Wiederholung

in der Zeile RHT\_OCP4\_KUBEADM\_PASSWD. Führen Sie alle nachfolgenden Aufgaben im Projekt review-troubleshoot aus.

- 1.1. Suchen Sie die Kursumgebungs-Konfigurationsdatei unter /usr/local/etc/ocp4.config, und melden Sie sich als kubeadmin-Benutzer an.

```
[student@workstation ~]$ source /usr/local/etc/ocp4.config
[student@workstation ~]$ oc login -u kubeadmin -p ${RHT_OCP4_KUBEADM_PASSWD} \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Erstellen Sie das Projekt review-troubleshoot.

```
[student@workstation ~]$ oc new-project review-troubleshoot
Now using project "review-troubleshoot" on server
"https://api.ocp4.example.com:6443".
...output omitted...
```

2. Führen Sie einen Smoke-Test des Clusters durch, um die grundlegende Funktionalität des Clusters zu überprüfen. Verwenden Sie eine deployment-Ressource, um eine Anwendung mit dem Namen hello-world-nginx zu erstellen. Der Anwendungsquellcode befindet sich im Unterverzeichnis hello-world-nginx des Repositorys <https://github.com/RedHatTraining/D0280-apps>.

Erstellen Sie eine Route für die Anwendung mit einem beliebigen verfügbaren Hostnamen in der Subdomain apps.ocp4.example.com, und überprüfen Sie anschließend, ob die Anwendung auf externe Anforderungen reagiert.

- 2.1. Führen Sie den Befehl oc new-app aus, um die Bereitstellung hello-world-nginx zu erstellen.

```
[student@workstation ~]$ oc new-app --name hello-world-nginx \
> https://github.com/RedHatTraining/D0280-apps \
> --context-dir hello-world-nginx
...output omitted...
--> Creating resources ...
imagestream.image.openshift.io "ubi8" created
imagestream.image.openshift.io "hello-world-nginx" created
buildconfig.build.openshift.io "hello-world-nginx" created
deployment.apps "hello-world-nginx" created
service "hello-world-nginx" created
--> Success
...output omitted...
```

- 2.2. Erstellen Sie eine Route zu der Anwendung, indem Sie den Service hello-world-nginx bereitstellen.

```
[student@workstation ~]$ oc expose service hello-world-nginx \
> --hostname hello-world.apps.ocp4.example.com
route.route.openshift.io/hello-world-nginx exposed
```

- 2.3. Warten Sie, bis der Anwendungs-Pod ausgeführt wird.

```
[student@workstation ~]$ oc get pods
NAME READY STATUS RESTARTS AGE
hello-world-nginx-1-build 0/1 Completed 0 2m59s
hello-world-nginx-695754d9f7-8rv4x 1/1 Running 0 100s
```

2.4. Überprüfen Sie den Zugriff auf die Anwendung.

```
[student@workstation ~]$ curl -s http://hello-world.apps.ocp4.example.com \
> | grep Hello
<h1>Hello, world from nginx!</h1>
```

3. Konfigurieren Sie den Cluster für die Verwendung eines HTPasswd-Identitätsanbieters. Der Name des Identitätsanbieters lautet `cluster-users`. Der Identitätsanbieter liest `htpasswd`-Anmeldedaten, die im Secret `comprevue-users` gespeichert sind.  
Stellen Sie sicher, dass vier Benutzerkonten vorhanden sind: `admin`, `leader`, `developer` und `qa-engineer`. Alle Benutzerkonten müssen `review` als Passwort verwenden.  
Fügen Sie dem Benutzer `admin` die Rolle `cluster-admin` hinzu.

- 3.1. Erstellen Sie eine temporäre `htpasswd`-Authentifizierungsdatei unter `/tmp/cluster-users`.

```
[student@workstation ~]$ touch /tmp/cluster-users
```

- 3.2. Füllen Sie die Datei `/tmp/cluster-users` mit den erforderlichen Benutzer- und Passwortwerten.

```
[student@workstation ~]$ for user in admin leader developer qa-engineer
> do
> htpasswd -B -b /tmp/cluster-users ${user} review
> done
Adding password for user admin
Adding password for user leader
Adding password for user developer
Adding password for user qa-engineer
```

- 3.3. Erstellen Sie ein `comprevue-users`-Secret aus der Datei `/tmp/cluster-users`.

```
[student@workstation ~]$ oc create secret generic comprevue-users \
> --from-file htpasswd=/tmp/cluster-users -n openshift-config
secret/comprevue-users created
```

- 3.4. Exportieren Sie die vorhandene OAuth-Ressource in eine YAML-Datei.

```
[student@workstation ~]$ oc get oauth cluster -o yaml > /tmp/oauth.yaml
```

- 3.5. Bearbeiten Sie die Datei `/tmp/oauth.yaml`, um die Identitätsanbieterdefinition `HTPasswd` zur Liste `identityProviders` hinzuzufügen. Legen Sie den Namen des Identitätsanbieters auf `cluster-users` und den `fileData`-Namen auf `comprevue-users` fest.

## Kapitel 9 | Ausführliche Wiederholung

Nachdem Sie diese Änderungen vorgenommen haben, sieht die Datei folgendermaßen aus:

```
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
 ...output omitted...
 name: cluster
 ...output omitted...
spec:
 identityProviders:
 - name: cluster-users
 mappingMethod: claim
 type: HTPasswd
 htpasswd:
 fileData:
 name: compreview-users
```



### Anmerkung

Für die Schlüssel `name`, `mappingMethod`, `type` und `htpasswd` wird jeweils die gleiche Einrückung verwendet.

- 3.6. Ersetzen Sie die vorhandene OAuth-Ressource durch die Ressourcendefinition in der geänderten Datei:

```
[student@workstation ~]$ oc replace -f /tmp/oauth.yaml
oauth.config.openshift.io/cluster replaced
```

- 3.7. Weisen Sie dem Benutzer `admin` die Rolle `cluster-admin` zu.

```
[student@workstation ~]$ oc adm policy add-cluster-role-to-user \
> cluster-admin admin
Warning: User 'admin' not found
clusterrole.rbac.authorization.k8s.io/cluster-admin added: "admin"
```



### Wichtig

Sie können die Warnung, dass der `admin`-Benutzer nicht gefunden wurde, bedenkenlos ignorieren. Die Ressource `user/admin` ist erst dann im OpenShift-Cluster vorhanden, wenn sich der `admin`-Benutzer zum ersten Mal anmeldet.

4. Erstellen Sie als `admin`-Benutzer drei Benutzergruppen: `leaders`, `developers` und `qa`. Weisen Sie der Gruppe `leaders` den Benutzer `leader`, der Gruppe `developers` den Benutzer `developer` und der Gruppe `qa` den Benutzer `qa-engineer` zu.  
Weisen Sie jeder Gruppe Rollen zu:
  - Weisen Sie der Gruppe `leaders` die Rolle `self-provisioner` zu. Mitglieder dieser Gruppe können dann Projekte erstellen. Damit diese Rolle wirksam ist, müssen Sie auch die Berechtigung für alle authentifizierten Benutzer, neue Projekte zu erstellen, aufheben.

**Kapitel 9 |** Ausführliche Wiederholung

- Weisen Sie der Gruppe `developers` die Rolle `edit` nur für das Projekt `review-troubleshoot` zu. Mit dieser Rolle können die Mitglieder Projektressourcen erstellen und löschen.
- Weisen Sie der Gruppe `qa` die Rolle `view` nur für das Projekt `review-troubleshoot` zu. Diese Rolle gewährt Mitgliedern der Gruppe Lesezugriff auf Projektressourcen.

4.1. Melden Sie sich als Benutzer `admin` an.

```
[student@workstation ~]$ oc login -u admin -p review
Login successful.
...output omitted...
```

4.2. Erstellen Sie die drei Benutzergruppen.

```
[student@workstation ~]$ for group in leaders developers qa
> do
> oc adm groups new ${group}
> done
group.user.openshift.io/leaders created
group.user.openshift.io/developers created
group.user.openshift.io/qa created
```

4.3. Fügen Sie die einzelnen Benutzer der entsprechenden Gruppe hinzu.

```
[student@workstation ~]$ oc adm groups add-users leaders leader
group.user.openshift.io/leaders added: "leader"
[student@workstation ~]$ oc adm groups add-users developers developer
group.user.openshift.io/developers added: "developer"
[student@workstation ~]$ oc adm groups add-users qa qa-engineer
group.user.openshift.io/qa added: "qa-engineer"
```

4.4. Erteilen Sie Mitgliedern der Gruppe `leaders` die Berechtigung, neue Projekte zu erstellen:

```
[student@workstation ~]$ oc adm policy add-cluster-role-to-group \
> self-provisioner leaders
clusterrole.rbac.authorization.k8s.io/self-provisioner added: "leaders"
```

4.5. Entfernen Sie die Cluster-Rolle `self-provisioner` aus der Gruppe `system:authenticated:oauth`.

```
[student@workstation ~]$ oc adm policy remove-cluster-role-from-group \
> self-provisioner system:authenticated:oauth
Warning: Your changes may get lost whenever a master is restarted,
unless you prevent reconciliation of this rolebinding using the
following command: oc annotate clusterrolebinding.rbac self-provisioners
'rbac.authorization.kubernetes.io/autoupdate=false' --overwrite
clusterrole.rbac.authorization.k8s.io/self-provisioner removed:
"system:authenticated:oauth"
```

**Kapitel 9 |** Ausführliche Wiederholung

- 4.6. Erteilen Sie Mitgliedern der Gruppe `developers` die Berechtigung, Ressourcen im Projekt `review-troubleshoot` zu erstellen und zu löschen:

```
[student@workstation ~]$ oc policy add-role-to-group edit developers
clusterrole.rbac.authorization.k8s.io/edit added: "developers"
```

- 4.7. Erteilen Sie Mitgliedern der Gruppe `qa` die Berechtigung, Projektressourcen anzuzeigen:

```
[student@workstation ~]$ oc policy add-role-to-group view qa
clusterrole.rbac.authorization.k8s.io/view added: "qa"
```

5. Erstellen Sie als `developer`-Benutzer mit `deployment` eine Anwendung mit dem Namen `mysql` im Projekt `review-troubleshoot`. Verwenden Sie das unter `registry.redhat.io/rhel8/mysql-80:1-139` verfügbare Image. Diese Anwendung bietet einen gemeinsamen Datenbankservice für andere Projektanwendungen.

Erstellen Sie ein generisches Secret mit dem Namen `mysql`, und verwenden Sie dabei `password` als Schlüssel und `r3dh4t123` als Wert.

Legen Sie die Umgebungsvariable `MYSQL_ROOT_` aus den Werten im `mysql`-Geheimnis fest. Konfigurieren Sie die MySQL-Datenbankanwendung so, dass eine Anforderung für ein persistentes Volume (PVC) im Verzeichnis `/var/lib/mysql/data` im Pod gemountet wird. Die PVC muss 2 GB groß sein und darf nur den `ReadWriteOnce`-Zugriffsmodus anfordern.

- 5.1. Melden Sie sich als Benutzer `developer` bei dem Cluster an.

```
[student@workstation ~]$ oc login -u developer -p review
Login successful.
...output omitted...
```

- 5.2. Erstellen Sie eine neue Anwendung, um einen `mysql`-Datenbankserver bereitzustellen. Verwenden Sie den Befehl `oc new-app` zur Erstellung einer Bereitstellung.

```
[student@workstation ~]$ oc new-app --name mysql \
> --docker-image registry.redhat.io/rhel8/mysql-80:1-139
...output omitted...
--> Creating resources ...
 imagestream.image.openshift.io "mysql" created
 deployment.apps "mysql" created
 service "mysql" created
--> Success
...output omitted...
```

- 5.3. Erstellen Sie ein generisches Secret für die MySQL-Datenbank mit dem Namen `mysql`, und verwenden Sie dabei `password` als Schlüssel und `r3dh4t123` als Wert.

```
[student@workstation ~]$ oc create secret generic mysql \
> --from-literal password=r3dh4t123
secret/mysql created
```

- 5.4. Verwenden Sie das Secret `mysql`, um Umgebungsvariablen in der Bereitstellung `mysql` zu initialisieren.

**Kapitel 9 |** Ausführliche Wiederholung

```
[student@workstation ~]$ oc set env deployment mysql \
> --prefix MYSQL_ROOT_ --from secret/mysql
deployment.apps/mysql updated
```

- 5.5. Verwenden Sie den Befehl `oc set volumes`, um persistenten Storage für die `mysql`-Bereitstellung zu konfigurieren. Der Befehl erstellt automatisch eine Anforderung für ein persistentes Volume mit der angegebenen Größe und dem Zugriffsmodus. Durch das Mounten des Volumes im Verzeichnis `/var/lib/mysql/data` wird der Zugriff auf gespeicherte Daten ermöglicht, selbst wenn ein Datenbank-Pod gelöscht und ein anderer Datenbank-Pod erstellt wird.

```
[student@workstation ~]$ oc set volumes deployment/mysql --name mysql-storage \
> --add --type pvc --claim-size 2Gi --claim-mode rwo \
> --mount-path /var/lib/mysql/data
deployment.apps/mysql volume updated
```

- 5.6. Überprüfen Sie, ob der `mysql`-Pod erfolgreich erneut bereitgestellt wird, nachdem die Bereitstellung für die Verwendung eines Secret und eines Volumes konfiguriert wurde. Sie müssen den Befehl `oc get pods` möglicherweise mehrmals ausführen, bis der `mysql`-Pod mit `1/1` und `Running` angezeigt wird.

```
[student@workstation ~]$ oc get pods -l deployment=mysql
NAME READY STATUS RESTARTS AGE
mysql-bbb6b5fbb-dmq9x 1/1 Running 0 63s
```

- 5.7. Überprüfen Sie, ob eine Anforderung für ein persistentes Volume mit der korrekten Größe und dem Zugriffsmodus existiert.

```
[student@workstation ~]$ oc get pvc
NAME STATUS ... CAPACITY ACCESS MODES STORAGECLASS AGE
pvc-ks52v Bound ... 2Gi RWO nfs-storage 2m33s
```

- 5.8. Überprüfen Sie, ob der ausgeführte `mysql`-Pod ein Volume im Verzeichnis `/var/lib/mysql/data` gemountet hat.

```
[student@workstation ~]$ oc exec mysql-bbb6b5fbb-dmq9x -- \
> df -h /var/lib/mysql/data
Filesystem
 Size Used Avail Use% Mounted on
192.168.50.254:/exports/review-troubleshoot-pvc-ks52v-pvc-0d6a63bd-286e-44ec-b29c-
ffbb34928b86 40G 859M 40G 3% /var/lib/mysql/data
```

6. Erstellen Sie als `developer`-Benutzer mit `deployment` eine Anwendung mit dem Namen `wordpress`. Erstellen Sie die Anwendung im Projekt `review-troubleshoot`. Verwenden Sie das unter `quay.io/redhattraining/wordpress:5.7-php7.4-apache` verfügbare Image.

Die Wordpress-Anwendung benötigt mehrere festgelegte Umgebungsvariablen. Die erforderlichen Umgebungsvariablen sind: `WORDPRESS_DB_HOST` mit dem Wert `mysql`, `WORDPRESS_DB_NAME` mit dem Wert `wordpress`, `WORDPRESS_USER` mit dem Wert `wuser`, `WORDPRESS_PASSWORD` mit dem Wert `wppass`, `WORDPRESS_TITLE` mit

**Kapitel 9 |** Ausführliche Wiederholung

dem Wert `review-troubleshoot`, `WORDPRESS_URL` mit dem Wert `wordpress`.  
 `${RHT_OCP4_WILDCARD_DOMAIN}` und `WORDPRESS_EMAIL` mit dem Wert  
`student@redhat.com`.

Legen Sie die `WORDPRESS_DB_*`-Umgebungsvariablen so fest, dass ihre Werte aus dem Geheimnis `mysql` abgerufen werden.

Für die Anwendung `wordpress` ist die Sicherheitskontextbeschränkung `anyuid` erforderlich. Erstellen Sie ein Servicekonto mit dem Namen `wordpress-sa`, und weisen Sie ihm dann die Sicherheitskontextbeschränkung `anyuid` zu. Konfigurieren Sie die `wordpress`-Bereitstellung so, dass sie das Servicekonto `wordpress-sa` verwendet.

Die Anwendung `wordpress` erfordert auch, dass die Datenbank `WORDPRESS_DB_NAME` auf dem Datenbankserver vorhanden ist. Erstellen Sie eine leere Datenbank mit dem Namen `wordpress`.

Erstellen Sie eine Route für die Anwendung mit einem beliebigen verfügbaren Hostnamen in der Subdomain `apps.ocp4.example.com`. Wenn Sie die Anwendung ordnungsgemäß bereitgestellt haben und über einen Browser auf die Anwendung zugreifen, wird ein Installations-Assistent angezeigt.

6.1. Stellen Sie eine `wordpress`-Anwendung als deployment bereit.

```
[student@workstation ~]$ oc new-app --name wordpress \
> --docker-image quay.io/redhattraining/wordpress:5.7-php7.4-apache \
> -e WORDPRESS_DB_HOST=mysql -e WORDPRESS_DB_NAME=wordpress \
> -e WORDPRESS_DB_USER=root \
> -e WORDPRESS_USER=wpuser -e WORDPRESS_PASSWORD=wppass \
> -e WORDPRESS_TITLE=review-troubleshoot \
> -e WORDPRESS_URL=wordpress.${RHT_OCP4_WILDCARD_DOMAIN} \
> -e WORDPRESS_EMAIL=student@redhat.com
...output omitted...
--> Creating resources ...
imagestream.image.openshift.io "wordpress" created
deployment.apps "wordpress" created
service "wordpress" created
--> Success
...output omitted...
```

6.2. Fügen Sie die `WORDPRESS_DB_*`-Umgebungsvariablen zur `wordpress`-Bereitstellung hinzu.

```
[student@workstation ~]$ oc set env deployment/wordpress \
> --prefix WORDPRESS_DB_ --from secret/mysql
deployment.apps/wordpress updated
```

6.3. Erstellen Sie das Servicekonto `wordpress-sa`.

```
[student@workstation ~]$ oc create serviceaccount wordpress-sa
serviceaccount/wordpress-sa created
```

6.4. Melden Sie sich als Benutzer `admin` bei dem Cluster an, und gewähren Sie dem Servicekonto `wordpress-sa` die Berechtigungen `anyuid`.

```
[student@workstation ~]$ oc login -u admin -p review
Login successful.
...output omitted...
[student@workstation ~]$ oc adm policy add-scc-to-user anyuid -z wordpress-sa
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:anyuid added:
"wordpress-sa"
```

- 6.5. Wechseln Sie zurück zum Benutzer developer, um die verbleibenden Schritte auszuführen. Melden Sie sich als Benutzer developer bei dem Cluster an.

```
[student@workstation ~]$ oc login -u developer -p review
Login successful.
...output omitted...
```

- 6.6. Konfigurieren Sie die wordpress-Bereitstellung so, dass sie das Servicekonto wordpress-sa verwendet.

```
[student@workstation ~]$ oc set serviceaccount deployment/wordpress \
> wordpress-sa
deployment.apps/wordpress serviceaccount updated
```

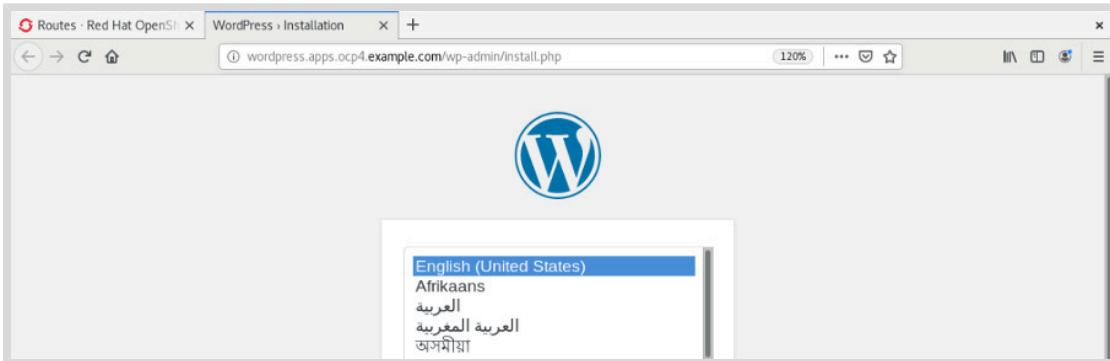
- 6.7. Erstellen Sie die wordpress-Datenbank. Verwenden Sie das mysql-Client-Tool auf dem ausgeführten Pod.

```
[student@workstation ~]$ oc get pods -l deployment=mysql
NAME READY STATUS RESTARTS AGE
mysql-fbf67ff96-c4sq7 1/1 Running 0 11s
[student@workstation ~]$ oc exec mysql-fbf67ff96-c4sq7 -- \
> /usr/bin/mysql -uroot -e "CREATE DATABASE wordpress"
```

- 6.8. Erstellen Sie eine Route für die Anwendung wordpress.

```
[student@workstation ~]$ oc expose service wordpress \
> --hostname wordpress.apps.ocp4.example.com
route.route.openshift.io/wordpress exposed
```

- 6.9. Überprüfen Sie mit einem Webbrowser den Zugriff auf die URL `http://wordpress.apps.ocp4.example.com`. Wenn Sie die Anwendung ordnungsgemäß bereitgestellt haben, wird ein Setup-Assistent in einem Browser angezeigt.



7. Stellen Sie als developer-Benutzer die Anwendung famous-quotes im Projekt review-troubleshoot bereit, indem Sie das Skript ~/D0280/labs/review-troubleshoot/deploy\_famous-quotes.sh ausführen. Dieses Skript erstellt die Datenbank defaultdb und die in der Datei ~/D0280/labs/review-troubleshoot/famous-quotes.yaml definierten Ressourcen.

Verwenden Sie das Geheimnis mysql, um Umgebungsvariablen in der Bereitstellung famous-quotes mit dem Präfix QUOTES\_ zu initialisieren.

Die Anwendungs-Pods werden erst nach der Ausführung des Skripts bereitgestellt. In der Bereitstellung famous-quotes ist ein Knoten-Selektor angegeben, aber es sind keine Cluster-Knoten mit einem übereinstimmenden Knoten-Label vorhanden.

Entfernen Sie den Knoten-Selektor aus der Bereitstellung, damit OpenShift die Anwendungs-Pods einem verfügbaren Knoten zuordnen kann.

Erstellen Sie eine Route für die Anwendung famous-quotes mit einem beliebigen verfügbaren Hostnamen in der Subdomain apps.ocp4.example.com, und überprüfen Sie anschließend, ob die Anwendung auf externe Anforderungen reagiert.

- 7.1. Führen Sie das Skript ~/D0280/labs/review-troubleshoot/deploy\_famous-quotes.sh aus.

```
[student@workstation ~]$ ~/D0280/labs/review-troubleshoot/deploy_famous-quotes.sh
Creating famous-quotes database
Deploying famous-quotes application
deployment.apps/famous-quotes created
service/famous-quotes created
```

- 7.2. Verwenden Sie das Geheimnis mysql, um Umgebungsvariablen mit dem Präfix QUOTES\_ zu initialisieren.

```
[student@workstation ~]$ oc set env deployment famous-quotes \
> --prefix QUOTES_ --from secret/mysql
```

- 7.3. Verifizieren Sie, dass der Anwendungs-Pod famous-quotes nicht für die Bereitstellung geplant ist.

```
[student@workstation ~]$ oc get pods
NAME READY STATUS RESTARTS AGE
famous-quotes-85ff8679d7-vhvbk 0/1 Pending 0 41s
...output omitted...
```

- 7.4. Überprüfen Sie, ob Projektereignisse Informationen zu dem Problem bereitstellen.

```
[student@workstation ~]$ oc get events --sort-by='-.lastTimestamp'
...output omitted...
34s Warning FailedScheduling pod/famous-quotes-1-deploy 0/3 nodes are
available: 3 node(s) didn't match node selector.
...output omitted...
```

- 7.5. Speichern Sie die Bereitstellungsressource famous-quotes in einer Datei.

```
[student@workstation ~]$ oc get deployment/famous-quotes \
> -o yaml > /tmp/famous-deploy.yaml
```

- 7.6. Verwenden Sie einen Editor, um den Knoten-Selektor aus der Datei /tmp/famous-deploy.yaml zu entfernen. Suchen Sie in der Datei nach der Zeile nodeSelector. Löschen Sie dann die folgenden zwei Zeilen aus der Datei /tmp/famous-deploy.yaml.

```
nodeSelector:
 env: quotes
```

- 7.7. Ersetzen Sie die Bereitstellung famous-quotes durch die geänderte Datei.

```
[student@workstation ~]$ oc replace -f /tmp/famous-deploy.yaml
deployment.apps/famous-quotes replaced
```

- 7.8. Warten Sie einige Augenblicke, und führen Sie dann den Befehl oc get pods aus, um sicherzustellen, dass die Anwendung famous-quotes jetzt ausgeführt wird.

```
[student@workstation ~]$ oc get pods -l app=famous-quotes
NAME READY STATUS RESTARTS AGE
famous-quotes-2-gmz2j 1/1 Running 0 53s
```

- 7.9. Erstellen Sie eine Route für die Anwendung famous-quotes.

```
[student@workstation ~]$ oc expose service famous-quotes \
> --hostname quotes.apps.ocp4.example.com
route.route.openshift.io/famous-quotes exposed
```

- 7.10. Überprüfen Sie, ob die Anwendung famous-quotes auf Anforderungen reagiert, die an die URL http://quotes.apps.ocp4.example.com gesendet werden.

```
[student@workstation ~]$ curl -s http://quotes.apps.ocp4.example.com \
> | grep Quote
 <title>Quotes</title>
 <h1>Quote List</h1>
```

## 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.

```
[student@workstation ~]$ lab review-troubleshoot grade
```

## Beenden

Führen Sie auf dem Rechner workstation als Benutzer student den Befehl lab aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab review-troubleshoot finish
```

Hiermit ist die praktische Übung beendet.

## ► Praktische Übung

# Konfigurieren einer Projektvorlage mit Ressourcen- und Netzwerkbeschränkungen

In dieser Übung konfigurieren Sie die standardmäßige Projektvorlage für Cluster, um sicherzustellen, dass neue Projekte Standardkontingente, Ressourcenbeschränkungen und Netzwerkrichtlinien beachten.

## Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Ändern der standardmäßigen Projektvorlage, sodass Einschränzungsbereiche, Ressourcenkontingente und Netzwerkrichtlinien automatisch erstellt werden
- Erstellen eines TLS-Geheimnisses mit den bereitgestellten Dateien
- Mounten eines Secrets als Volume in einer Anwendung
- Erstellen einer Passthrough-Route zu einer Anwendung
- Konfigurieren einer Anwendung für die automatische Skalierung

## Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl:

- Stellt sicher, dass die Cluster-API erreichbar ist.
- Konfiguriert den HTPasswd-Identitätsanbieter und ermöglicht den Zugriff für die Benutzer `admin`, `leader` und `developer`.
- Lädt YAML-Beispieldateien in `~/D0280/labs/review-template/sample-files/` herunter.
- Erstellt das Projekt `review-template-test` und stellt eine Anwendung für das Projekt bereit, mit dem Sie Ihre Netzwerkrichtlinien testen können.
- Fügt das Label `network.openshift.io/policy-group=ingress` zum `default-`Namespace hinzu.
- Generiert die benötigten Zertifikatdateien für die sichere Anwendung.

```
[student@workstation ~]$ lab review-template start
```



### Anmerkung

Wenn Sie Hilfe bei den ersten Schritten benötigen, können Sie das Kapitel *Post-installation network configuration* in der Anleitung für Aufgaben nach der Installation von Red Hat OpenShift Container Platform als Referenz verwenden.

## Anweisungen

Führen Sie die folgenden Aufgaben aus:

1. Aktualisieren Sie den OpenShift-Cluster als `admin`-Benutzer, um eine neue Projektvorlage zu verwenden. Die Projektvorlage muss automatisch: Netzwerkrichtlinien erstellen, Bereiche einschränken und Kontingentressourcen für neue Projekte hinzufügen. Neue Projekte

**Kapitel 9 |** Ausführliche Wiederholung

müssen automatisch ein Label erhalten, das mit dem Namen des Projekts übereinstimmt. Ein Projekt mit dem Namen `test` erhält beispielsweise das Label `name=test`.

In der folgenden Tabelle finden Sie eine Anleitung zu den benötigten Ressourcen.

Ressource	Anforderungen
Projekt	<ul style="list-style-type: none"> <li>Enthält ein Label mit dem Namen des Projekts.</li> </ul>
NetworkPolicy	<p>Richtlinie 1:</p> <ul style="list-style-type: none"> <li>Routen sind für den externen Datenverkehr zugänglich. Dies bedeutet, dass der Datenverkehr von Pods in Namespaces mit dem Label <code>network.openshift.io/policy-group=ingress</code> zugelassen wird.</li> </ul> <p>Richtlinie 2:</p> <ul style="list-style-type: none"> <li>Pods im selben Namespace können miteinander kommunizieren.</li> <li>Pods reagieren nicht auf Pods in anderen Namespaces, mit Ausnahme von Namespaces mit dem Label <code>network.openshift.io/policy-group=ingress</code>.</li> </ul>
LimitRange	<ul style="list-style-type: none"> <li>Jeder Container fordert 30 Millicores CPU an.</li> <li>Jeder Container fordert 30 MiB Arbeitsspeicher an.</li> <li>Jeder Container ist auf 100 Millicores CPU beschränkt.</li> <li>Jeder Container ist auf 100 MiB Arbeitsspeicher beschränkt.</li> </ul>
ResourceQuota	<ul style="list-style-type: none"> <li>Projekte sind auf 10 Pods beschränkt.</li> <li>Projekte können maximal 1 GiB Arbeitsspeicher anfordern.</li> <li>Projekte können maximal 2 CPUs anfordern.</li> <li>Projekte können maximal 4 GiB Arbeitsspeicher verwenden.</li> <li>Projekte können maximal 4 CPUs verwenden.</li> </ul>

- Erstellen Sie als Benutzer `developer` ein neues Projekt mit dem Namen `review-template`. Stellen Sie sicher, dass das Projekt `review-template` die in der neuen Projektvorlage angegebenen Einstellungen übernimmt. Erstellen Sie im Projekt `review-template` eine neue Bereitstellung mit dem Namen `hello-secure` und dem Container-Image unter `quay.io/redhattraining/hello-world-secure:v1.0`.

**Anmerkung**

Der `hello-secure`-Pod wird erst erfolgreich ausgeführt, nachdem Sie den Zugriff auf das TLS-Zertifikat und den Schlüssel für den NGINX-Server bereitgestellt haben.

- Erstellen Sie als Benutzer `developer` ein TLS-Secret mit dem Zertifikat `hello-secure-combined.pem` und dem Schlüssel `hello-secure-key.pem` aus dem Verzeichnis `~/D0280/labs/review-template/`. Verwenden Sie die Protokolle aus dem

fehlgeschlagenen `hello-secure`-Pod, um den erwarteten Mount-Punkt für das Zertifikat zu ermitteln. Mounten Sie das TLS-Secret mit dem angegebenen Verzeichnis als Volume im Pod. Vergewissern Sie sich, dass der Pod `hello-secure` erfolgreich erneut bereitgestellt wird.

4. Das Zertifikat `hello-secure-combined.pem` ist für einen einzelnen Hostnamen gültig. Verwenden Sie den Befehl `openssl x509` mit den Optionen `-noout` und `-ext 'subjectAltName'`, um das Zertifikat `hello-secure-combined.pem` zu lesen und den Hostnamen zu ermitteln. Erstellen Sie als Benutzer `developer` eine Passthrough-Route zum `hello-secure`-Service mit dem identifizierten Hostnamen. Überprüfen Sie, ob die Route auf externe Anforderungen reagiert.



#### Anmerkung

Die Manpage `x509 (1)` enthält Informationen zum `openssl x509`-Befehl.

5. Konfigurieren Sie als Benutzer `developer` die `hello-secure`-Bereitstellung so, dass sie automatisch skaliert wird. In der Bereitstellung muss mindestens ein Pod ausgeführt werden. Wenn die durchschnittliche CPU-Auslastung 80 % überschreitet, wird die Bereitstellung auf bis zu fünf Pods skaliert.



#### Anmerkung

Mit dem Skript unter `~/D0280/solutions/review-template/test-hpa.sh` können Sie testen, ob Ihre Bereitstellung erwartungsgemäß skaliert wird.

## 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.

```
[student@workstation ~]$ lab review-template grade
```

## Beenden

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab review-template finish
```

Hiermit ist die praktische Übung beendet.

## ► Lösung

# Konfigurieren einer Projektvorlage mit Ressourcen- und Netzwerkbeschränkungen

In dieser Übung konfigurieren Sie die standardmäßige Projektvorlage für Cluster, um sicherzustellen, dass neue Projekte Standardkontingente, Ressourcenbeschränkungen und Netzwerkrichtlinien beachten.

## Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Ändern der standardmäßigen Projektvorlage, sodass Einschränzungsbereiche, Ressourcenkontingente und Netzwerkrichtlinien automatisch erstellt werden
- Erstellen eines TLS-Geheimnisses mit den bereitgestellten Dateien
- Mounten eines Secrets als Volume in einer Anwendung
- Erstellen einer Passthrough-Route zu einer Anwendung
- Konfigurieren einer Anwendung für die automatische Skalierung

## Bevor Sie Beginnen

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um Ihr System für diese Übung vorzubereiten.

Der Befehl:

- Stellt sicher, dass die Cluster-API erreichbar ist.
- Konfiguriert den HTPasswd-Identitätsanbieter und ermöglicht den Zugriff für die Benutzer `admin`, `leader` und `developer`.
- Lädt YAML-Beispieldateien in `~/D0280/labs/review-template/sample-files/` herunter.
- Erstellt das Projekt `review-template-test` und stellt eine Anwendung für das Projekt bereit, mit dem Sie Ihre Netzwerkrichtlinien testen können.
- Fügt das Label `network.openshift.io/policy-group=ingress` zum `default-`Namespace hinzu.
- Generiert die benötigten Zertifikatdateien für die sichere Anwendung.

```
[student@workstation ~]$ lab review-template start
```



### Anmerkung

Wenn Sie Hilfe bei den ersten Schritten benötigen, können Sie das Kapitel *Post-installation network configuration* in der Anleitung für Aufgaben nach der Installation von Red Hat OpenShift Container Platform als Referenz verwenden.

## Anweisungen

Führen Sie die folgenden Aufgaben aus:

1. Aktualisieren Sie den OpenShift-Cluster als `admin`-Benutzer, um eine neue Projektvorlage zu verwenden. Die Projektvorlage muss automatisch: Netzwerkrichtlinien erstellen, Bereiche einschränken und Kontingentressourcen für neue Projekte hinzufügen. Neue Projekte

**Kapitel 9 |** Ausführliche Wiederholung

müssen automatisch ein Label erhalten, das mit dem Namen des Projekts übereinstimmt. Ein Projekt mit dem Namen `test` erhält beispielsweise das Label `name=test`.

In der folgenden Tabelle finden Sie eine Anleitung zu den benötigten Ressourcen.

Ressource	Anforderungen
Projekt	<ul style="list-style-type: none"> <li>Enthält ein Label mit dem Namen des Projekts.</li> </ul>
NetworkPolicy	<p>Richtlinie 1:</p> <ul style="list-style-type: none"> <li>Routen sind für den externen Datenverkehr zugänglich. Dies bedeutet, dass der Datenverkehr von Pods in Namespaces mit dem Label <code>network.openshift.io/policy-group=ingress</code> zugelassen wird.</li> </ul> <p>Richtlinie 2:</p> <ul style="list-style-type: none"> <li>Pods im selben Namespace können miteinander kommunizieren.</li> <li>Pods reagieren nicht auf Pods in anderen Namespaces, mit Ausnahme von Namespaces mit dem Label <code>network.openshift.io/policy-group=ingress</code>.</li> </ul>
LimitRange	<ul style="list-style-type: none"> <li>Jeder Container fordert 30 Millicores CPU an.</li> <li>Jeder Container fordert 30 MiB Arbeitsspeicher an.</li> <li>Jeder Container ist auf 100 Millicores CPU beschränkt.</li> <li>Jeder Container ist auf 100 MiB Arbeitsspeicher beschränkt.</li> </ul>
ResourceQuota	<ul style="list-style-type: none"> <li>Projekte sind auf 10 Pods beschränkt.</li> <li>Projekte können maximal 1 GiB Arbeitsspeicher anfordern.</li> <li>Projekte können maximal 2 CPUs anfordern.</li> <li>Projekte können maximal 4 GiB Arbeitsspeicher verwenden.</li> <li>Projekte können maximal 4 CPUs verwenden.</li> </ul>

- 1.1. Melden Sie sich als Benutzer `admin` bei Ihrem OpenShift-Cluster an.

```
[student@workstation ~]$ oc login -u admin -p redhat \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

- 1.2. Verwenden Sie den Befehl `oc adm create-bootstrap-project-template`, um eine neue YAML-Datei zu erstellen, die Sie für diese Übung anpassen werden. Speichern Sie die Datei unter `~/D0280/labs/review-template/project-template.yaml`.

**Kapitel 9 |** Ausführliche Wiederholung

```
[student@workstation ~]$ oc adm create-bootstrap-project-template \
> -o yaml > ~/D0280/labs/review-template/project-template.yaml
```

- 1.3. Wechseln Sie zum Verzeichnis ~/D0280/labs/review-template/.

```
[student@workstation ~]$ cd ~/D0280/labs/review-template/
```

- 1.4. Bearbeiten Sie die Datei project-template.yaml, und fügen Sie ein Label für neue Projekte hinzu. Fügen Sie die fett formatierten Zeilen mit der richtigen Einrückung hinzu, und speichern Sie die Datei. Die Variable PROJECT\_NAME erhält den Wert des Projektnamens.

```
...output omitted...
- apiVersion: project.openshift.io/v1
 kind: Project
 metadata:
 labels:
 name: ${PROJECT_NAME}
 annotations:
 ...output omitted...
```

- 1.5. Listen Sie die Dateien im Verzeichnis ~/D0280/labs/review-template/sample-files/ auf. Das Verzeichnis enthält zwei Beispieldateien mit Netzwerkrichtlinien: eine Datei für einen Einschränkungsbereich und eine Datei für ein Ressourcenkontingent.

```
[student@workstation review-template]$ ls sample-files/
allow-from-openshift-ingress.yaml allow-same-namespace.yaml limitrange.yaml
resourcequota.yaml
```

- 1.6. Fügen Sie den Inhalt der Dateien sample-files/allow-\*.yaml zur Datei project-template.yaml hinzu. Da die Datei project-template.yaml eine Liste von Ressourcen erwartet, muss die erste Zeile jeder Ressource mit einem - beginnen, und der Rest des Inhalts muss entsprechend eingerückt sein.

Bearbeiten Sie die Datei project-template.yaml, und fügen Sie die Ressourcen für Netzwerkrichtlinien hinzu. Fügen Sie die fett formatierten Zeilen mit der richtigen Einrückung hinzu, und speichern Sie die Datei.

```
...output omitted...
subjects:
- apiGroup: rbac.authorization.k8s.io
 kind: User
 name: ${PROJECT_ADMIN_USER}
- apiVersion: networking.k8s.io/v1
 kind: NetworkPolicy
 metadata:
 name: allow-from-openshift-ingress
 spec:
 podSelector: {}
 ingress:
 - from:
 - namespaceSelector:
```

```
matchLabels:
 network.openshift.io/policy-group: ingress
- apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
 name: allow-same-namespace
spec:
 podSelector: {}
 ingress:
 - from:
 - podSelector: {}
parameters:
- name: PROJECT_NAME
...output omitted...
```

- 1.7. Fügen Sie den Inhalt der Dateien `sample-files/limitrange.yaml` und `sample-files/resourcequota.yaml` zur Datei `project-template.yaml` hinzu. Da die Datei `project-template.yaml` eine Liste von Ressourcen erwartet, muss die erste Zeile jeder Ressource mit einem `-` beginnen, und der Rest des Inhalts muss entsprechend eingerückt sein.

Bearbeiten Sie die Datei `project-template.yaml`, und fügen Sie die Ressourcen für Einschränkungsbereiche und Ressourcenkontingente hinzu. Fügen Sie die fett formatierten Zeilen mit der richtigen Einrückung hinzu, und speichern Sie die Datei.

```
...output omitted...
ingress:
 - from:
 - podSelector: {}
- apiVersion: v1
kind: LimitRange
metadata:
 name: project-limitrange
spec:
 limits:
 - default:
 memory: 100Mi
 cpu: 100m
 defaultRequest:
 memory: 30Mi
 cpu: 30m
 type: Container
- apiVersion: v1
kind: ResourceQuota
metadata:
 name: project-quota
spec:
 hard:
 pods: '10'
 requests.cpu: '2'
 requests.memory: 1Gi
 limits.cpu: '4'
 limits.memory: 4Gi
```

```
parameters:
- name: PROJECT_NAME
...output omitted...
```

- 1.8. Führen Sie den Befehl `oc create` aus, um eine neue Vorlagenressource im Namespace `openshift-config` mit der Datei `project-template.yaml` zu erstellen.



### Anmerkung

Die Datei `/home/student/D0280/solutions/review-template/project-template.yaml` enthält die richtige Konfiguration und kann als Vergleich verwendet werden.

```
[student@workstation review-template]$ oc create -f project-template.yaml \
> -n openshift-config
template.template.openshift.io/project-request created
```

- 1.9. Listen Sie die Vorlagen im Namespace `openshift-config` auf. Sie verwenden den Vorlagennamen im nächsten Schritt.

```
[student@workstation review-template]$ oc get templates -n openshift-config
NAME DESCRIPTION PARAMETERS OBJECTS
project-request 5 (5 blank) 6
```

- 1.10. Aktualisieren Sie die Ressource `projects.config.openshift.io/cluster` so, dass sie die neue Vorlage verwendet.

```
[student@workstation review-template]$ oc edit \
> projects.config.openshift.io/cluster
```

Ändern Sie die Zeile `spec: {}` so, dass Sie den folgenden fett formatierten Zeilen entspricht. Verwenden Sie den Vorlagennamen, den Sie im Namespace `openshift-config` identifiziert haben. Überprüfen Sie die Einrückung und speichern Sie Ihre Änderungen.

```
...output omitted...
spec:
 projectRequestTemplate:
 name: project-request
```

- 1.11. Wenn die Änderung erfolgreich war, werden die `apiserver`-Pods im `openshift-apiserver`-Namespace neu bereitgestellt. Überwachen Sie die erneute Bereitstellung.

```
[student@workstation review-template]$ watch oc get pods -n openshift-apiserver
```

Drücken Sie Strg+C, um den Befehl `watch` zu beenden, nachdem alle drei neuen Pods ausgeführt wurden.

```
Every 2.0s: oc get pods -n openshift-apiserver
...
NAME READY STATUS RESTARTS AGE
apiserver-75cfdfc877-257vs 2/2 Running 0 61s
apiserver-75cfdfc877-l2xnv 2/2 Running 0 29s
apiserver-75cfdfc877-rn9fs 2/2 Running 0 47s
```

2. Erstellen Sie als Benutzer `developer` ein neues Projekt mit dem Namen `review-template`. Stellen Sie sicher, dass das Projekt `review-template` die in der neuen Projektvorlage angegebenen Einstellungen übernimmt. Erstellen Sie im Projekt `review-template` eine neue Bereitstellung mit dem Namen `hello-secure` und dem Container-Image unter `quay.io/redhattraining/hello-world-secure:v1.0`.



### Anmerkung

Der `hello-secure`-Pod wird erst erfolgreich ausgeführt, nachdem Sie den Zugriff auf das TLS-Zertifikat und den Schlüssel für den NGINX-Server bereitgestellt haben.

- 2.1. Melden Sie sich als Benutzer `developer` bei Ihrem OpenShift-Cluster an.

```
[student@workstation review-template]$ oc login -u developer -p developer
Login successful.
...output omitted...
```

- 2.2. Erstellen Sie das Projekt `review-template`.

```
[student@workstation review-template]$ oc new-project review-template
Now using project "review-template" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

- 2.3. Listen Sie die Netzwerkrichtlinie, den Einschränkungsbereich und die Kontingentressourcen im Projekt `review-template` auf.

```
[student@workstation review-template]$ oc get \
> networkpolicy,limitrange,resourcequota
NAME
networkpolicy.networking.k8s.io/allow-from-openshift-ingress ...
networkpolicy.networking.k8s.io/allow-same-namespace ...
```

NAME	CREATED AT
limitrange/project-limitrange	2020-10-19T16:17:19Z

NAME	AGE	REQUEST
resourcequota/project-quota	37s	pods: 0/10, requests.cpu: 0/2, requests.memory: 0/1Gi

LIMIT
limits.cpu: 0/4, limits.memory: 0/4Gi

**Kapitel 9 |** Ausführliche Wiederholung

- 2.4. Überprüfen Sie, ob das Projekt `review-template` das Label `name=review-template` hat.

```
[student@workstation review-template]$ oc get project/review-template \
> --show-labels
NAME DISPLAY NAME STATUS LABELS
review-template Active name=review-template
```

- 2.5. Verwenden Sie den Befehl `oc new-app`, um die Bereitstellung `hello-secure` mit dem Container-Image `quay.io/redhattraining/hello-world-secure:v1.0` zu erstellen.

```
[student@workstation review-template]$ oc new-app --name hello-secure \
> --docker-image quay.io/redhattraining/hello-world-secure:v1.0
...output omitted...
--> Creating resources ...
 imagestream.image.openshift.io "hello-secure" created
 deployment.apps "hello-secure" created
 service "hello-secure" created
--> Success
...output omitted...
```

- 2.6. Vergewissern Sie sich, dass der Pod `hello-secure` nicht erfolgreich gestartet wird.

```
[student@workstation review-template]$ watch oc get pods
```

Drücken Sie Strg+C, um den Befehl `watch` zu beenden, nachdem der Pod mit dem Status `CrashLoopBackOff` oder `Error` angezeigt wird.

```
Every 2.0s: oc get pods

NAME READY STATUS RESTARTS AGE
hello-secure-6475f657c9-rmgsr 0/1 CrashLoopBackOff 1 14s
```

3. Erstellen Sie als Benutzer `developer` ein TLS-Secret mit dem Zertifikat `hello-secure-combined.pem` und dem Schlüssel `hello-secure-key.pem` aus dem Verzeichnis `~/D0280/labs/review-template/`. Verwenden Sie die Protokolle aus dem fehlgeschlagenen `hello-secure`-Pod, um den erwarteten Mount-Punkt für das Zertifikat zu ermitteln. Mounten Sie das TLS-Secret mit dem angegebenen Verzeichnis als Volume im Pod. Vergewissern Sie sich, dass der Pod `hello-secure` erfolgreich erneut bereitgestellt wird.

- 3.1. Erstellen Sie ein TLS-Secret mit dem Zertifikat `hello-secure-combined.pem` und dem Schlüssel `hello-secure-key.pem`.

```
[student@workstation review-template]$ oc create secret tls hello-tls \
> --cert hello-secure-combined.pem --key hello-secure-key.pem
secret/hello-tls created
```

- 3.2. Ermitteln Sie den Namen des fehlerhaften Pods.

**Kapitel 9 |** Ausführliche Wiederholung

```
[student@workstation review-template]$ oc get pods
NAME READY STATUS RESTARTS AGE
hello-secure-6475f657c9-rmgsr 0/1 CrashLoopBackOff 4 2m45s
```

- 3.3. Überprüfen Sie die Protokolle des fehlerhaften Pods. Die Protokolle deuten darauf hin, dass der Pod versucht, die Datei /run/secrets/nginx/tls.crt zu verwenden, die jedoch nicht vorhanden ist.

```
[student@workstation review-template]$ oc logs hello-secure-6475f657c9-rmgsr
...output omitted...
nginx: [emerg] BIO_new_file("/run/secrets/nginx/tls.crt") failed (SSL:
error:02001002:system library:fopen:No such file or directory:fopen('/run/
secrets/nginx/tls.crt','r') error:2006D080:BIO routines:BIO_new_file:no such file)
```

- 3.4. Verwenden Sie den Befehl `oc set volumes`, um das Secret im Verzeichnis /run/secrets/nginx zu mounten.

```
[student@workstation review-template]$ oc set volumes deployment/hello-secure \
> --add --type secret --secret-name hello-tls --mount-path /run/secrets/nginx
info: Generated volume name: volume-hlrf
deployment.apps/hello-secure volume updated
```

- 3.5. Vergewissern Sie sich, dass der Pod `hello-secure` erfolgreich erneut bereitgestellt wird.

```
[student@workstation review-template]$ watch oc get pods
```

Drücken Sie Strg+C, um den Befehl `watch` zu beenden, nachdem der Pod mit 1/1 und mit Running angezeigt wird.

```
Every 2.0s: oc get pods
...
NAME READY STATUS RESTARTS AGE
hello-secure-6bd8fccb4-nhwr2 1/1 Running 0 25s
```

4. Das Zertifikat `hello-secure-combined.pem` ist für einen einzelnen Hostnamen gültig. Verwenden Sie den Befehl `openssl x509` mit den Optionen `-noout` und `-ext 'subjectAltName'`, um das Zertifikat `hello-secure-combined.pem` zu lesen und den Hostnamen zu ermitteln. Erstellen Sie als Benutzer `developer` eine Passthrough-Route zum `hello-secure`-Service mit dem identifizierten Hostnamen. Überprüfen Sie, ob die Route auf externe Anforderungen reagiert.

**Anmerkung**

Die Manpage `x509 (1)` enthält Informationen zum `openssl x509`-Befehl.

- 4.1. Untersuchen Sie das Zertifikat `hello-secure-combined.pem` mit dem Befehl `openssl x509`.

```
[student@workstation review-template]$ openssl x509 \
> -in hello-secure-combined.pem -noout -ext 'subjectAltName'
X509v3 Subject Alternative Name:
DNS:hello-secure.apps.ocp4.example.com
```

- 4.2. Erstellen Sie eine Passthrough-Route zum hello-secure-Service, der auf hello-secure.apps.ocp4.example.com verweist.

```
[student@workstation review-template]$ oc create route passthrough \
> --service hello-secure --hostname hello-secure.apps.ocp4.example.com
route.route.openshift.io/hello-secure created
```

- 4.3. Kehren Sie zum Verzeichnis /home/student/ zurück.

```
[student@workstation review-template]$ cd
```

Überprüfen Sie mit dem Befehl curl, ob die Route auf externe Anfragen reagiert.

```
[student@workstation ~]$ curl -s https://hello-secure.apps.ocp4.example.com \
> | grep Hello
<h1>Hello, world from nginx!</h1>
```

5. Konfigurieren Sie als Benutzer developer die hello-secure-Bereitstellung so, dass sie automatisch skaliert wird. In der Bereitstellung muss mindestens ein Pod ausgeführt werden. Wenn die durchschnittliche CPU-Auslastung 80 % überschreitet, wird die Bereitstellung auf bis zu fünf Pods skaliert.



#### Anmerkung

Mit dem Skript unter ~/D0280/solutions/review-template/test-hpa.sh können Sie testen, ob Ihre Bereitstellung erwartungsgemäß skaliert wird.

- 5.1. Verwenden Sie den Befehl oc autoscale, um die horizontale Pod-autoscaler-Ressource zu erstellen.

```
[student@workstation ~]$ oc autoscale deployment/hello-secure \
> --min 1 --max 5 --cpu-percent 80
horizontalpodautoscaler.autoscaling/hello-secure autoscaled
```

- 5.2. Führen Sie das bereitgestellte Skript test-hpa.sh aus. Das Skript verwendet den Befehl ab, um mit dem Apache-Benchmarking-Tool Last zu generieren.

```
[student@workstation ~]$ ~/D0280/solutions/review-template/test-hpa.sh
...output omitted...
Benchmarking hello-secure.apps.ocp4.example.com (be patient)
Completed 10000 requests
Completed 20000 requests
...output omitted...
Finished 100000 requests
...output omitted...
```

**Kapitel 9 |** Ausführliche Wiederholung

- 5.3. Überprüfen Sie, ob mindestens zwei, jedoch nicht mehr als fünf hello-secure-Pods ausgeführt werden.

```
[student@workstation ~]$ oc get pods
NAME READY STATUS RESTARTS AGE
hello-secure-6bd8fcccb4-7qjc2 1/1 Running 0 96s
hello-secure-6bd8fcccb4-d67xd 1/1 Running 0 66s
hello-secure-6bd8fcccb4-m8vxp 1/1 Running 0 96s
hello-secure-6bd8fcccb4-nhwr2 1/1 Running 0 9m36s
```

## 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.

```
[student@workstation ~]$ lab review-template grade
```

## Beenden

Führen Sie auf dem Rechner `workstation` als Benutzer `student` den Befehl `lab` aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab review-template finish
```

Hiermit ist die praktische Übung beendet.