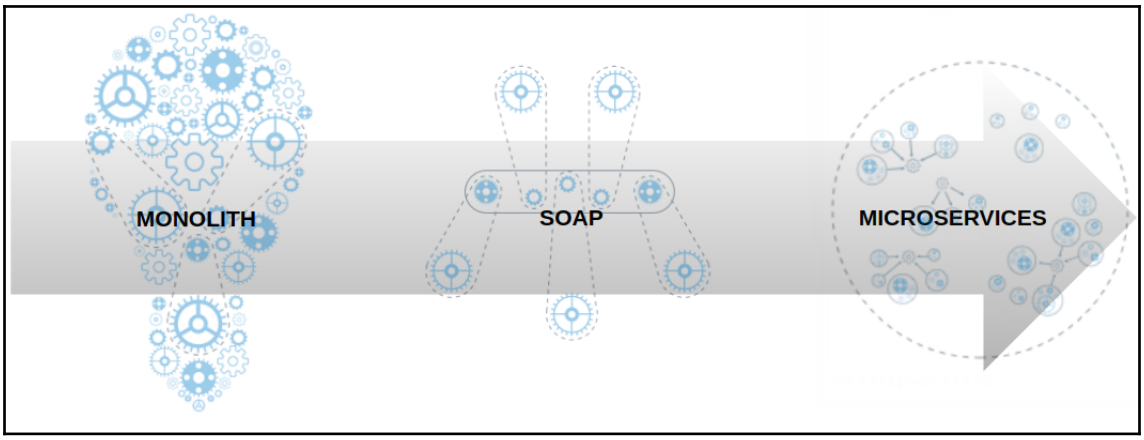
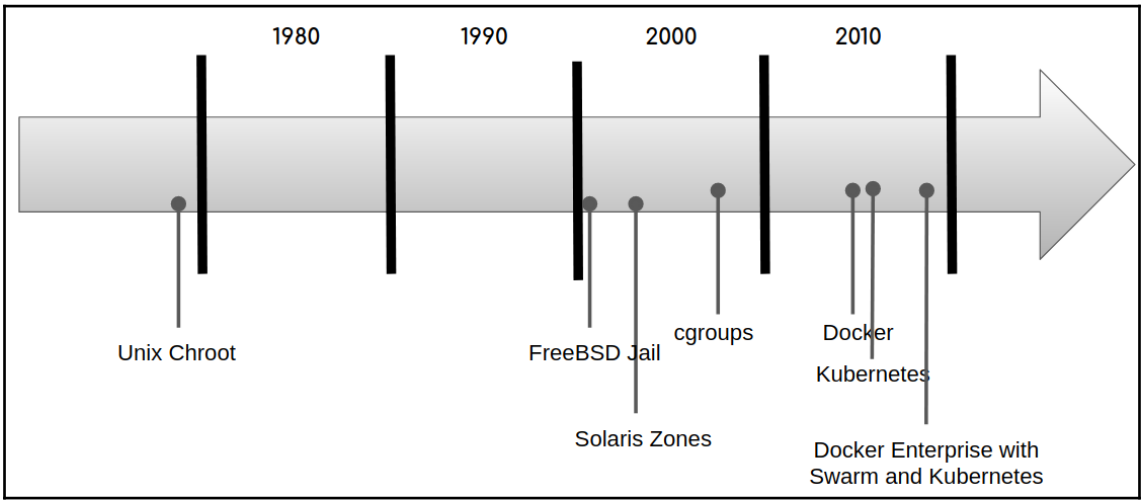
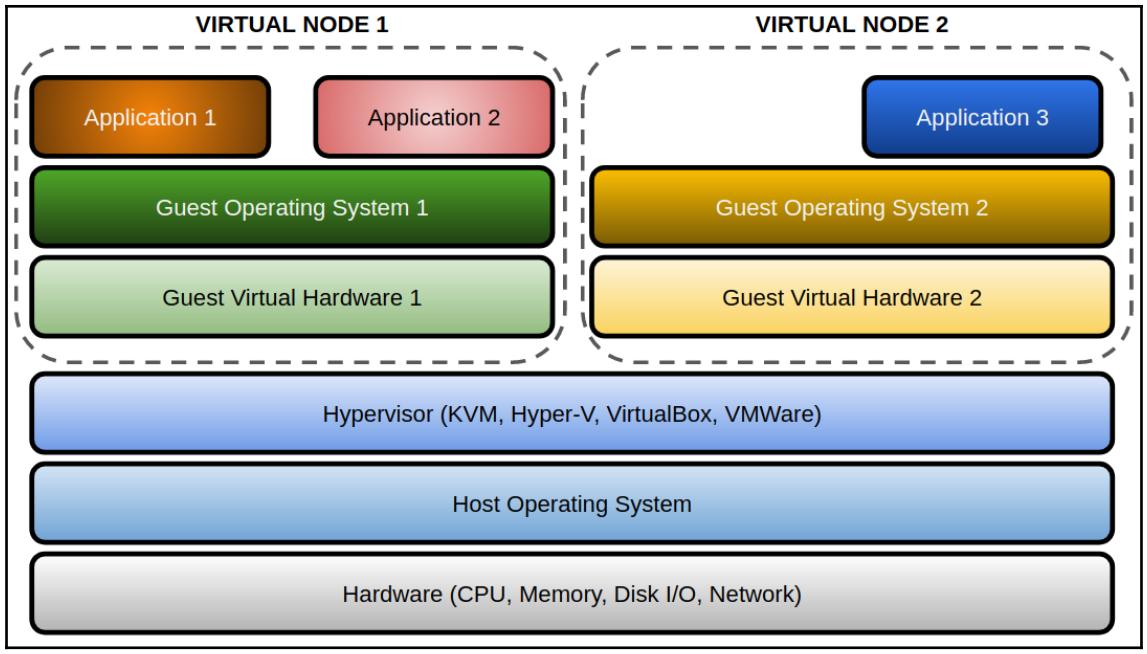
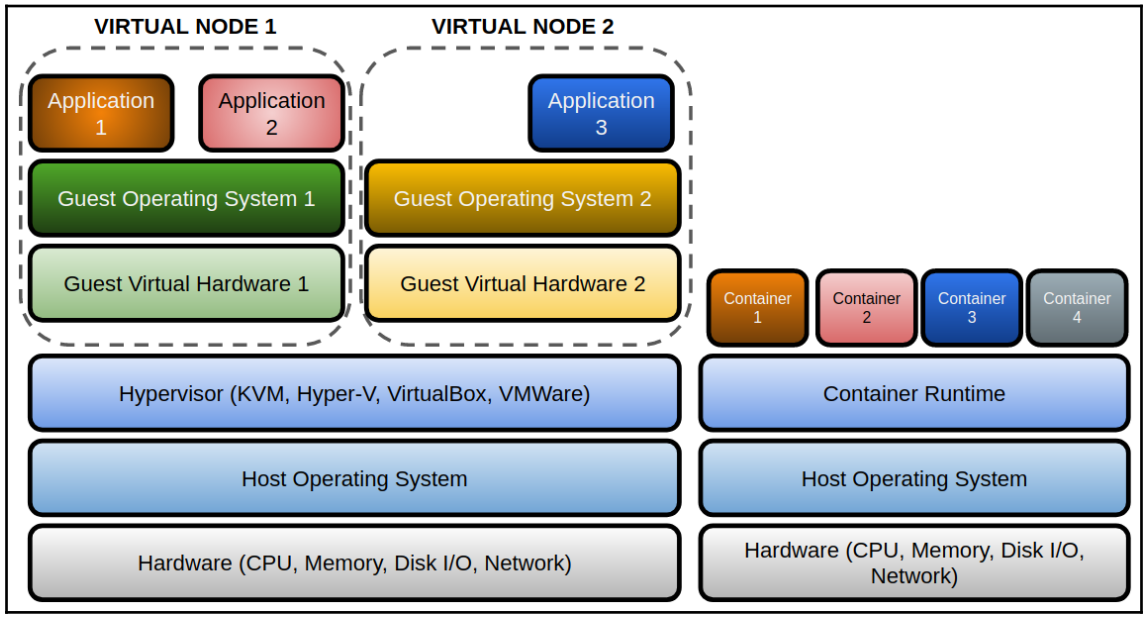
Graphics Bundle

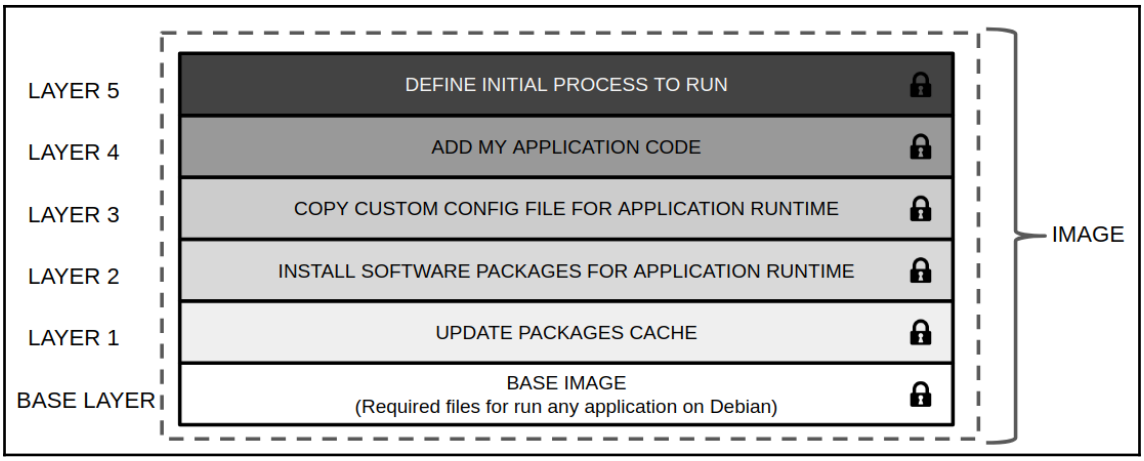
Chapter 1: Modern Infrastructures and Applications with Docker

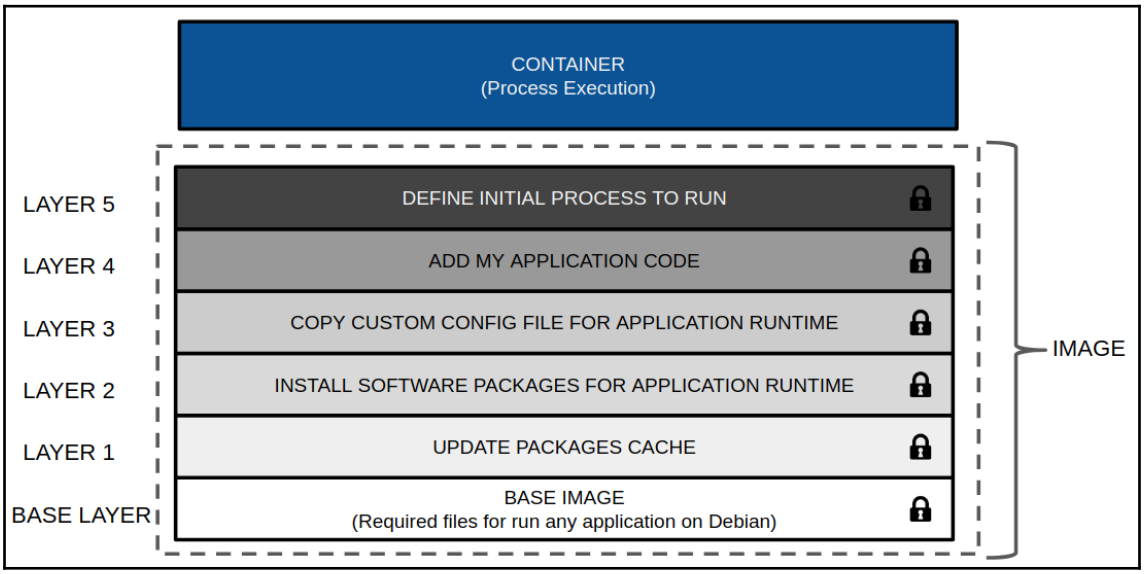


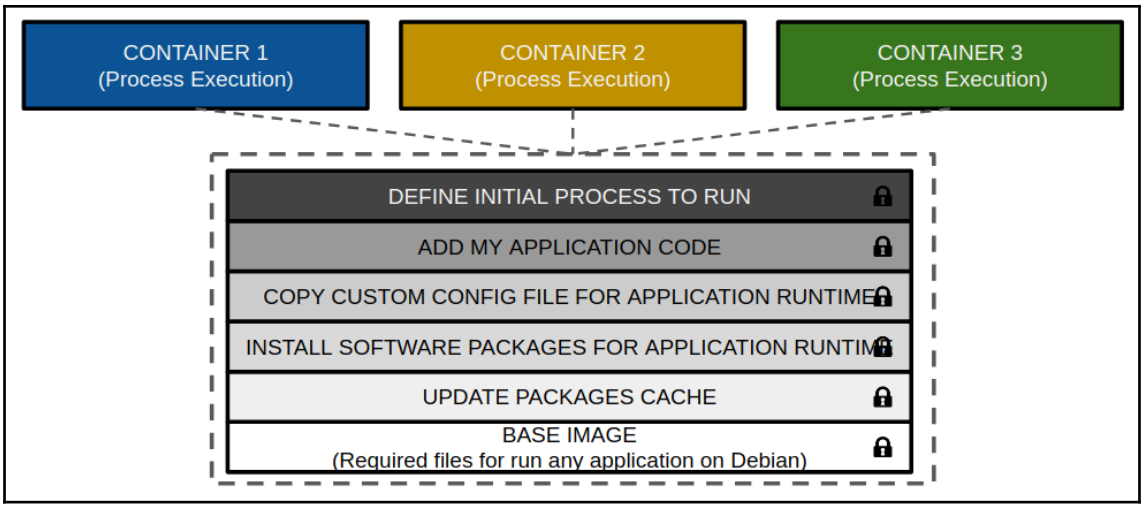


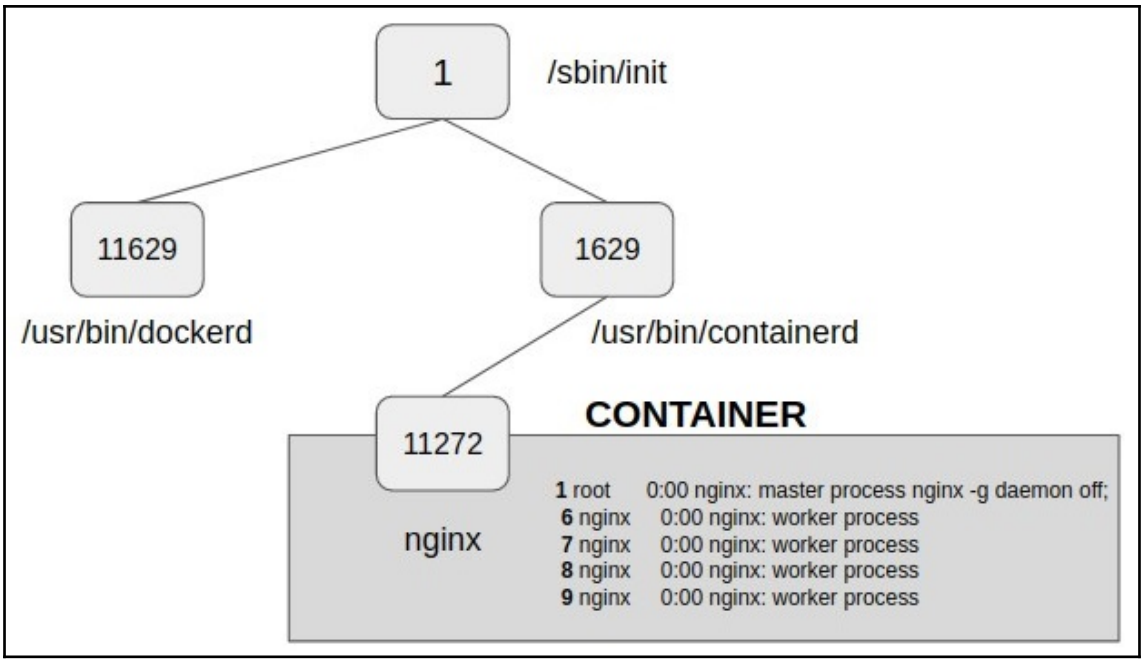

VIRTUAL NODE 1
VIRTUAL NODE 2
Application 1
Application 2
Application 3
Guest Operating System 1
Guest Operating System 2
Guest Virtual Hardware 1
Guest Virtual Hardware 2
Hypervisor (KVM, Hyper-V, VirtualBox, VMWare)
Host Operating System
Hardware (CPU, Memory, Disk I/O, Network)

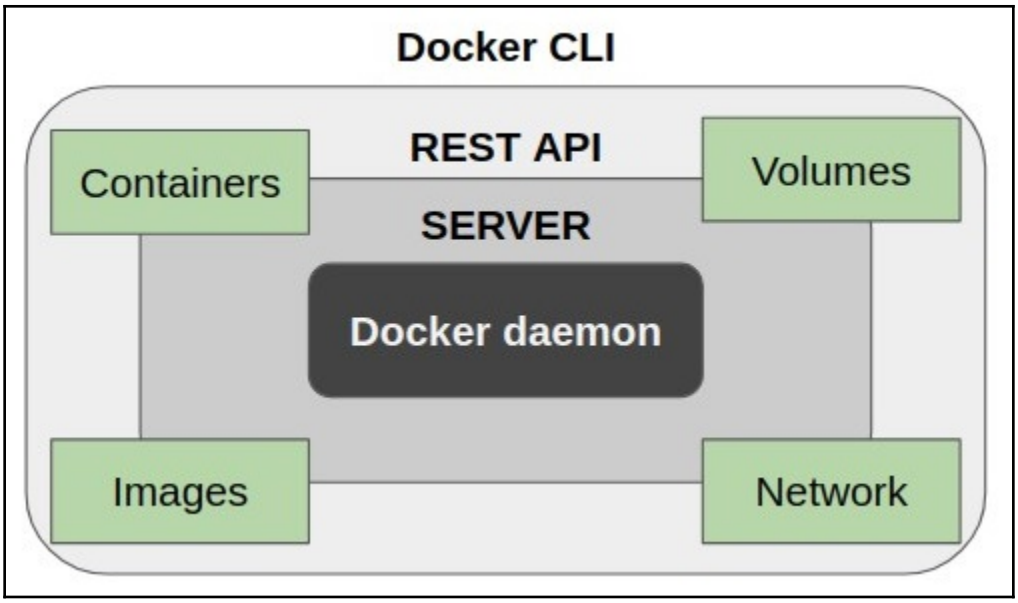

VIRTUAL NODE 1
VIRTUAL NODE 2
Application 1
Application 2
Application 3
Guest Operating System 1
Guest Operating System 2
Guest Virtual Hardware 1
Guest Virtual Hardware 2
Container 1
Container 2
Container 3
Container 4
Hypervisor (KVM, Hyper-V, VirtualBox, VMWare)
Container Runtime
Host Operating System
Host Operating System
Hardware (CPU, Memory, Disk I/O, Network)
Hardware (CPU, Memory, Disk I/O, Network)

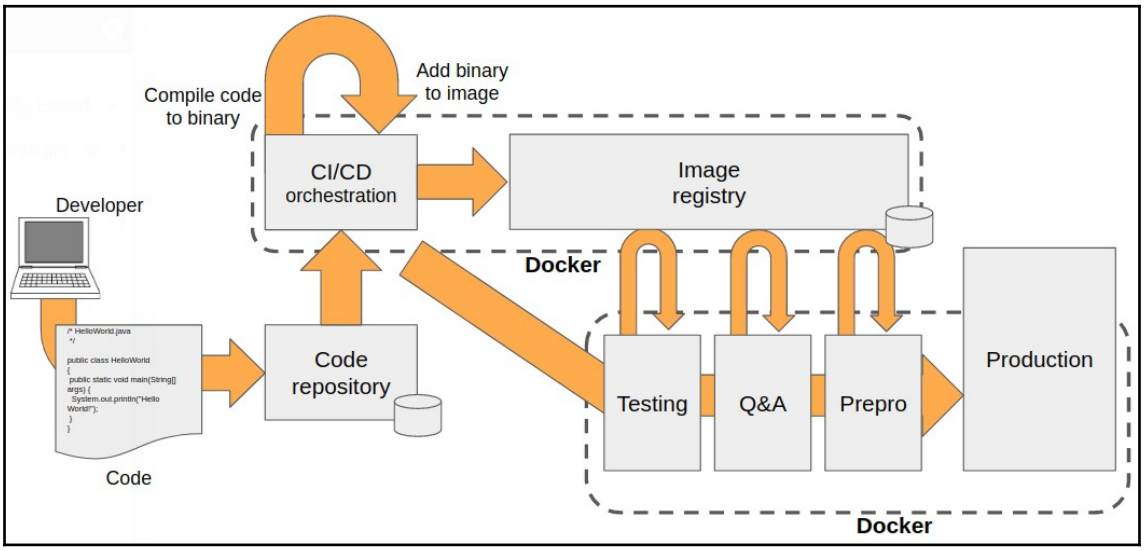

LAYER 5
LAYER 4
LAYER 3
LAYER 2
LAYER 1
BASE LAYER
DEFINE INITIAL PROCESS TO RUN
ADD MY APPLICATION CODE
COPY CUSTOM CONFIG FILE FOR APPLICATION RUNTIME
INSTALL SOFTWARE PACKAGES FOR APPLICATION RUNTIME
UPDATE PACKAGES CACHE
BASE IMAGE
(Required files for run any application on Debian)
IMAGE

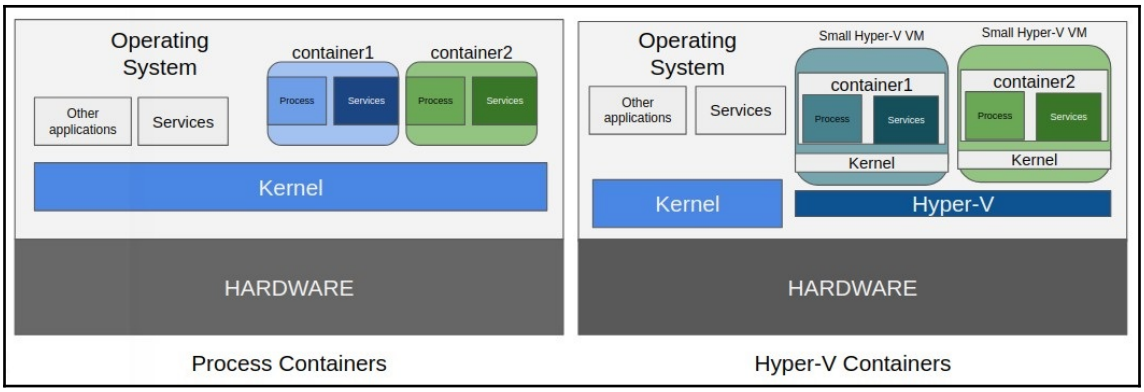

CONTAINER
(Process Execution)
LAYER 5
LAYER 4
LAYER 3
LAYER 2
LAYER 1
BASE LAYER
DEFINE INITIAL PROCESS TO RUN
ADD MY APPLICATION CODE
COPY CUSTOM CONFIG FILE FOR APPLICATION RUNTIME
INSTALL SOFTWARE PACKAGES FOR APPLICATION RUNTIME
UPDATE PACKAGES CACHE
BASE IMAGE
(Required files for run any application on Debian)
IMAGE


CONTAINER 1
(Process Execution)
CONTAINER 2
(Process Execution)
CONTAINER 3
(Process Execution)
DEFINE INITIAL PROCESS TO RUN
ADD MY APPLICATION CODE
COPY CUSTOM CONFIG FILE FOR APPLICATION RUNTIME
INSTALL SOFTWARE PACKAGES FOR APPLICATION RUNTIME
UPDATE PACKAGES CACHE
BASE IMAGE
(Required files for run any application on Debian)


1
/sbin/init
11629
/usr/bin/dockerd
1629
/usr/bin/containerd
11272
nginx
CONTAINER
1 root 0:00 nginx: master process nginx -g daemon off;
6 nginx 0:00 nginx: worker process
7 nginx 0:00 nginx: worker process
8 nginx 0:00 nginx: worker process
9 nginx 0:00 nginx: worker process


Docker CLI
REST API
SERVER
Containers
Volumes
Docker daemon
Images
Network

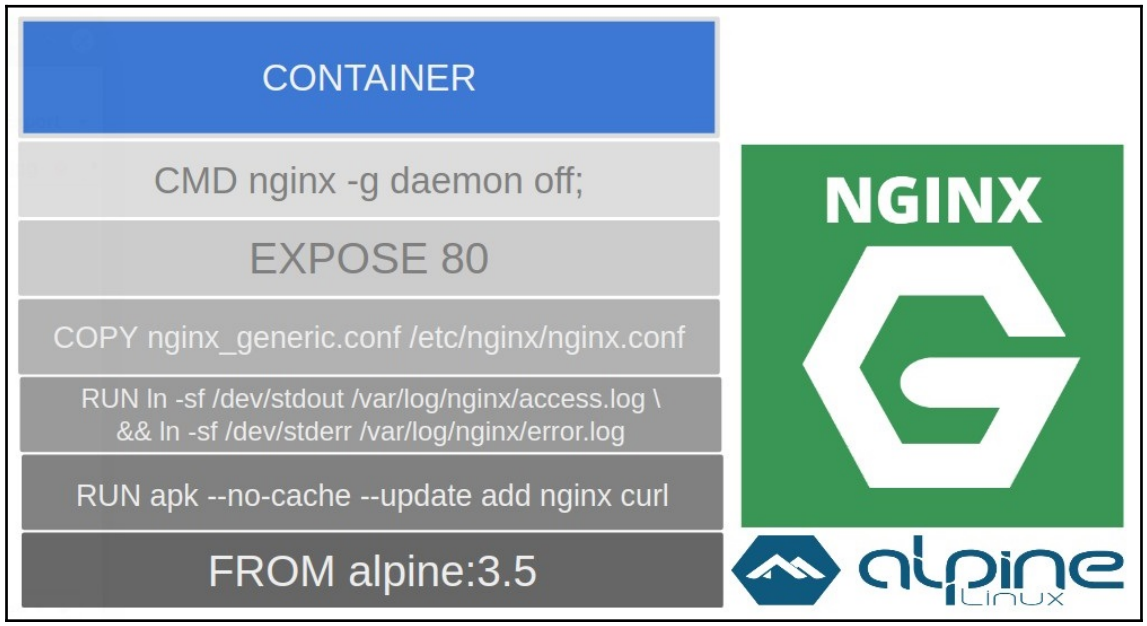

Compile code
to binary
Add binary
to image
CI/CD
orchestration
Image
registry
Developer
Docker
Code
repository
Testing
Q&A
Prepro
Production
Code
Docker


Operating System
Other applications
Services
container1
Process
Services
container2
Process
Services
Kernel
HARDWARE
Process Containers
Operating System
Other applications
Services
Small Hyper-V VM
container1
Process
Services
Kernel
Small Hyper-V VM
container2
Process
Services
Kernel
Kernel
Hyper-V
HARDWARE
Hyper-V Containers

Chapter 2: Building Docker Images

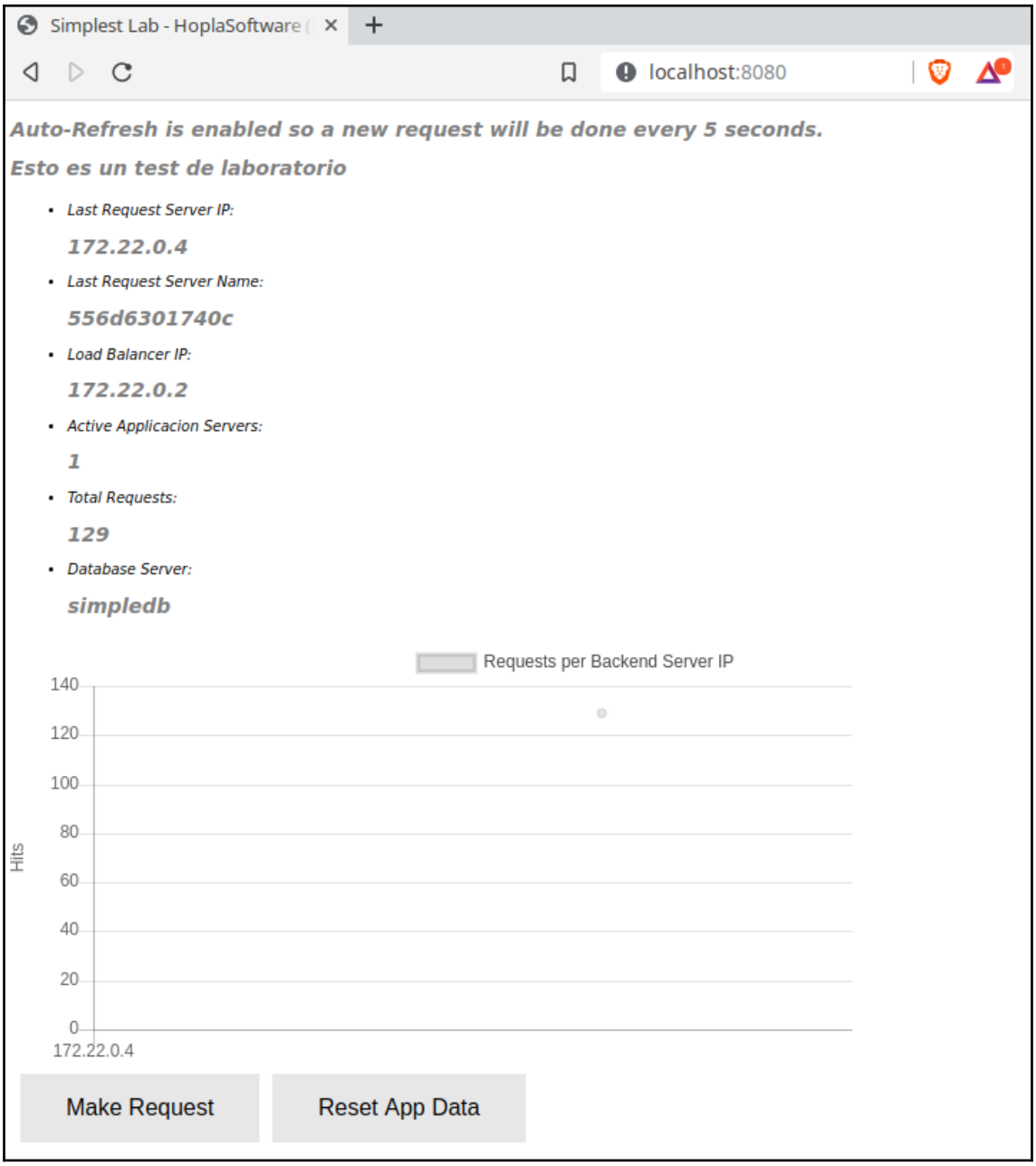
There is an image action that provides a detailed review of the steps to create an image. `docker image history` will provide a historic view of the steps that were taken to create that image. However, it will not work on images that are created using committed containers. We will just have a line with a bash, for example, indicating that all the actions that were taken were made on an active container and therefore, no additional information can be extracted. For example, using the previously created image, executing `docker image history debian-with-postfix` will provide the following output:

```
zero@sirius:~$ docker image history debian-with-postfix
IMAGE               CREATED             CREATED BY                                      SIZE
a852d20d57c9        2 minutes ago       bash                                            96.4MB
67e34c1c9477        4 weeks ago         /bin/sh -c #(nop)  CMD ["bash"]                 0B
<missing>           4 weeks ago         /bin/sh -c #(nop) ADD file:9b7d9295bf7e8307b…   114MB
```

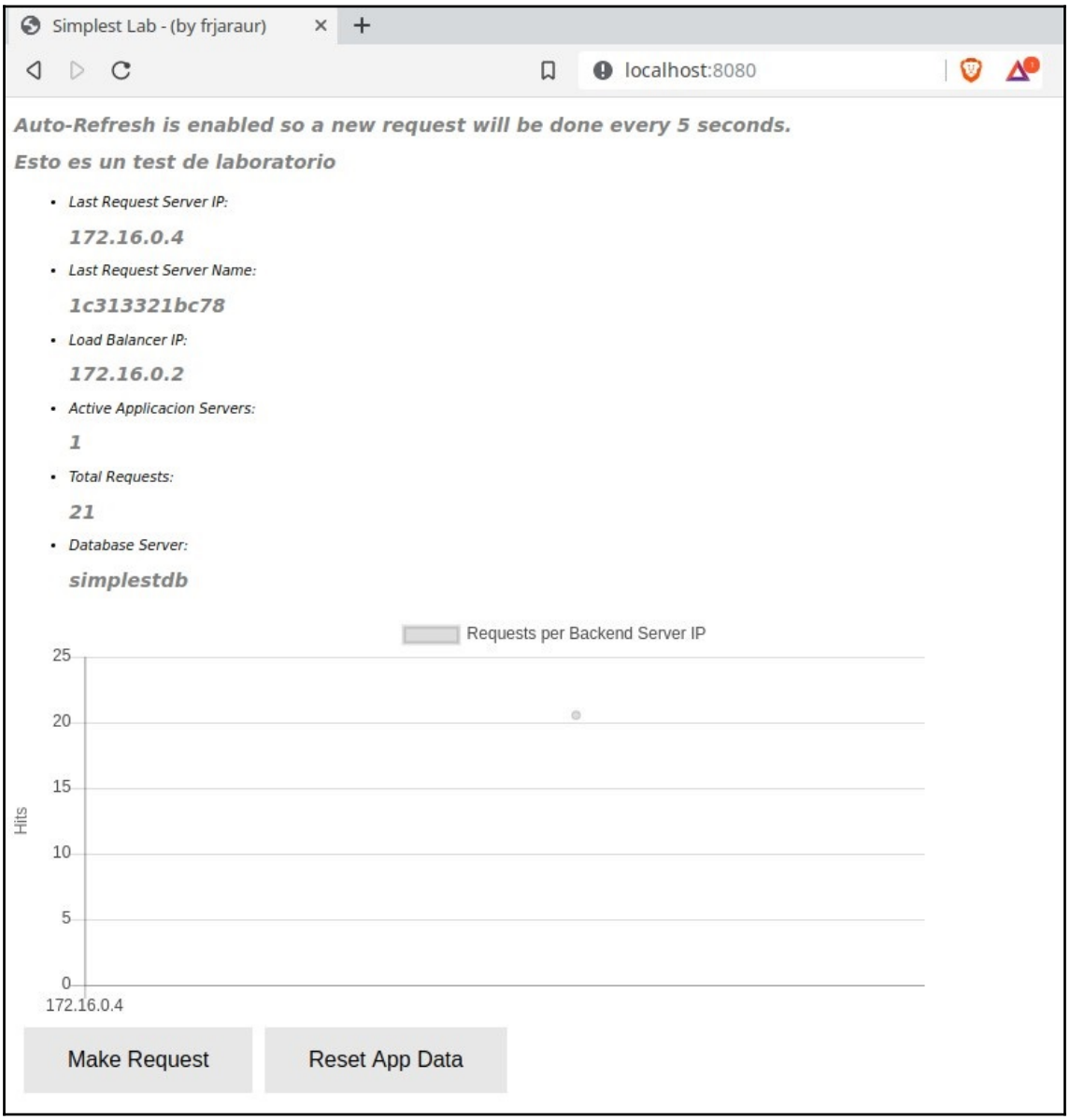


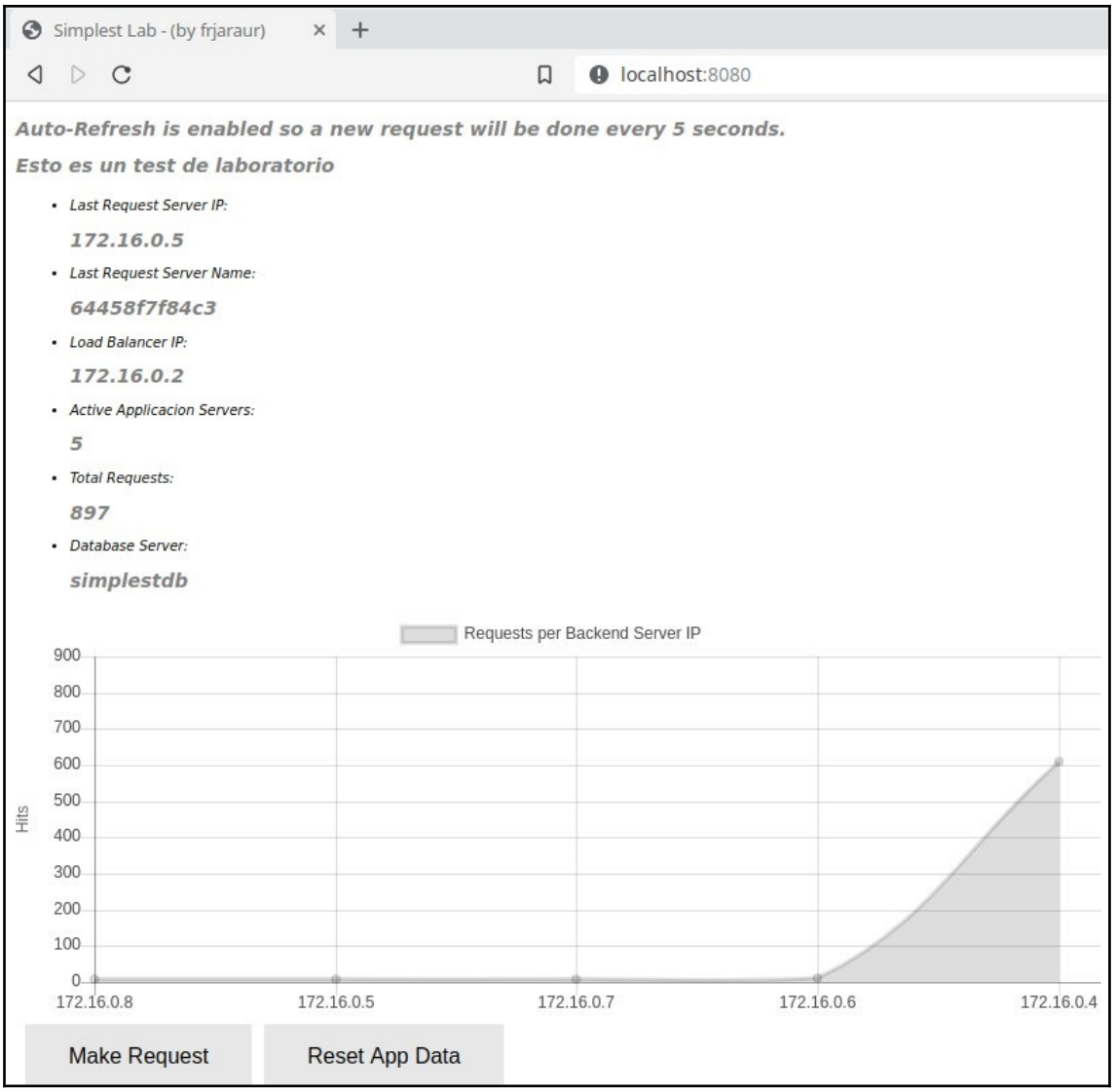
Chapter 3: Running Docker Containers

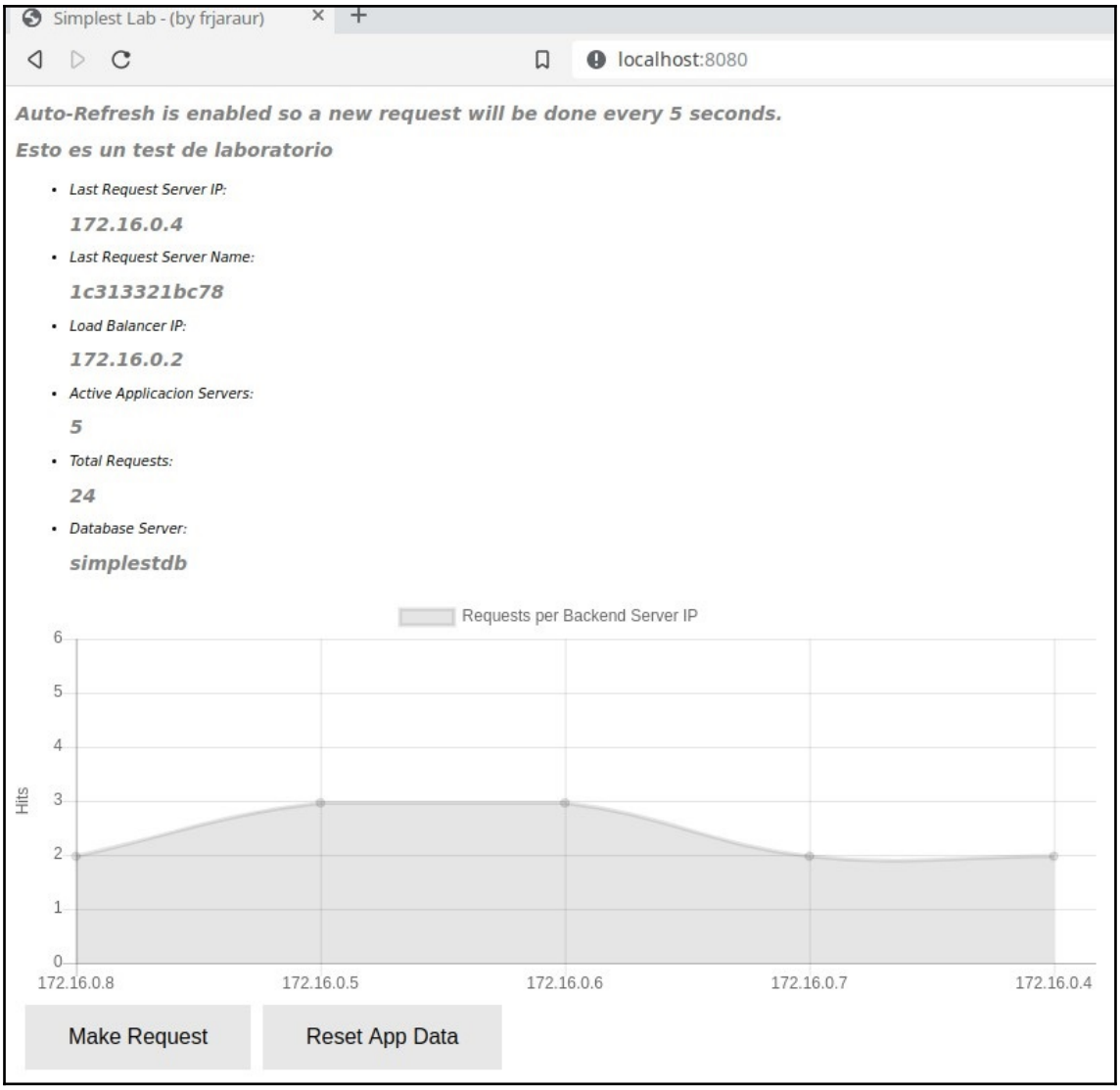
Chapter 4: Container Persistency and Networking

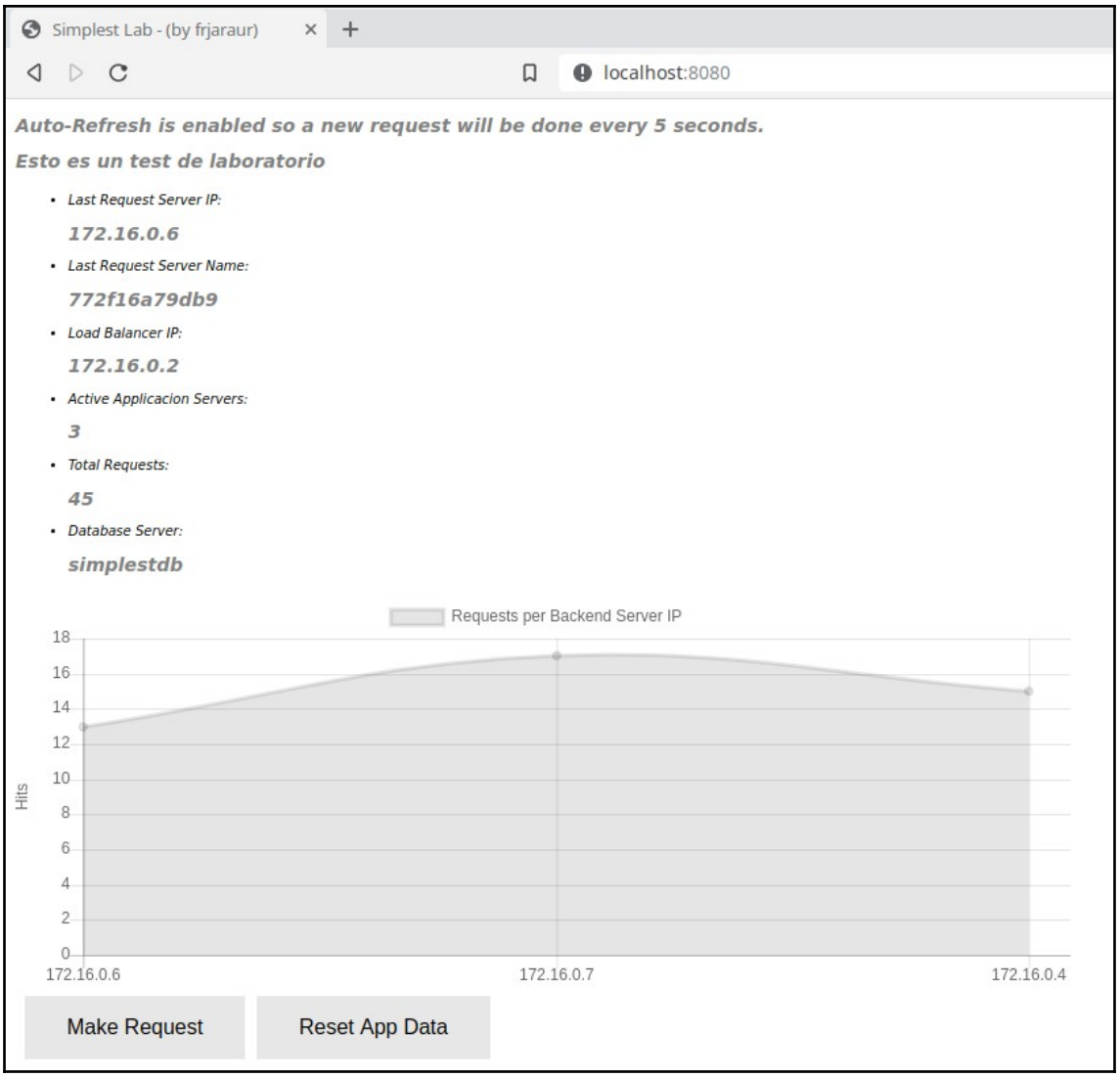

Simplest Lab - HoplaSoftware
localhost:8080
Auto-Refresh is enabled so a new request will be done every 5 seconds.
Esto es un test de laboratorio
Last Request Server IP:
172.22.0.4
Last Request Server Name:
556d6301740c
Load Balancer IP:
172.22.0.2
Active Applicacion Servers:
1
Total Requests:
129
Database Server:
simpledb
Requests per Backend Server IP
Hits
140
120
100
80
60
40
20
0
172.22.0.4
Make Request
Reset App Data

Chapter 5: Deploying Multi-Container Applications

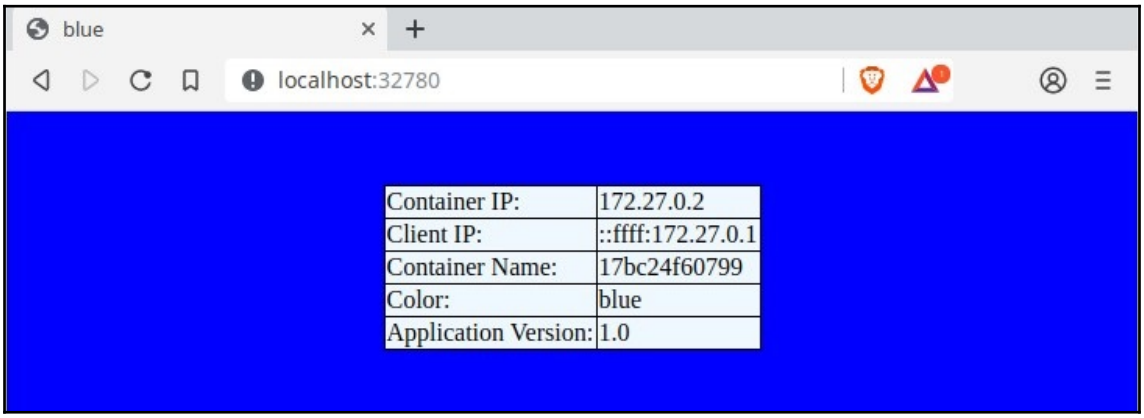



Simplest Lab - (by frjaraur)
localhost:8080
Auto-Refresh is enabled so a new request will be done every 5 seconds.
Esto es un test de laboratorio
Last Request Server IP:
172.16.0.5
Last Request Server Name:
64458f7f84c3
Load Balancer IP:
172.16.0.2
Active Applicacion Servers:
5
Total Requests:
897
Database Server:
simplestdb
Requests per Backend Server IP
Hits
900
800
700
600
500
400
300
200
100
0
172.16.0.8
172.16.0.5
172.16.0.7
172.16.0.6
172.16.0.4
Make Request
Reset App Data

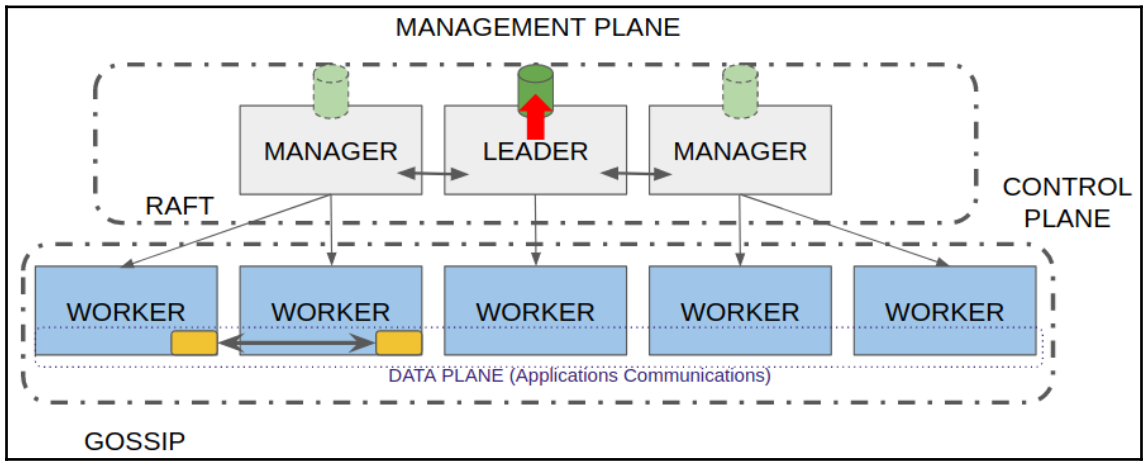

Simplest Lab - (by frjaraur)
localhost:8080
Auto-Refresh is enabled so a new request will be done every 5 seconds.
Esto es un test de laboratorio
Last Request Server IP:
172.16.0.4
Last Request Server Name:
1c313321bc78
Load Balancer IP:
172.16.0.2
Active Applicacion Servers:
5
Total Requests:
24
Database Server:
simplestdb
Requests per Backend Server IP
Hits
0
1
2
3
4
5
6
172.16.0.8
172.16.0.5
172.16.0.6
172.16.0.7
172.16.0.4
Make Request
Reset App Data

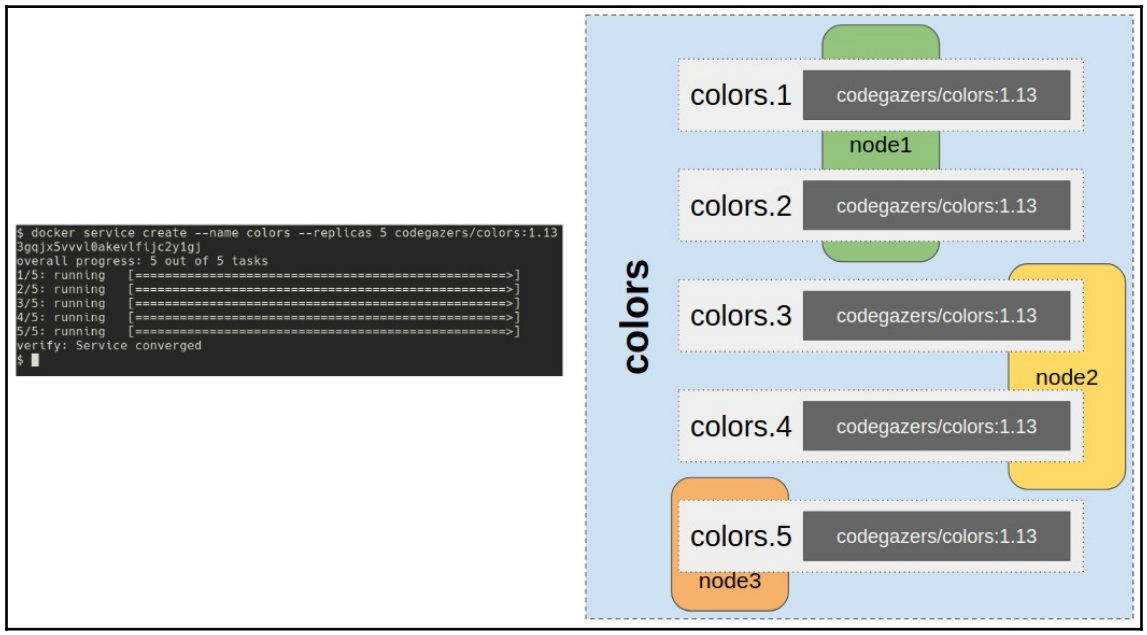

Simplest Lab - (by frjaraur)
localhost:8080
Auto-Refresh is enabled so a new request will be done every 5 seconds.
Esto es un test de laboratorio
Last Request Server IP:
172.16.0.6
Last Request Server Name:
772f16a79db9
Load Balancer IP:
172.16.0.2
Active Applicacion Servers:
3
Total Requests:
45
Database Server:
simplestdb
Requests per Backend Server IP
Hits
18
16
14
12
10
8
6
4
2
0
172.16.0.6
172.16.0.7
172.16.0.4
Make Request
Reset App Data

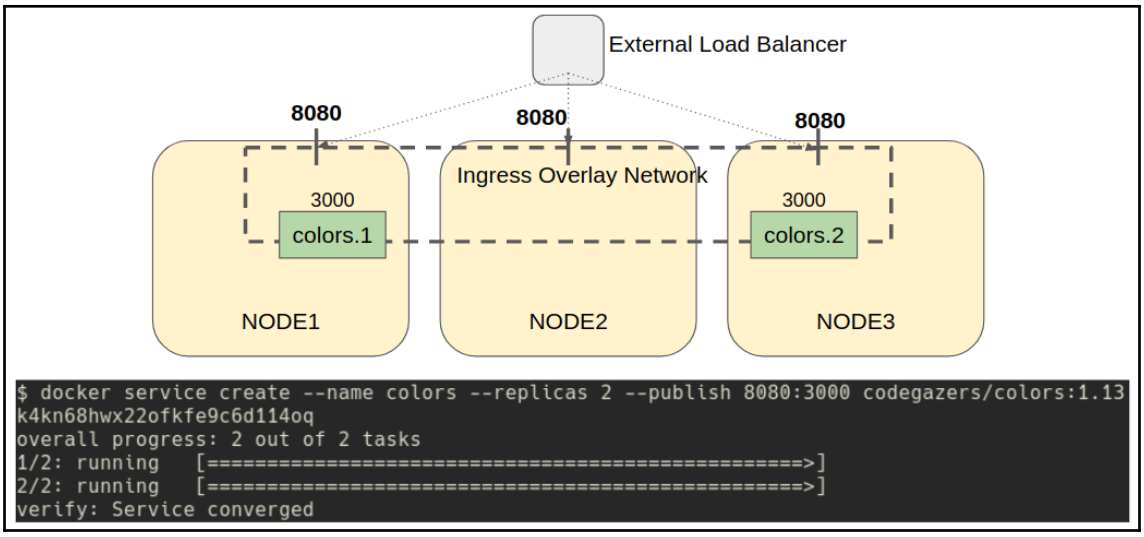
```
$ docker container ls --filter name=^/simplest* --format "table {{ .Names }}\t {{.Labels}}\n\n"
NAMES               LABELS


simplest-lab_app_3    com.docker.compose.version=1.24.0,com.docker.compose.config-hash=9ea63f358f0d5376566073c929b0d487d8c1c3d7dccc83506c8d9b042f9007fa,com.docker.compose.container-number=3,com.docker.compose.oneoff=False,com.docker.compose.project=simplest-lab,com.docker.compose.service=app


simplest-lab_app_2    com.docker.compose.container-number=2,com.docker.compose.oneoff=False,com.docker.compose.project=simplest-lab,com.docker.compose.service=app,com.docker.compose.version=1.24.0,com.docker.compose.config-hash=9ea63f358f0d5376566073c929b0d487d8c1c3d7dccc83506c8d9b042f9007fa


simplest-lab_app_1    com.docker.compose.oneoff=False,com.docker.compose.project=simplest-lab,com.docker.compose.service=app,com.docker.compose.version=1.24.0,com.docker.compose.config-hash=9ea63f358f0d5376566073c929b0d487d8c1c3d7dccc83506c8d9b042f9007fa,com.docker.compose.container-number=1


simplest-lab_lb_1    com.docker.compose.container-number=1,com.docker.compose.oneoff=False,com.docker.compose.project=simplest-lab,com.docker.compose.service=lb,com.docker.compose.version=1.24.0,com.docker.compose.config-hash=fc6feb3cf80fac4d71a389bc8187fce64045ac3ed6c7da4413c9a7043c2088f8


simplest-lab_db_1    com.docker.compose.version=1.24.0,com.docker.compose.config-hash=827c1a8becc9b2b4337d04dcfe2b01115da440ece3a6a2a974db41c5d2d25ad5,com.docker.compose.container-number=1,com.docker.compose.oneoff=False,com.docker.compose.project=simplest-lab,com.docker.compose.service=db
```



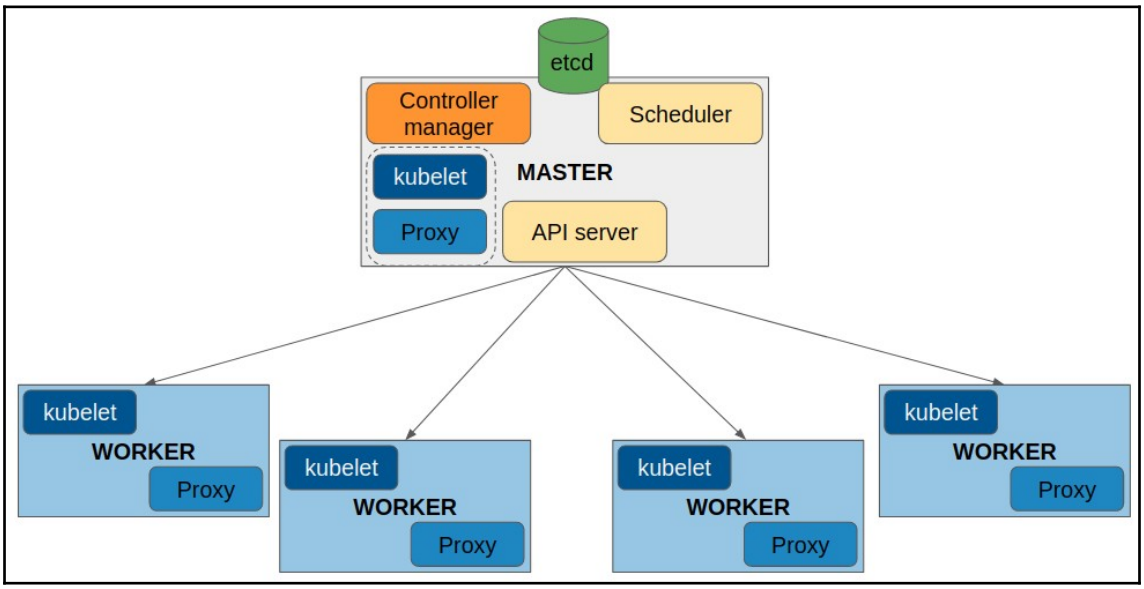
Chapter 8: Orchestration Using Docker Swarm

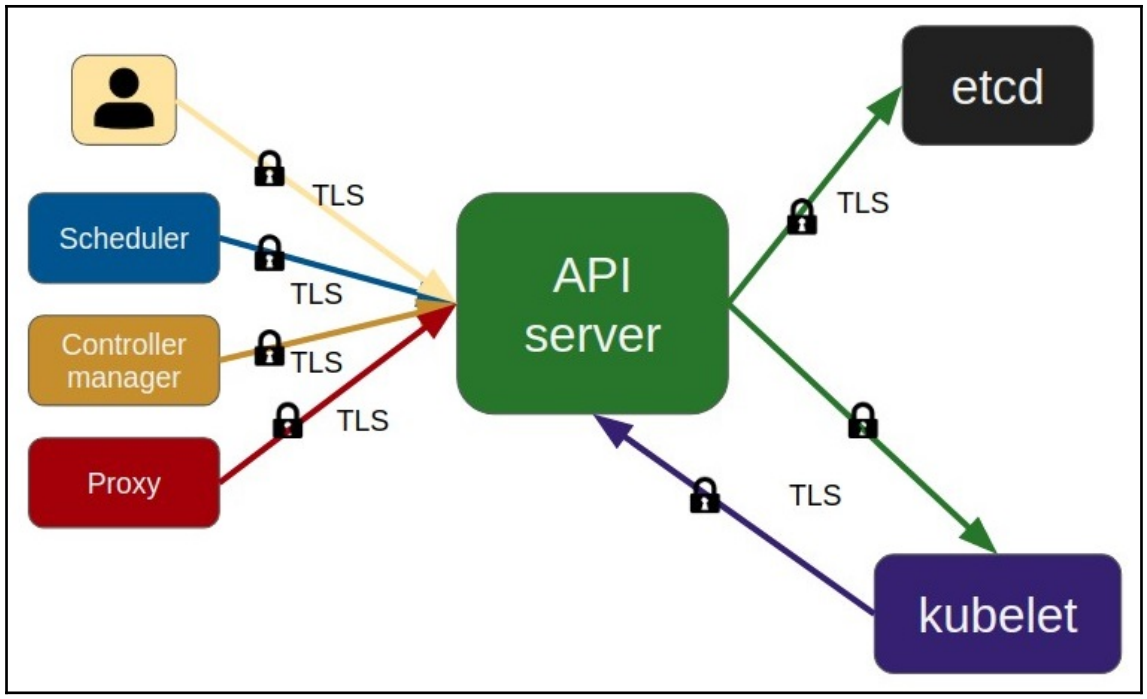


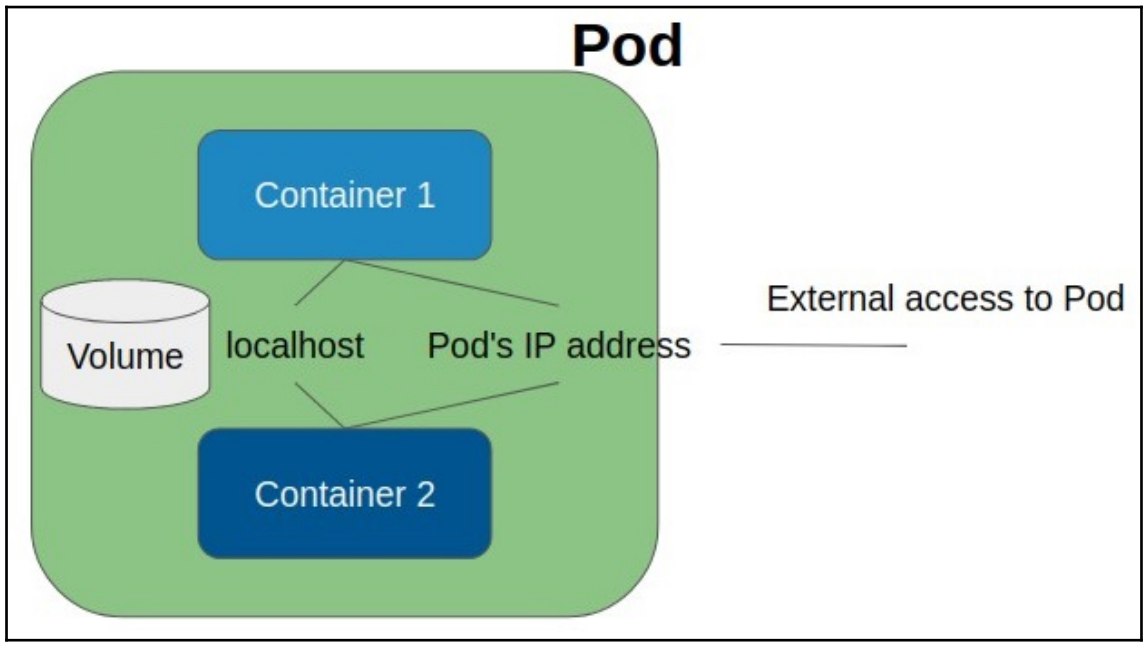


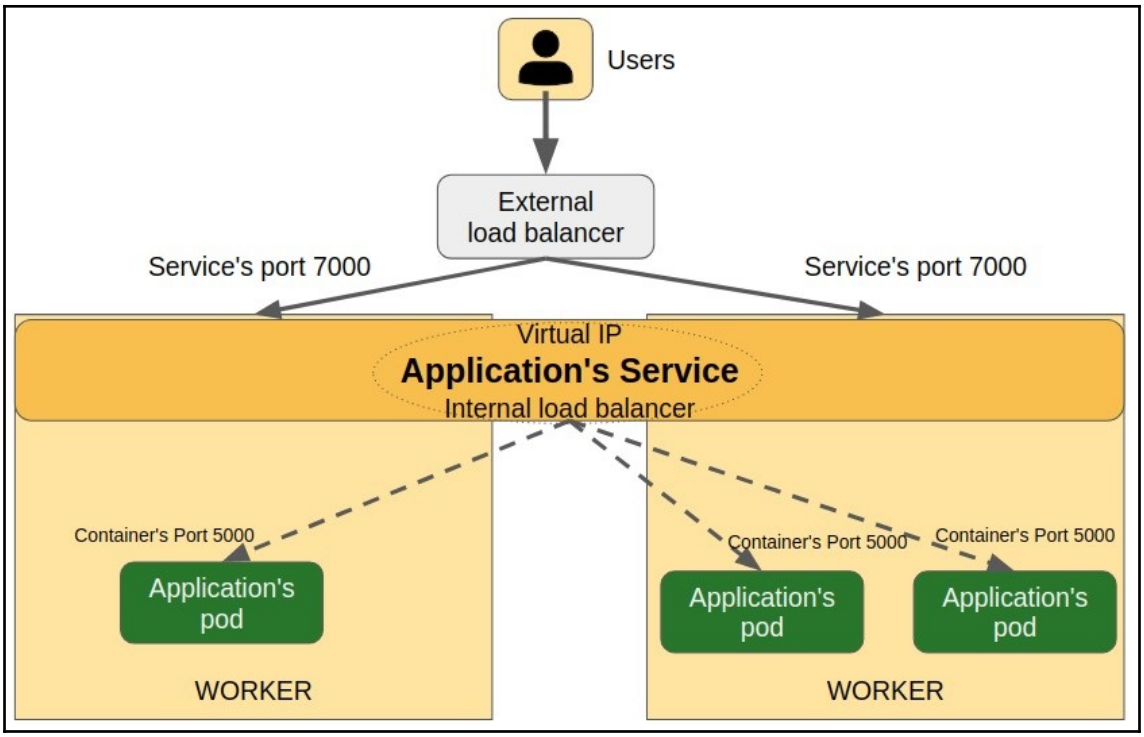

External Load Balancer
8080
8080
8080
Ingress Overlay Network
3000
colors.1
3000
colors.2
NODE1
NODE2
NODE3
$ docker service create --name colors --replicas 2 --publish 8080:3000 codegazers/colors:1.13
k4kn68hwx22ofkfe9c6d114oq
overall progress: 2 out of 2 tasks
1/2: running [==>]
2/2: running [==>]
verify: Service converged

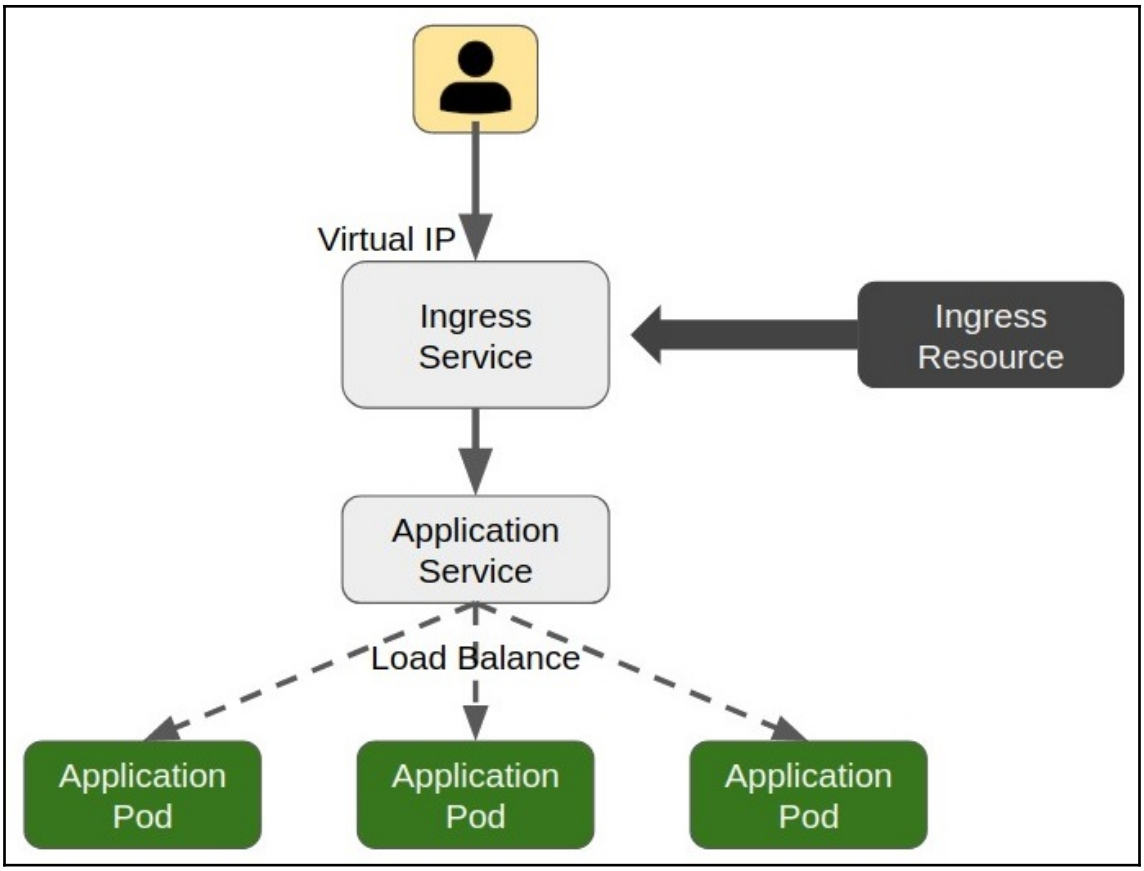
Chapter 9: Orchestration Using Kubernetes



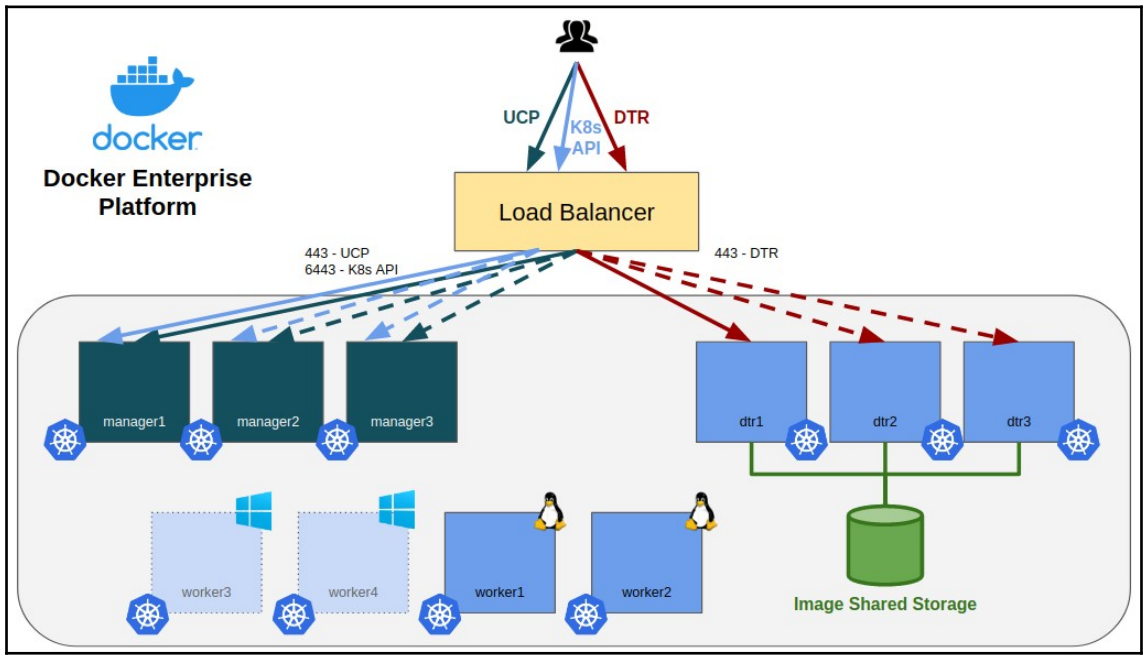

etcd
TLS
Scheduler
TLS
API server
TLS
Controller manager
TLS
TLS
Proxy
TLS
kubelet


Pod
Container 1
Volume
localhost
Pod's IP address
External access to Pod
Container 2

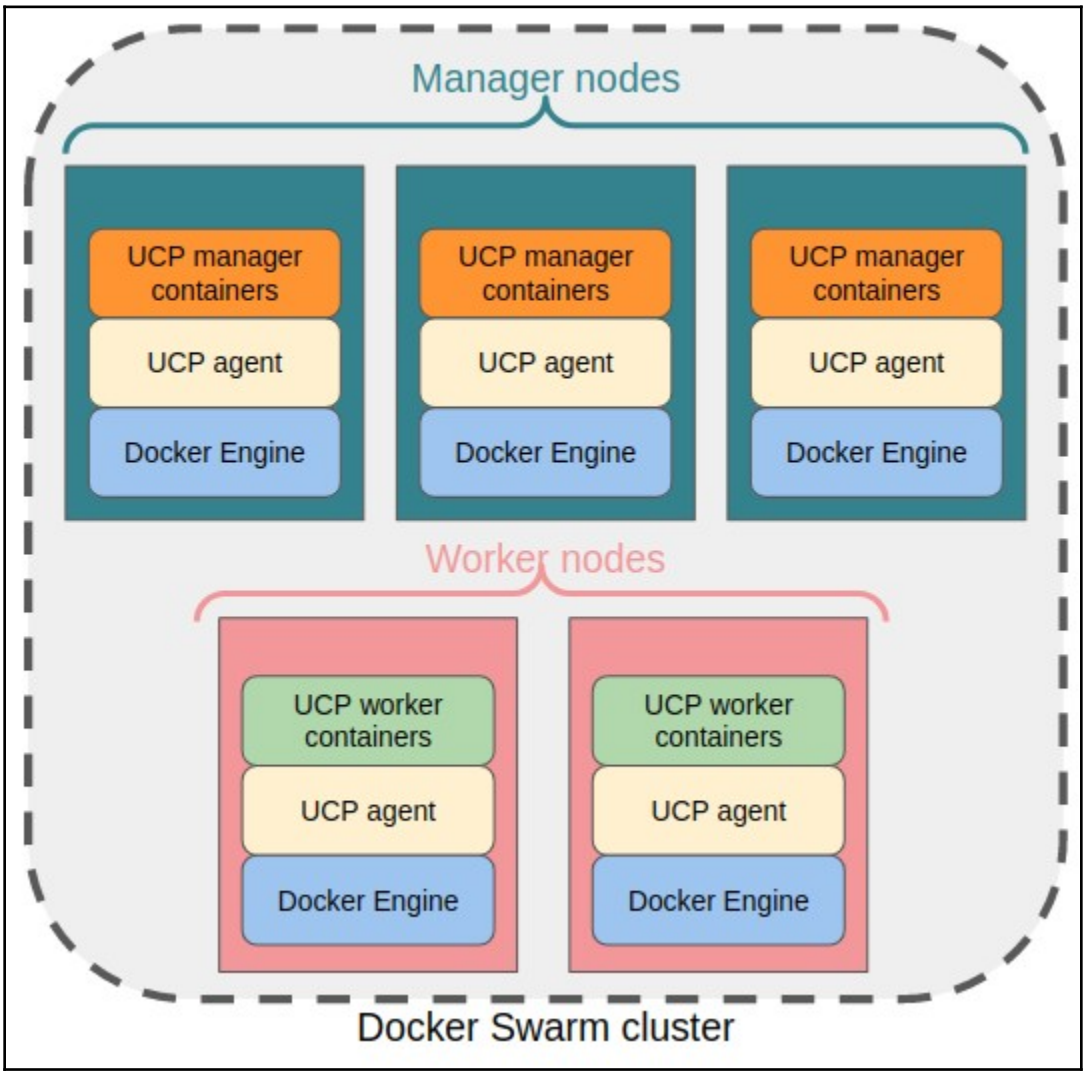

Users
External
load balancer
Service's port 7000
Service's port 7000
Virtual IP
Application's Service
Internal load balancer
Container's Port 5000
Application's
pod
WORKER
Container's Port 5000
Application's
pod
Container's Port 5000
Application's
pod
WORKER

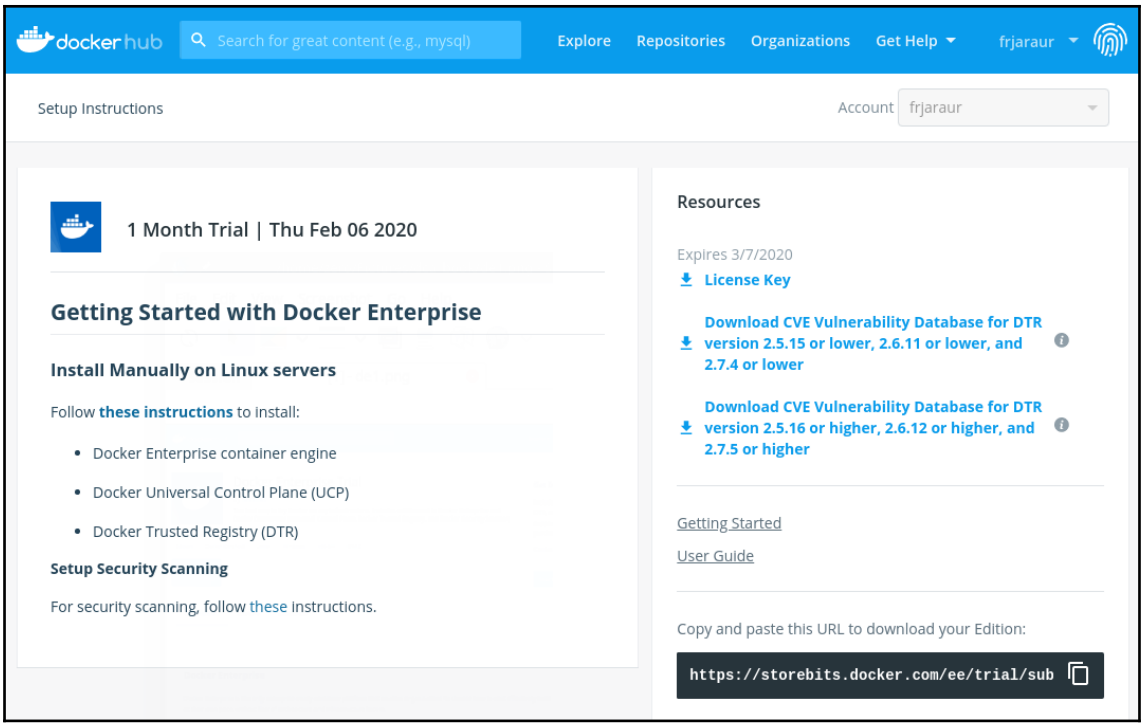

Virtual IP
Ingress Service
Ingress Resource
Application Service
Load Balance
Application Pod
Application Pod
Application Pod

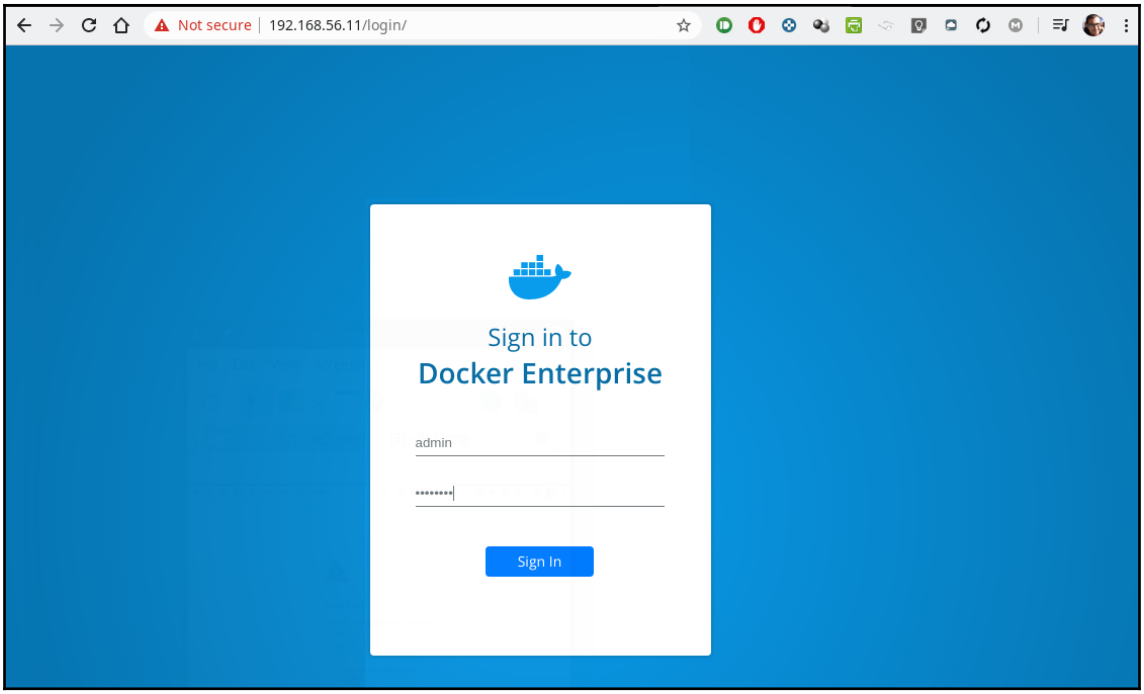
Chapter 10: Introduction to the Docker Enterprise Platform

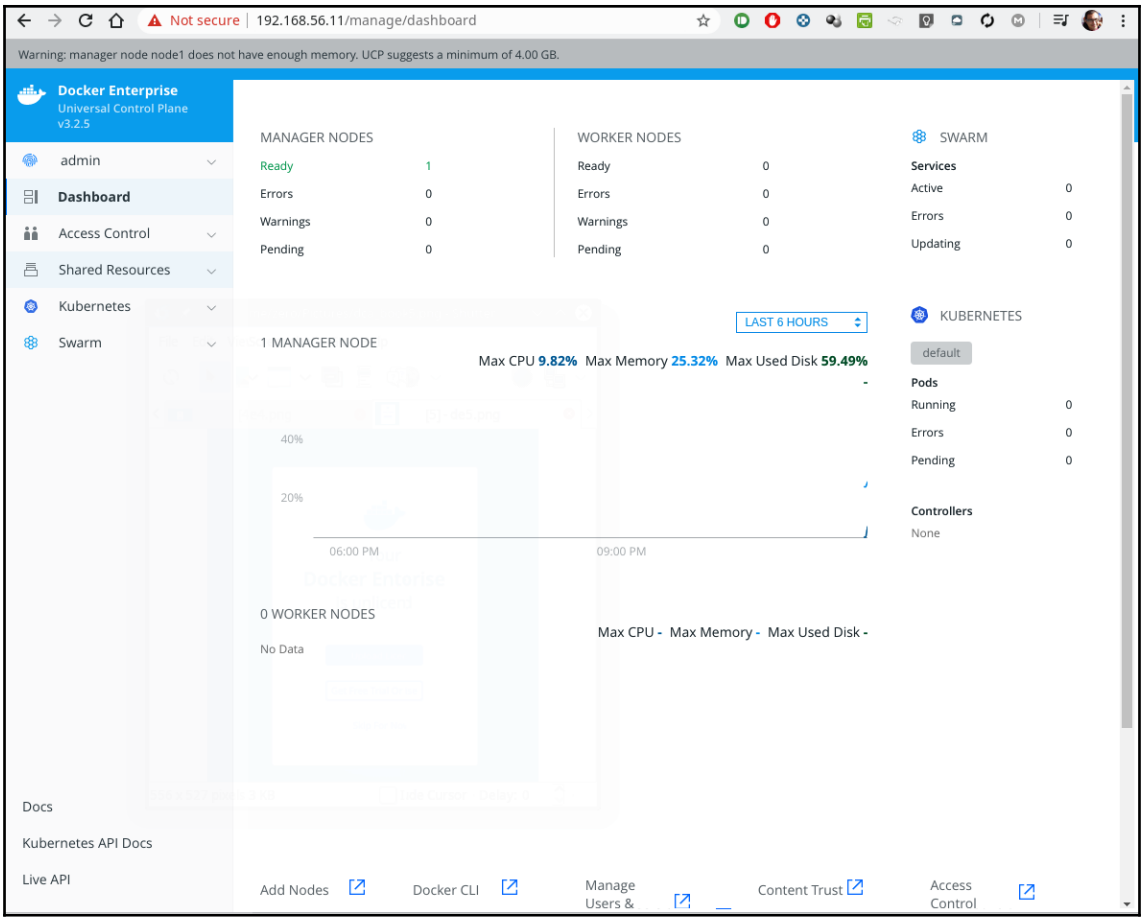


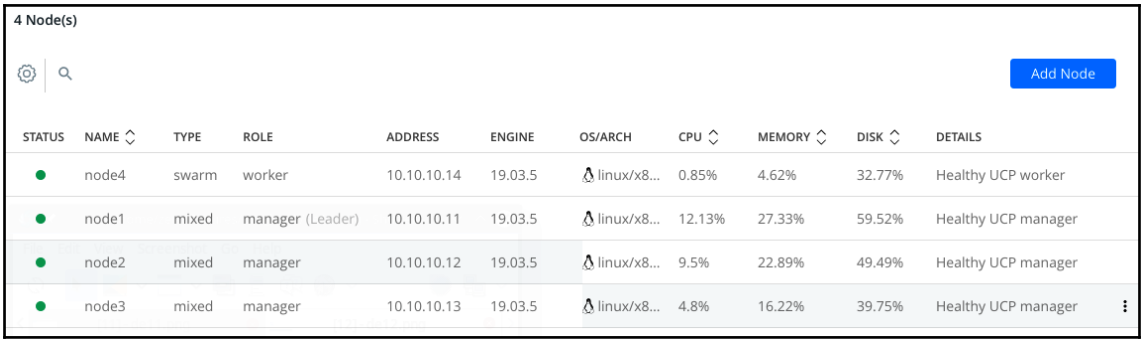
Chapter 11: Universal Control Plane

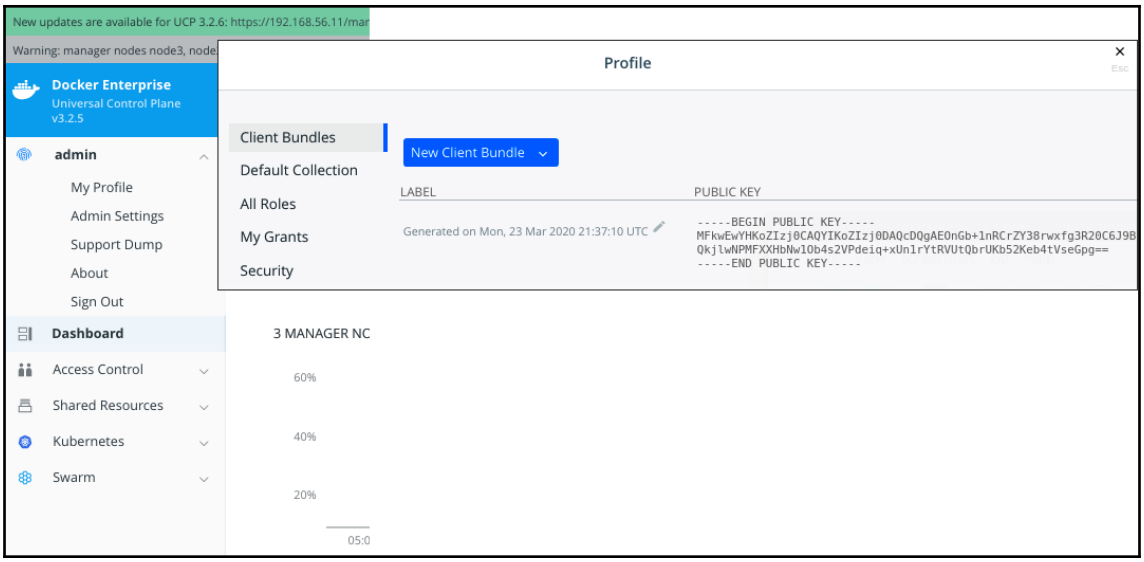


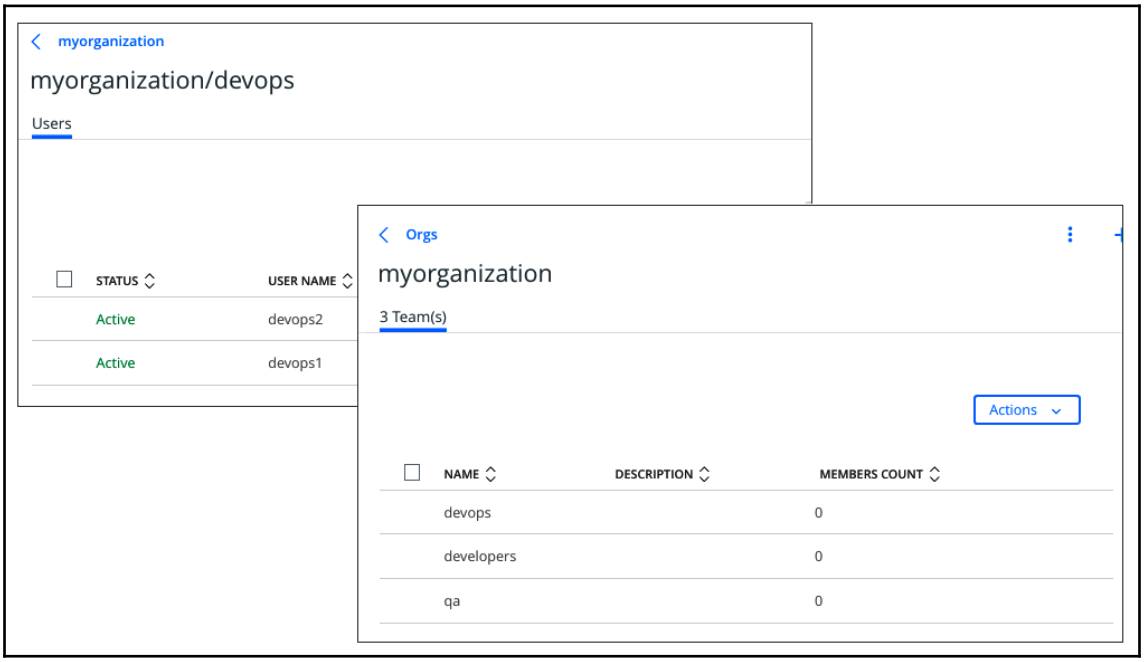

dockerhub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
Get Help
frjaraur
Setup Instructions
Account
frjaraur
1 Month Trial | Thu Feb 06 2020
Getting Started with Docker Enterprise
Install Manually on Linux servers
Follow these instructions to install:
Docker Enterprise container engine
Docker Universal Control Plane (UCP)
Docker Trusted Registry (DTR)
Setup Security Scanning
For security scanning, follow these instructions.
Resources
Expires 3/7/2020
License Key
Download CVE Vulnerability Database for DTR version 2.5.15 or lower, 2.6.11 or lower, and 2.7.4 or lower
Download CVE Vulnerability Database for DTR version 2.5.16 or higher, 2.6.12 or higher, and 2.7.5 or higher
Getting Started
User Guide
Copy and paste this URL to download your Edition:
https://storebits.docker.com/ee/trial/sub

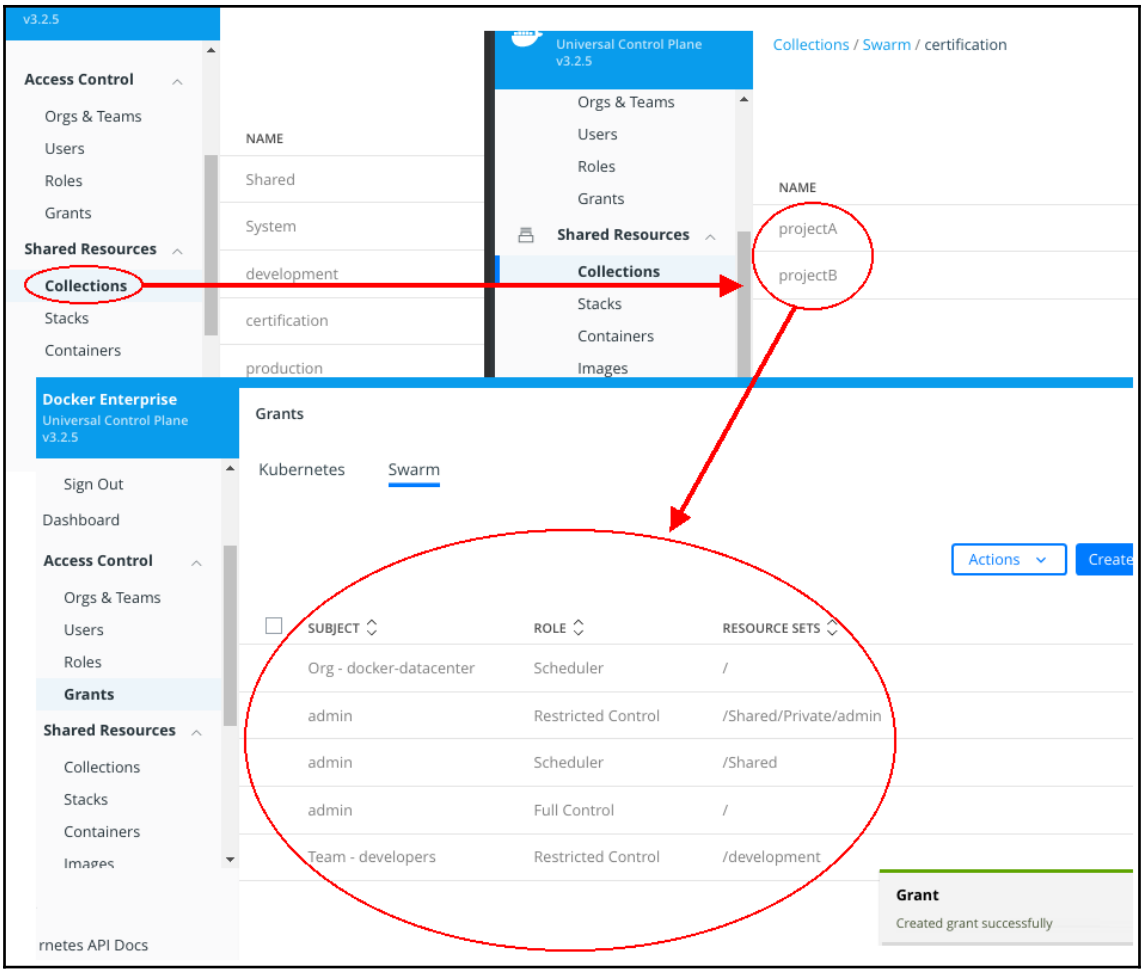

Not secure
192.168.56.11/login/
Sign in to
Docker Enterprise
admin
Sign In

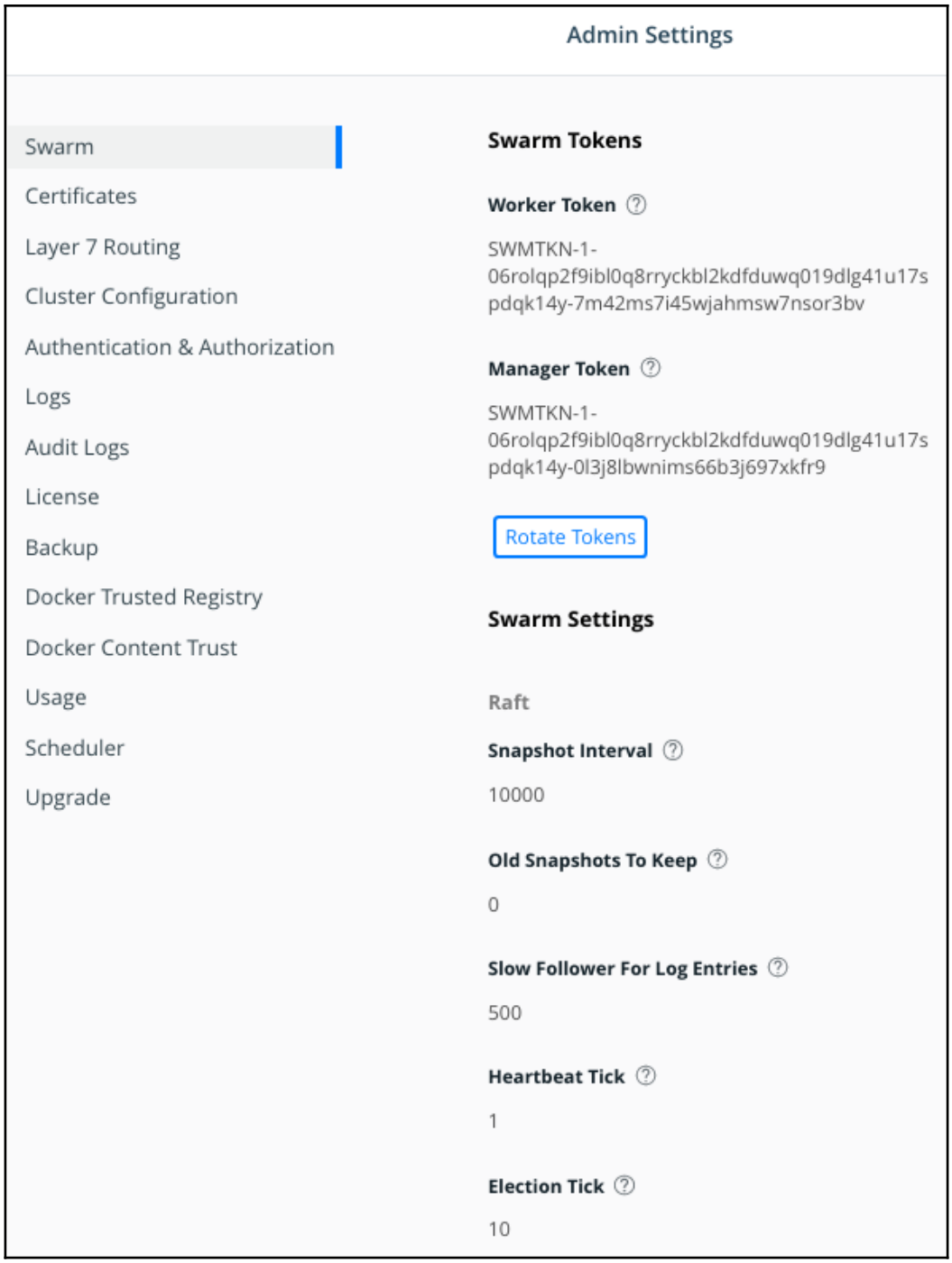

Not secure | 192.168.56.11/manage/dashboard
Warning: manager node node1 does not have enough memory. UCP suggests a minimum of 4.00 GB.
Docker Enterprise
Universal Control Plane
v3.2.5
admin
Dashboard
Access Control
Shared Resources
Kubernetes
Swarm
MANAGER NODES
Ready 1
Errors 0
Warnings 0
Pending 0
WORKER NODES
Ready 0
Errors 0
Warnings 0
Pending 0
SWARM
Services
Active 0
Errors 0
Updating 0
KUBERNETES
default
Pods
Running 0
Errors 0
Pending 0
Controllers
None
LAST 6 HOURS
1 MANAGER NODE
Max CPU 9.82% Max Memory 25.32% Max Used Disk 59.49%
40%
20%
06:00 PM
09:00 PM
0 WORKER NODES
No Data
Max CPU - Max Memory - Max Used Disk -
Docs
Kubernetes API Docs
Live API
Add Nodes
Docker CLI
Manage Users &
Content Trust
Access Control

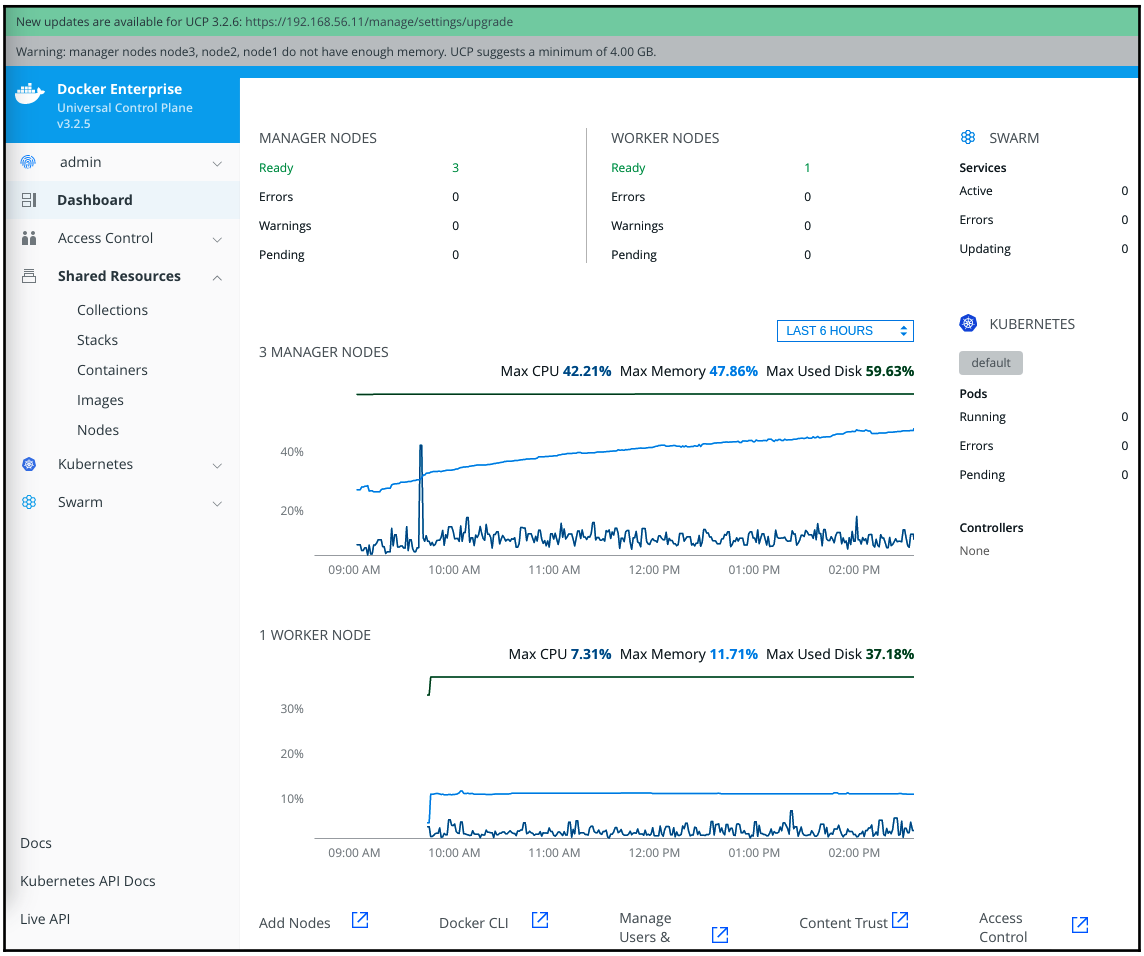

4 Node(s)
Add Node
STATUS NAME TYPE ROLE ADDRESS ENGINE OS/ARCH CPU MEMORY DISK DETAILS
node4 swarm worker 10.10.10.14 19.03.5 linux/x8... 0.85% 4.62% 32.77% Healthy UCP worker
node1 mixed manager (Leader) 10.10.10.11 19.03.5 linux/x8... 12.13% 27.33% 59.52% Healthy UCP manager
node2 mixed manager 10.10.10.12 19.03.5 linux/x8... 9.5% 22.89% 49.49% Healthy UCP manager
node3 mixed manager 10.10.10.13 19.03.5 linux/x8... 4.8% 16.22% 39.75% Healthy UCP manager


New updates are available for UCP 3.2.6: https://192.168.56.11/mar
Warning: manager nodes node3, node
Docker Enterprise
Universal Control Plane
v3.2.5
admin
My Profile
Admin Settings
Support Dump
About
Sign Out
Dashboard
Access Control
Shared Resources
Kubernetes
Swarm
Profile
Client Bundles
Default Collection
All Roles
My Grants
Security
New Client Bundle
LABEL
PUBLIC KEY
Generated on Mon, 23 Mar 2020 21:37:10 UTC
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE0nGb+1nRCrZY38rwxfg3R20C6J9B
QkjlwNPMFXXHbNw1Ob4s2VPdeiq+xUn1rYtRVUtQbrUKb52Keb4tVseGpg==
-----END PUBLIC KEY-----
3 MANAGER NC
60%
40%
20%
05:0

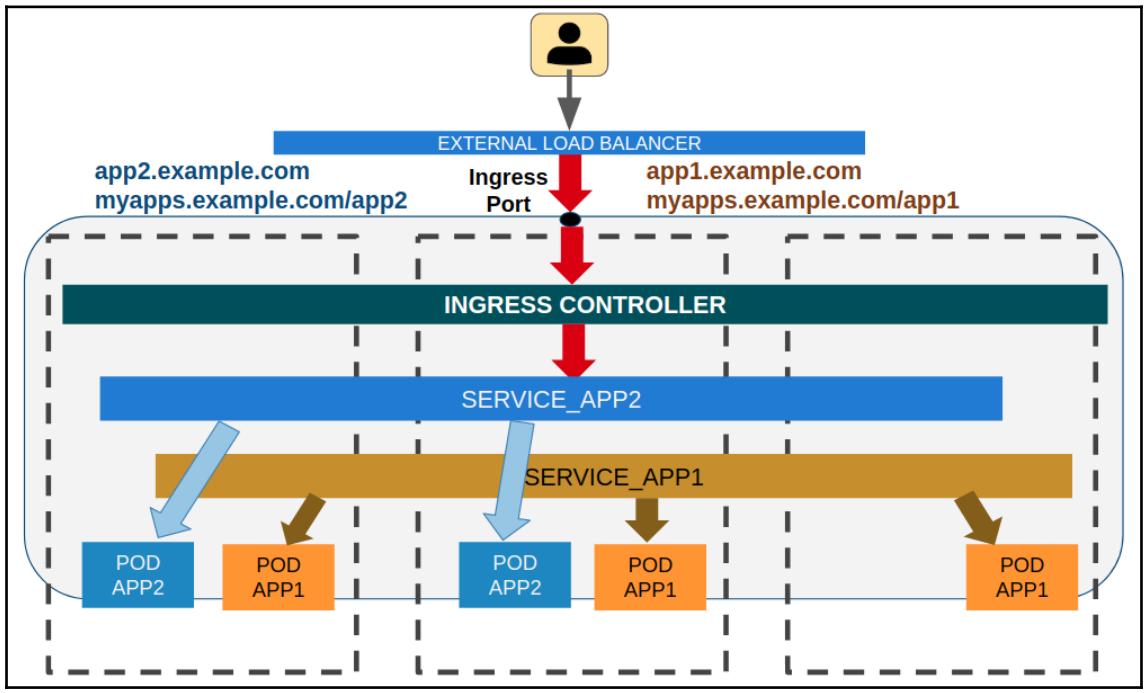

myorganization
myorganization/devops
Users
STATUS
USER NAME
Active
devops2
Active
devops1
Orgs
myorganization
3 Team(s)
Actions
NAME
DESCRIPTION
MEMBERS COUNT
devops
0
developers
0
qa
0

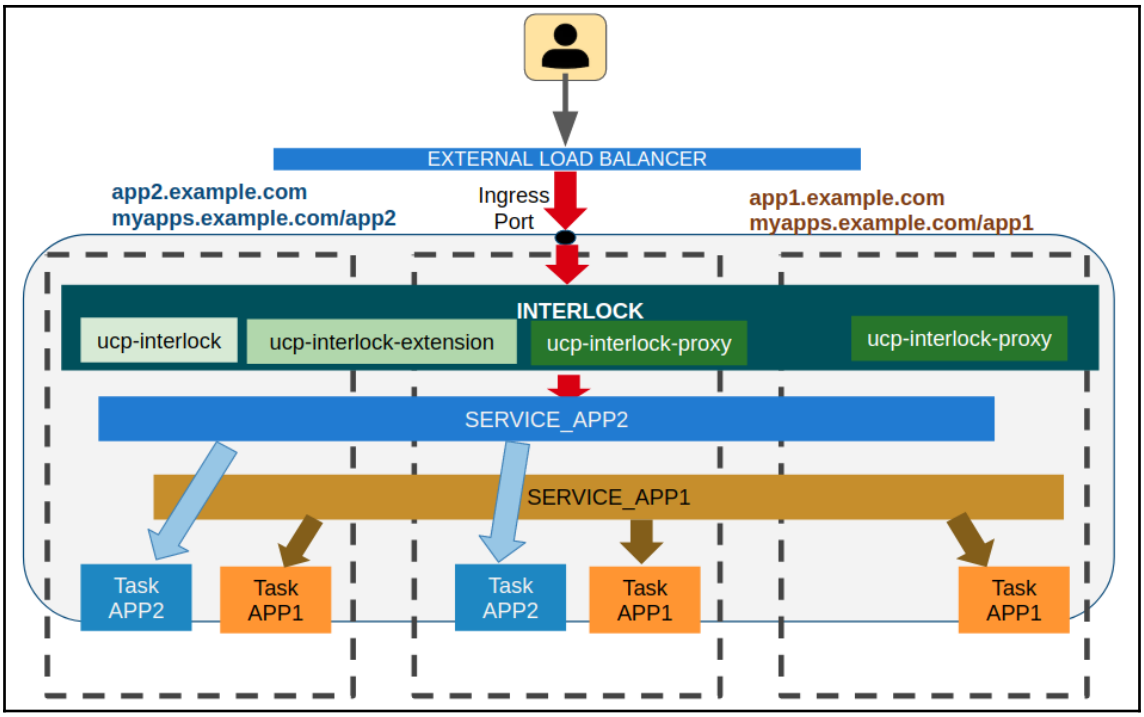

v3.2.5
Access Control
Orgs & Teams
Users
Roles
Grants
Shared Resources
Collections
Stacks
Containers
NAME
Shared
System
development
certification
production
Universal Control Plane
v3.2.5
Orgs & Teams
Users
Roles
Grants
Shared Resources
Collections
Stacks
Containers
Images
Collections / Swarm / certification
NAME
projectA
projectB
Docker Enterprise
Universal Control Plane
v3.2.5
Sign Out
Dashboard
Access Control
Orgs & Teams
Users
Roles
Grants
Shared Resources
Collections
Stacks
Containers
Images
rnetes API Docs
Grants
Kubernetes
Swarm
Actions
Create
SUBJECT
ROLE
RESOURCE SETS
Org - docker-datacenter
Scheduler
/
admin
Restricted Control
/Shared/Private/admin
admin
Scheduler
/Shared
admin
Full Control
/
Team - developers
Restricted Control
/development
Grant
Created grant successfully

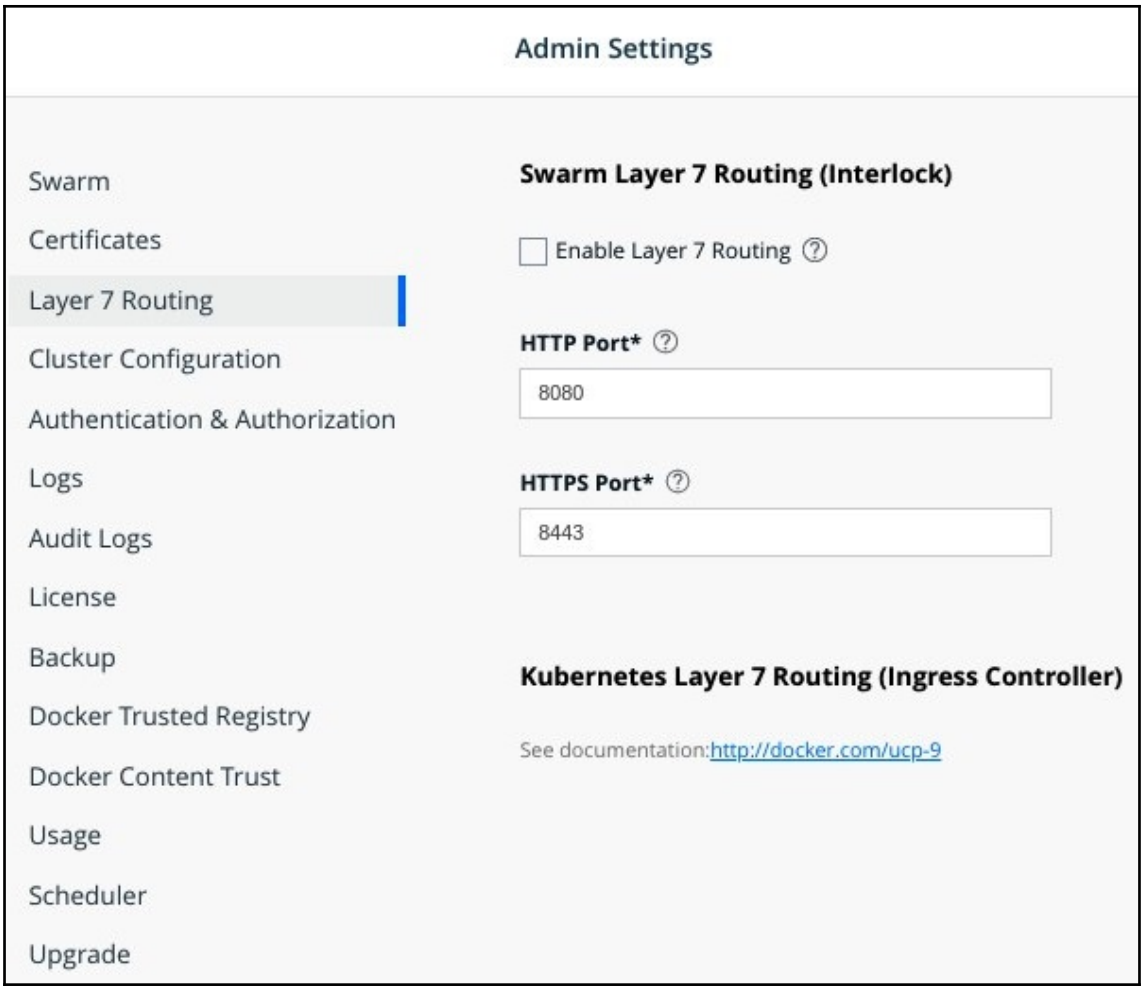

Admin Settings
Swarm
Certificates
Layer 7 Routing
Cluster Configuration
Authentication & Authorization
Logs
Audit Logs
License
Backup
Docker Trusted Registry
Docker Content Trust
Usage
Scheduler
Upgrade
Swarm Tokens
Worker Token
SWMTKN-1-06rolqp2f9ibl0q8rryckbl2kdfduwq019dlg41u17spdqk14y-7m42ms7i45wjahmsw7nsor3bv
Manager Token
SWMTKN-1-06rolqp2f9ibl0q8rryckbl2kdfduwq019dlg41u17spdqk14y-0l3j8lbwnims66b3j697xkfr9
Rotate Tokens
Swarm Settings
Raft
Snapshot Interval
10000
Old Snapshots To Keep
0
Slow Follower For Log Entries
500
Heartbeat Tick
1
Election Tick
10


New updates are available for UCP 3.2.6: https://192.168.56.11/manage/settings/upgrade
Warning: manager nodes node3, node2, node1 do not have enough memory. UCP suggests a minimum of 4.00 GB.
Docker Enterprise
Universal Control Plane
v3.2.5
admin
Dashboard
Access Control
Shared Resources
Collections
Stacks
Containers
Images
Nodes
Kubernetes
Swarm
Docs
Kubernetes API Docs
Live API
MANAGER NODES
Ready 3
Errors 0
Warnings 0
Pending 0
WORKER NODES
Ready 1
Errors 0
Warnings 0
Pending 0
SWARM
Services
Active 0
Errors 0
Updating 0
LAST 6 HOURS
3 MANAGER NODES
Max CPU 42.21% Max Memory 47.86% Max Used Disk 59.63%
40%
20%
09:00 AM
10:00 AM
11:00 AM
12:00 PM
01:00 PM
02:00 PM
KUBERNETES
default
Pods
Running 0
Errors 0
Pending 0
Controllers
None
1 WORKER NODE
Max CPU 7.31% Max Memory 11.71% Max Used Disk 37.18%
30%
20%
10%
09:00 AM
10:00 AM
11:00 AM
12:00 PM
01:00 PM
02:00 PM
Add Nodes
Docker CLI
Manage Users &
Content Trust
Access Control

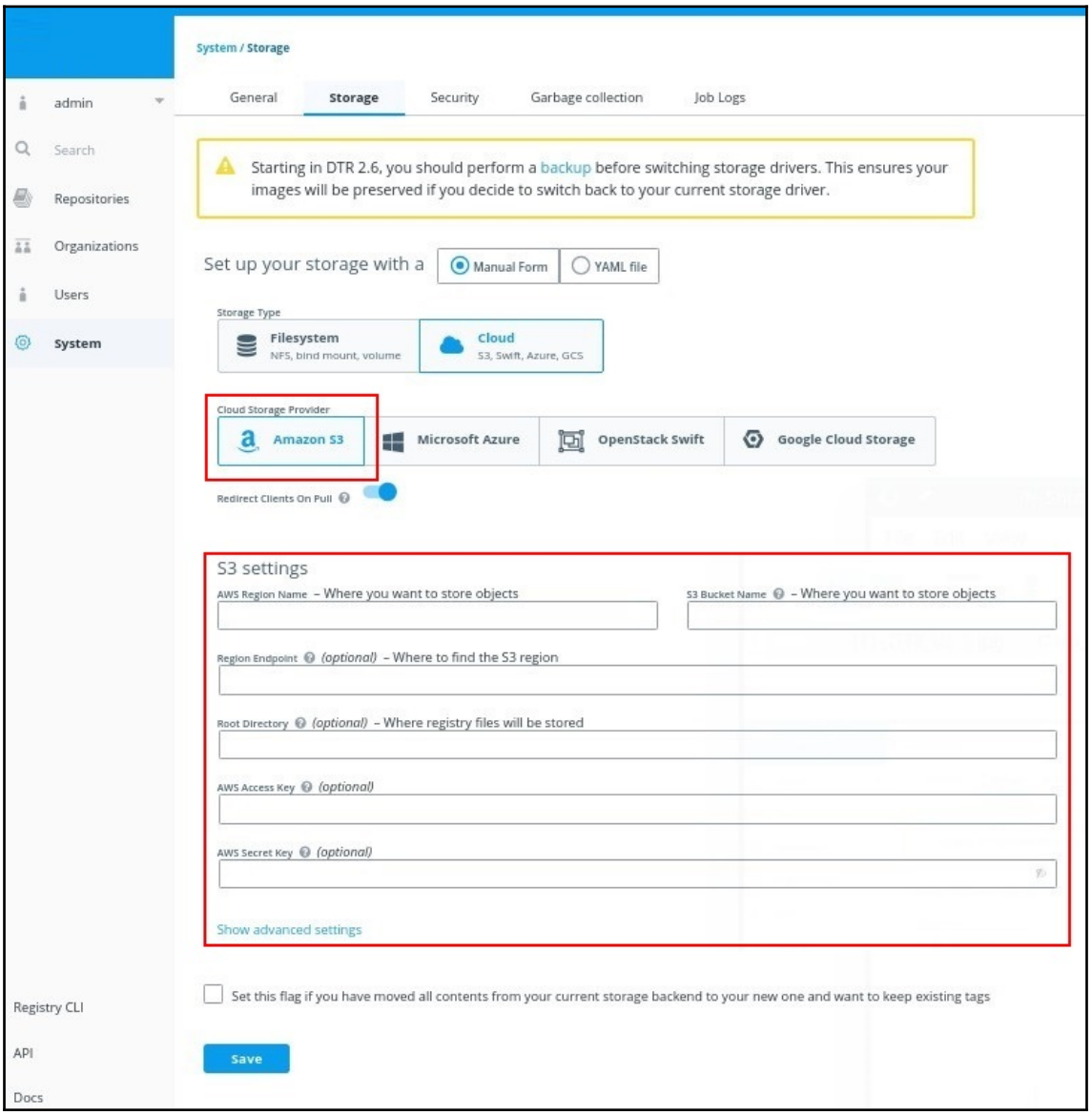
Chapter 12: Publishing Applications in Docker Enterprise

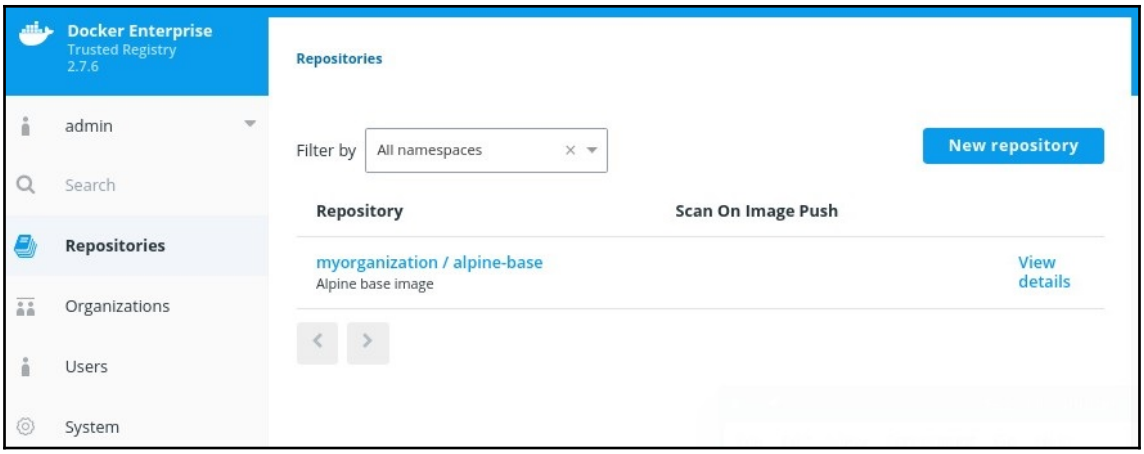


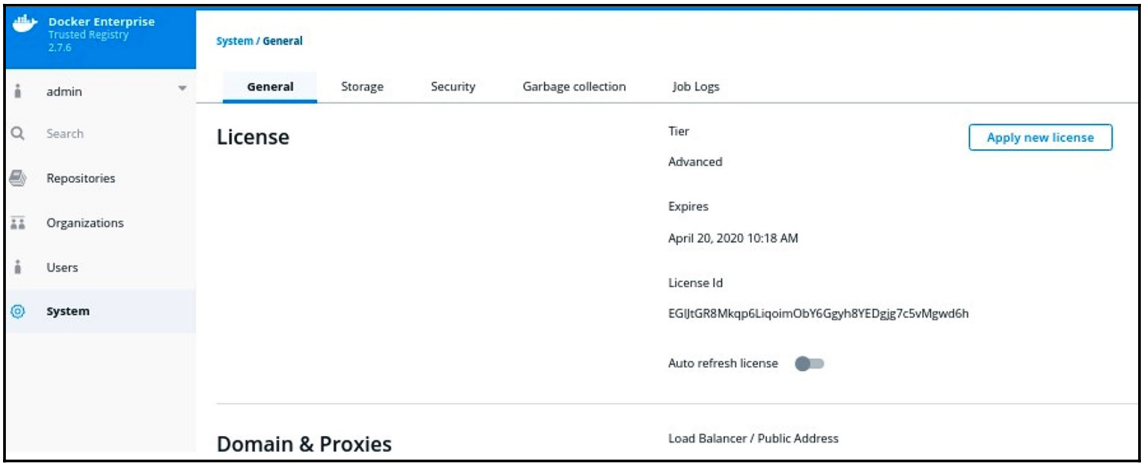

EXTERNAL LOAD BALANCER
app2.example.com
myapps.example.com/app2
Ingress
Port
app1.example.com
myapps.example.com/app1
INTERLOCK
ucp-interlock
ucp-interlock-extension
ucp-interlock-proxy
ucp-interlock-proxy
SERVICE_APP2
SERVICE_APP1
Task
APP2
Task
APP1
Task
APP2
Task
APP1
Task
APP1

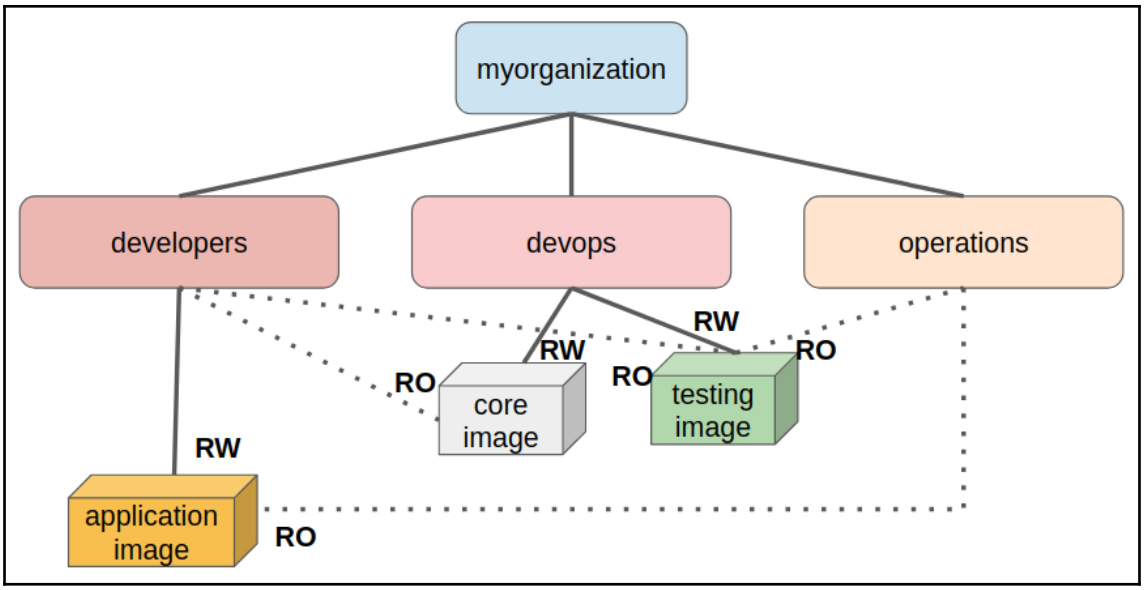

Admin Settings
Swarm
Certificates
Layer 7 Routing
Cluster Configuration
Authentication & Authorization
Logs
Audit Logs
License
Backup
Docker Trusted Registry
Docker Content Trust
Usage
Scheduler
Upgrade
Swarm Layer 7 Routing (Interlock)
Enable Layer 7 Routing
HTTP Port*
8080
HTTPS Port*
8443
Kubernetes Layer 7 Routing (Ingress Controller)
See documentation:http://docker.com/ucp-9

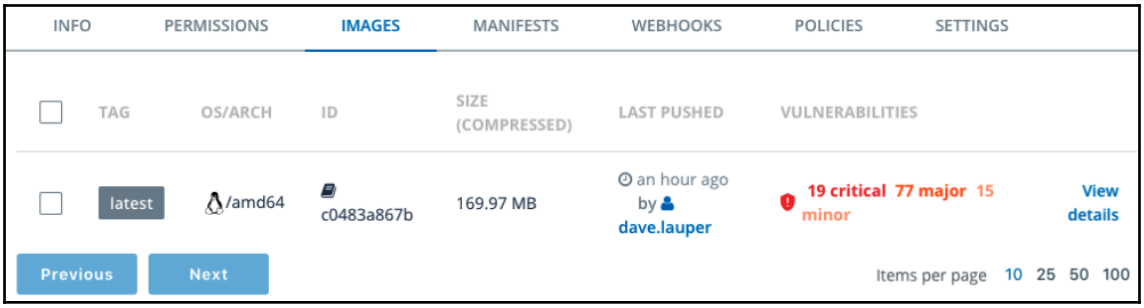
Chapter 13: Implementing an Enterprise-Grade Registry with DTR

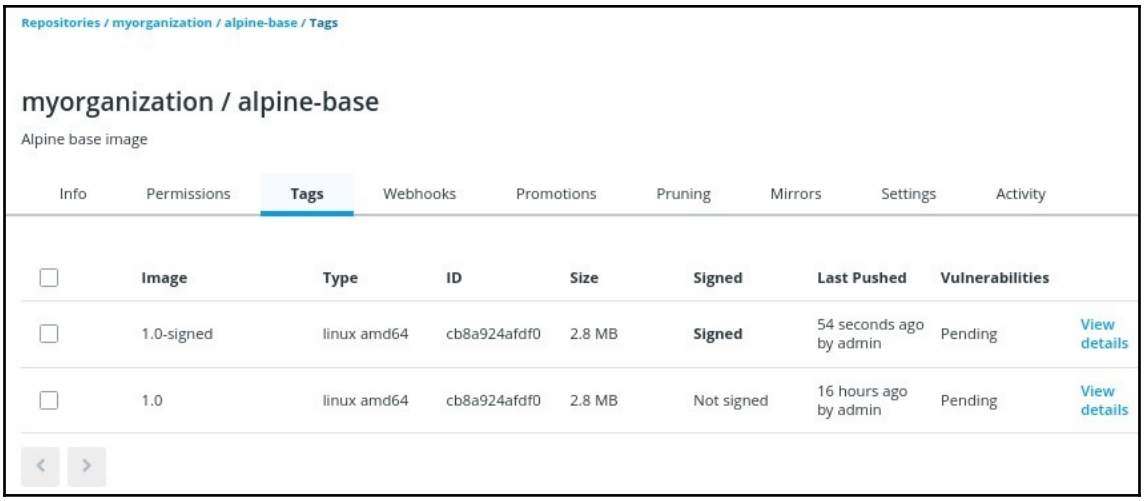


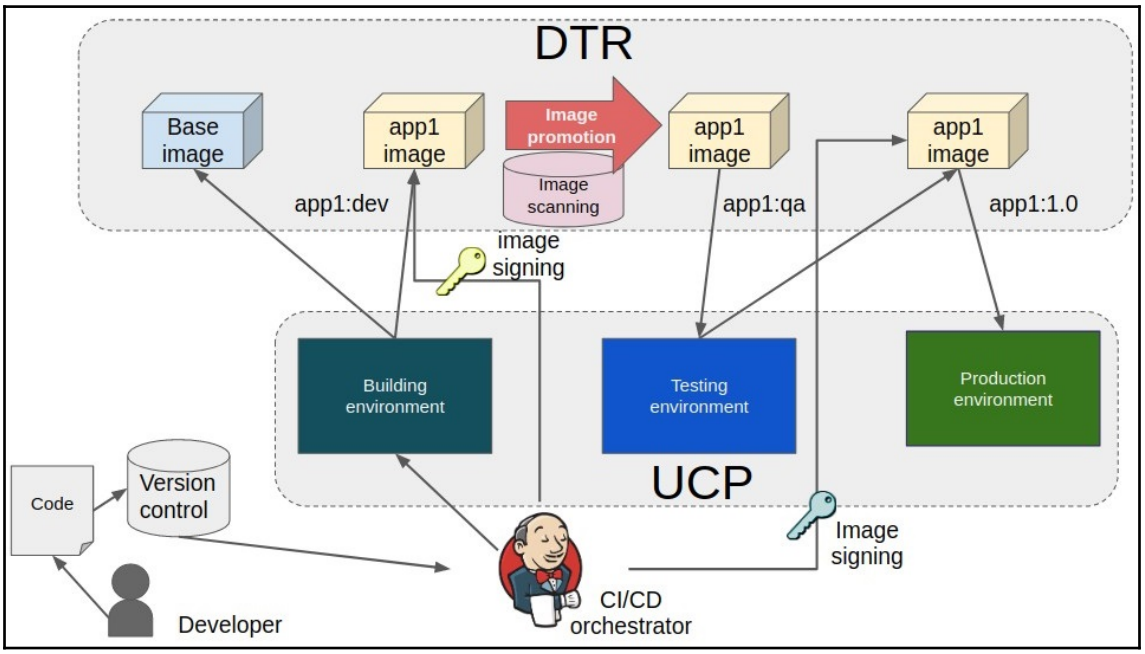

Docker Enterprise
Trusted Registry
2.7.6
admin
Search
Repositories
Organizations
Users
System
Repositories
Filter by
All namespaces
New repository
Repository
Scan On Image Push
myorganization / alpine-base
Alpine base image
View details

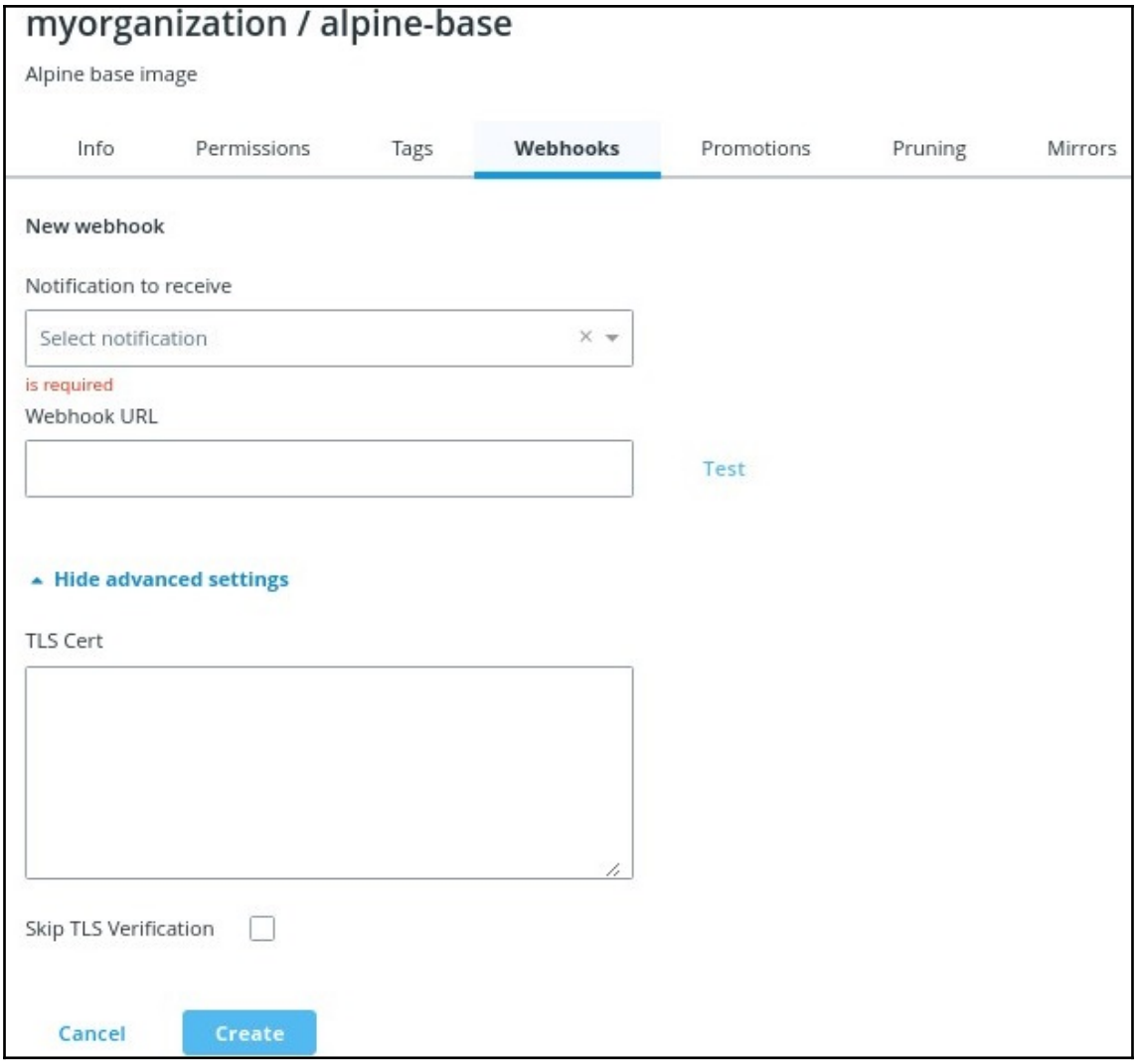

Docker Enterprise
Trusted Registry
2.7.6
admin
Search
Repositories
Organizations
Users
System
System / General
General
Storage
Security
Garbage collection
Job Logs
License
Tier
Advanced
Apply new license
Expires
April 20, 2020 10:18 AM
License Id
EGlJtGR8Mkqp6LiqoimObY6Ggyh8YEDgjg7c5vMgwd6h
Auto refresh license
Domain & Proxies
Load Balancer / Public Address

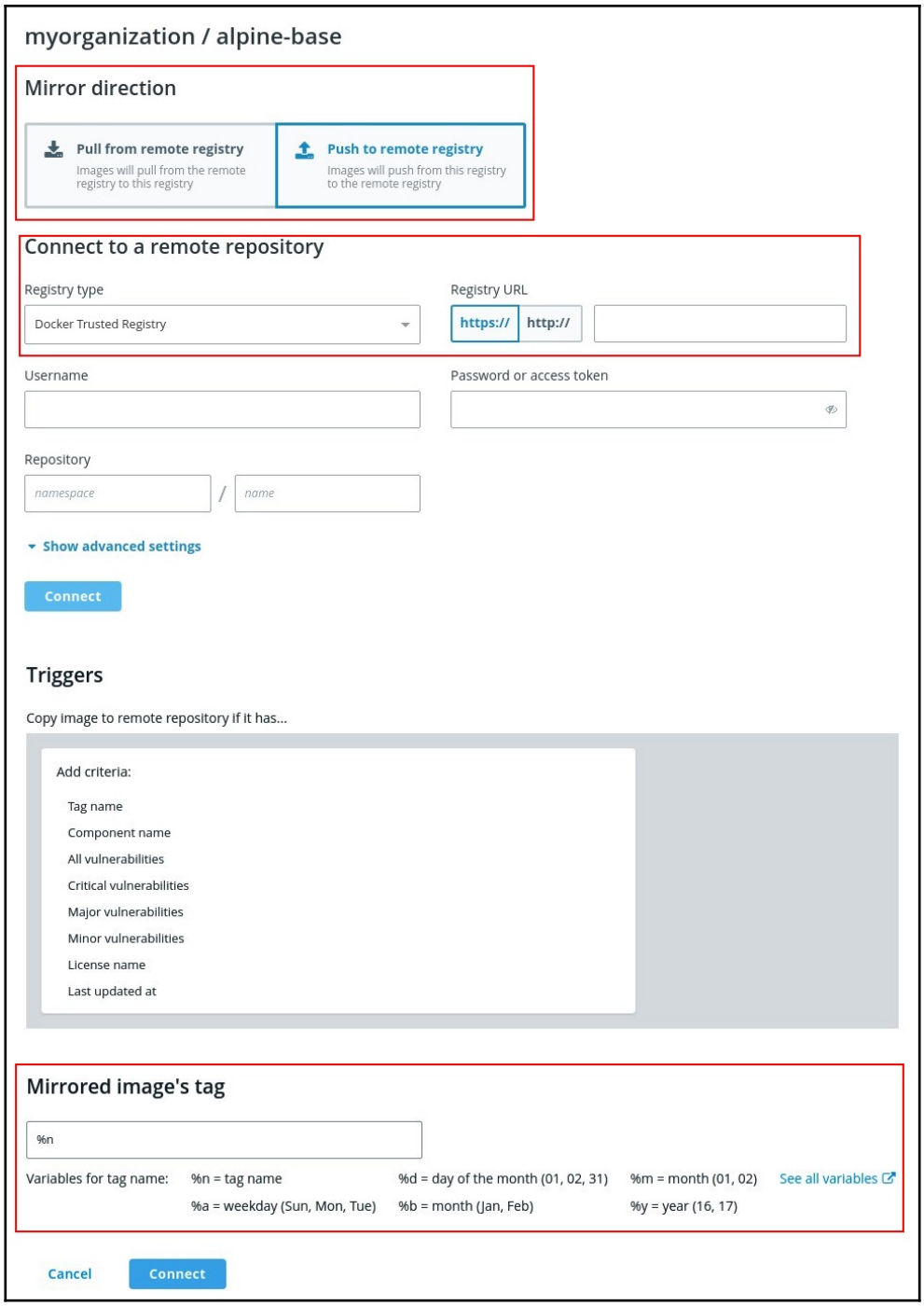

myorganization
developers
devops
operations
RW
RW
RO
RO
RO
core image
testing image
RW
application image
RO

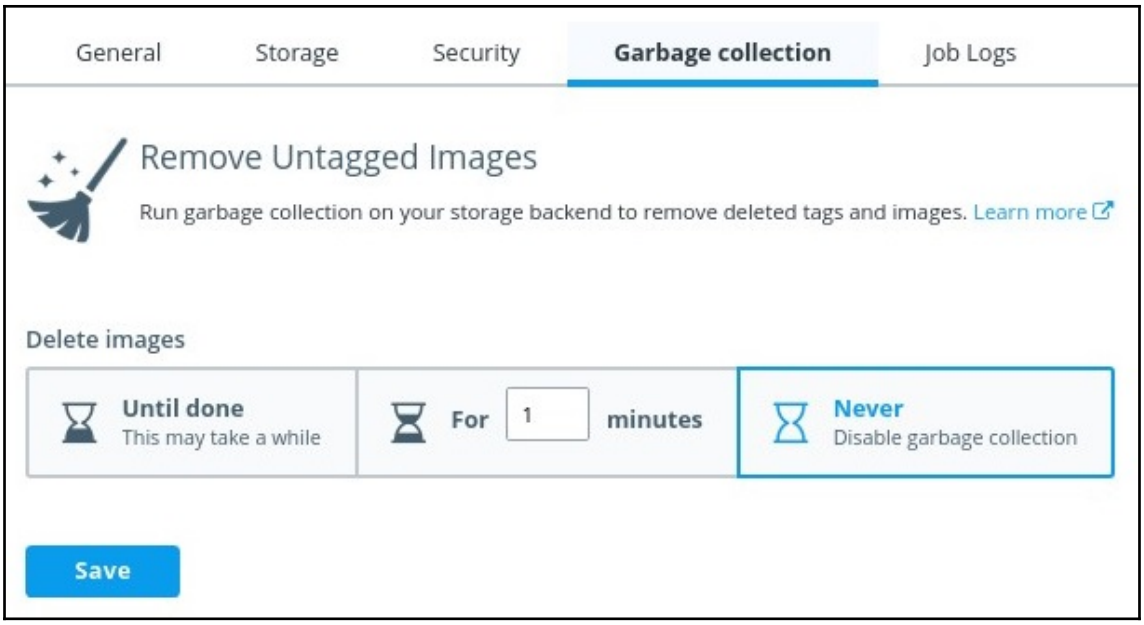

INFO
PERMISSIONS
IMAGES
MANIFESTS
WEBHOOKS
POLICIES
SETTINGS
TAG
OS/ARCH
ID
SIZE (COMPRESSED)
LAST PUSHED
VULNERABILITIES
latest
/amd64
c0483a867b
169.97 MB
an hour ago by dave.lauper
19 critical 77 major 15 minor
View details
Previous
Next
Items per page 10 25 50 100


Repositories / myorganization / alpine-base / Tags
myorganization / alpine-base
Alpine base image
Info
Permissions
Tags
Webhooks
Promotions
Pruning
Mirrors
Settings
Activity
Image
Type
ID
Size
Signed
Last Pushed
Vulnerabilities
1.0-signed
linux amd64
cb8a924afdf0
2.8 MB
Signed
54 seconds ago by admin
Pending
View details
1.0
linux amd64
cb8a924afdf0
2.8 MB
Not signed
16 hours ago by admin
Pending
View details


DTR
Base image
app1 image
Image promotion
Image scanning
app1 image
app1 image
app1:dev
app1:qa
app1:1.0
image signing
Building environment
Testing environment
Production environment
UCP
Code
Version control
Developer
CI/CD orchestrator
Image signing


myorganization / alpine-base
Alpine base image
Info
Permissions
Tags
Webhooks
Promotions
Pruning
Mirrors
New webhook
Notification to receive
Select notification
is required
Webhook URL
Test
Hide advanced settings
TLS Cert
Skip TLS Verification
Cancel
Create


myorganization / alpine-base
Mirror direction
Pull from remote registry
Images will pull from the remote registry to this registry
Push to remote registry
Images will push from this registry to the remote registry
Connect to a remote repository
Registry type
Docker Trusted Registry
Registry URL
https://
http://
Username
Password or access token
Repository
namespace
/
name
Show advanced settings
Connect
Triggers
Copy image to remote repository if it has...
Add criteria:
Tag name
Component name
All vulnerabilities
Critical vulnerabilities
Major vulnerabilities
Minor vulnerabilities
License name
Last updated at
Mirrored image's tag
%n
Variables for tag name:
%n = tag name
%d = day of the month (01, 02, 31)
%m = month (01, 02)
See all variables
%a = weekday (Sun, Mon, Tue)
%b = month (Jan, Feb)
%y = year (16, 17)
Cancel
Connect


General
Storage
Security
Garbage collection
Job Logs
Remove Untagged Images
Run garbage collection on your storage backend to remove deleted tags and images. Learn more
Delete images
Until done
This may take a while
For
1
minutes
Never
Disable garbage collection
Save