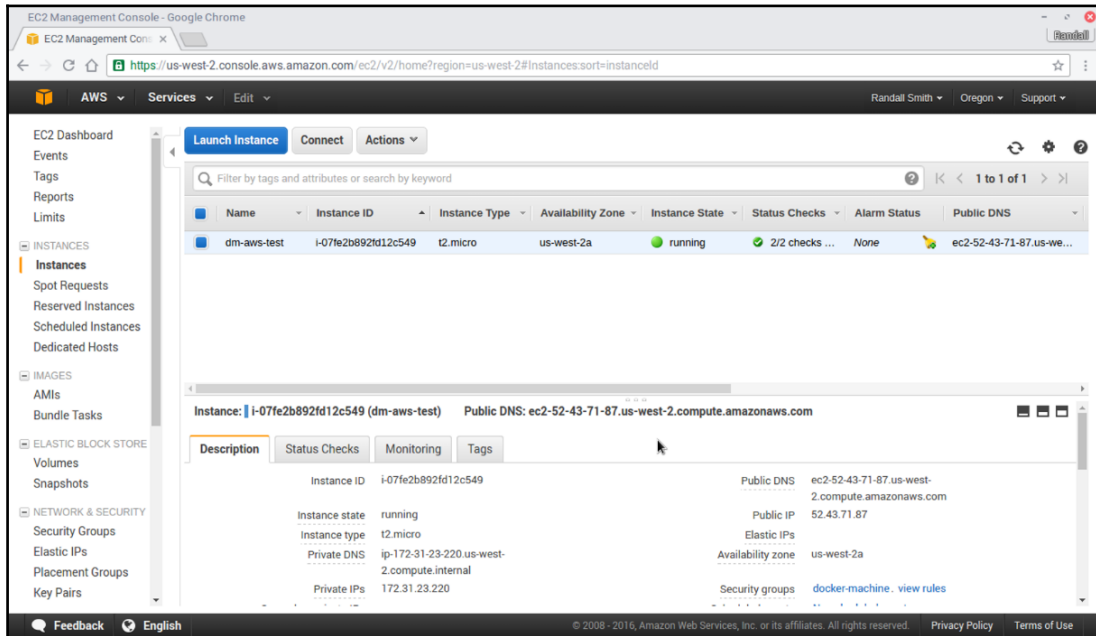


Chapter 1: Getting Started with Docker Orchestration



Compute Engine - docker-test - Google Chrome

https://console.cloud.google.com/compute/instances?project=docker-test-141618&graph=GCE_CPU&duration=PT1H

Google Cloud Platform docker-test

Compute Engine VM instances

CREATE INSTANCE CREATE INSTANCE GROUP START STOP RESET DELETE

Filter by label or name Columns Labels Recommendations

CPU utilization 1 hour 6h 12h 1 day 2d 4d 7d 14d 30d

CPU % CPU Aug 27, 1:24 PM

150
100
50

Aug 27, 12:30 PM Aug 27, 12:45 PM Aug 27, 1:00 PM Aug 27, 1:15 PM Aug 27, 1:26 PM

CPU: 148

Name	Zone	Machine type	Recommendation	In use by	Internal IP	External IP	Connect
<input checked="" type="checkbox"/> dm-gce-test	us-central1-a	1 vCPU, 0.6 GB			10.128.0.2	104.154.24.36	SSH

Virtual machines - Microsoft Azure - Google Chrome

https://portal.azure.com/#blade/HubsExtension/BrowseResourceBlade/resourceType/Microsoft.Compute%2FVirtualMachines

Microsoft Azure Virtual machines

Virtual machines (Default Directory)

+ Add Columns Refresh

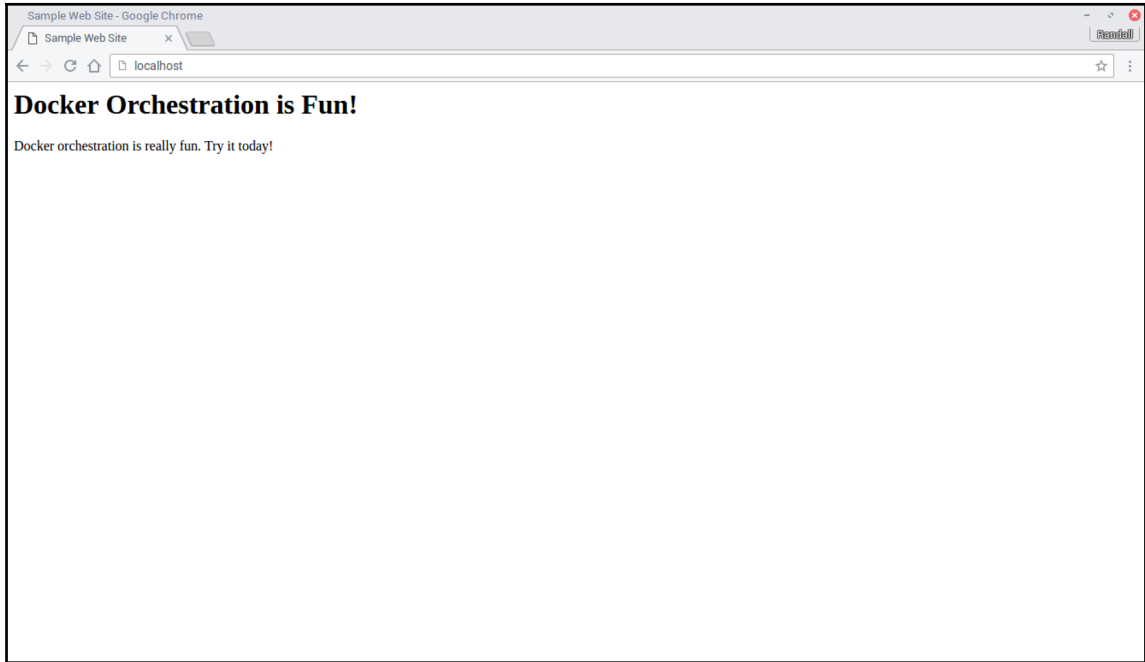
Subscriptions: Free Trial

Filter items...

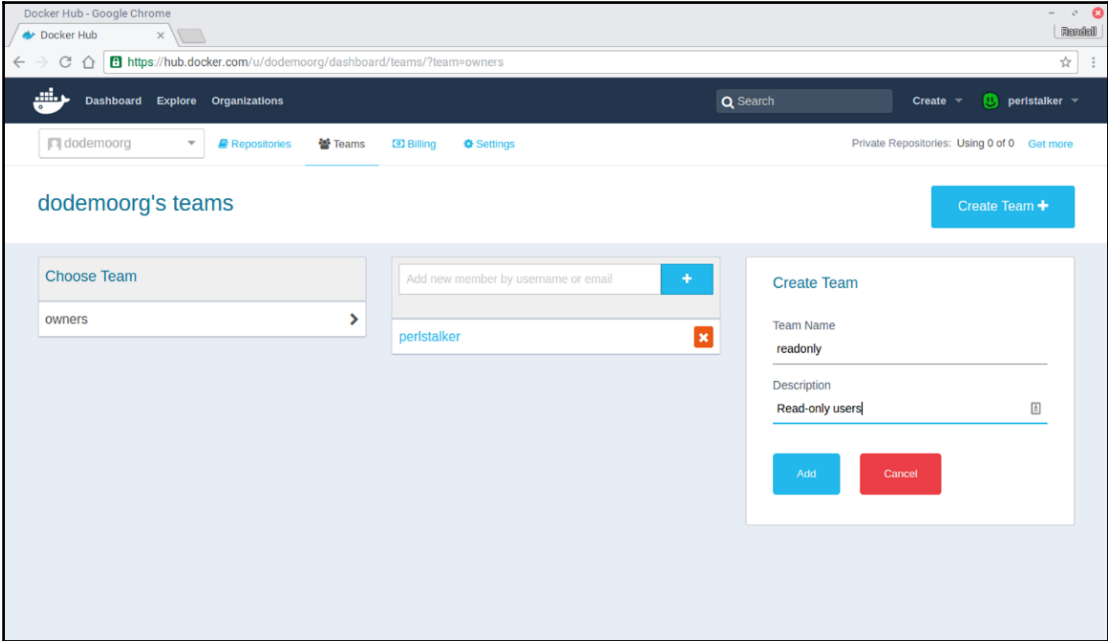
NAME	STATUS	RESOURCE GROUP	LOCATION	SUBSCRIPTION
dm-azure-test	Creating	docker-machine	West US	Free Trial

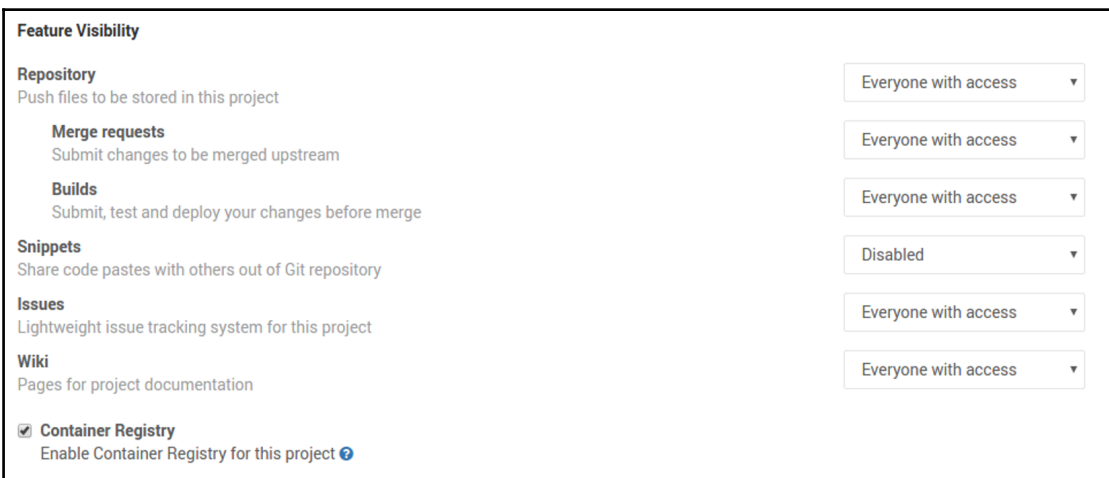
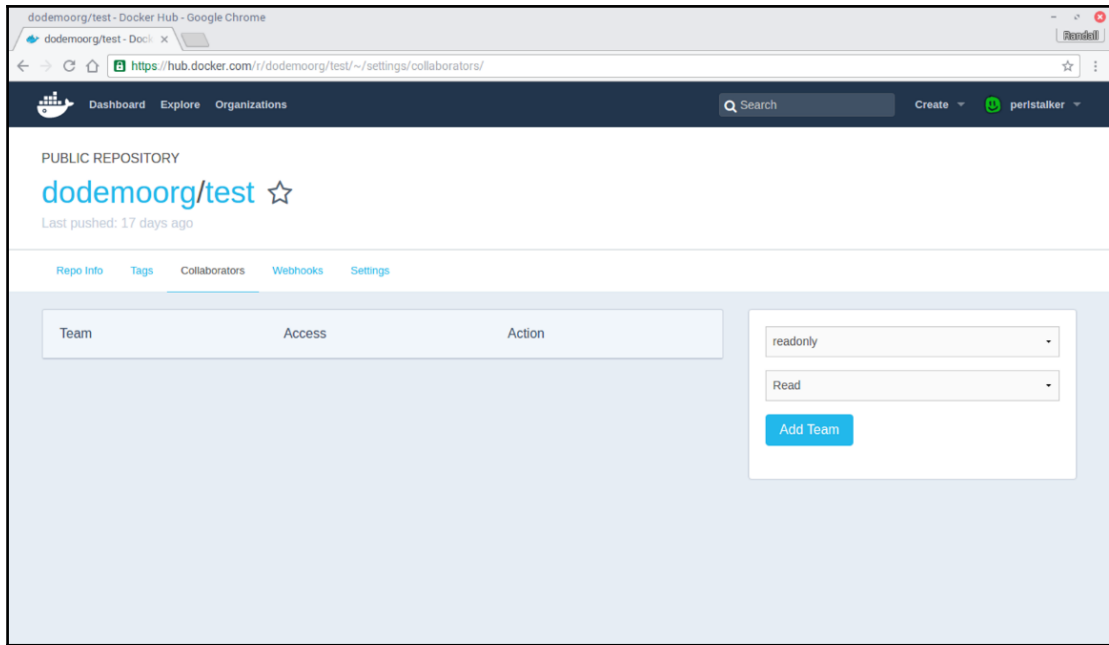


Chapter 2: Building Multi-Container Applications with Docker Compose

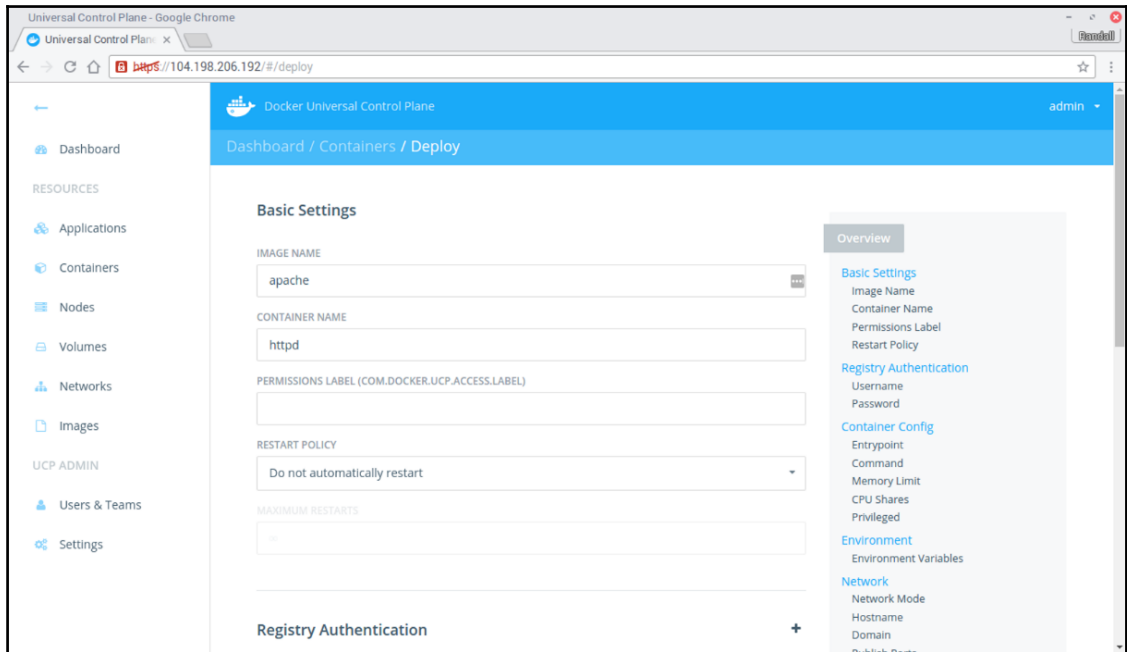


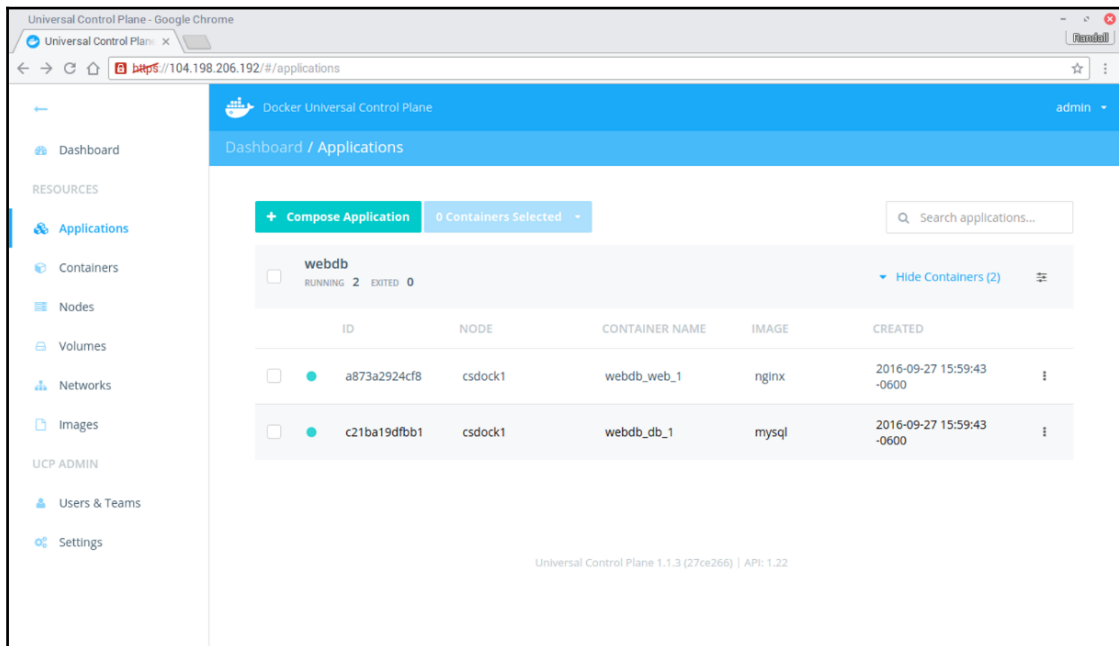
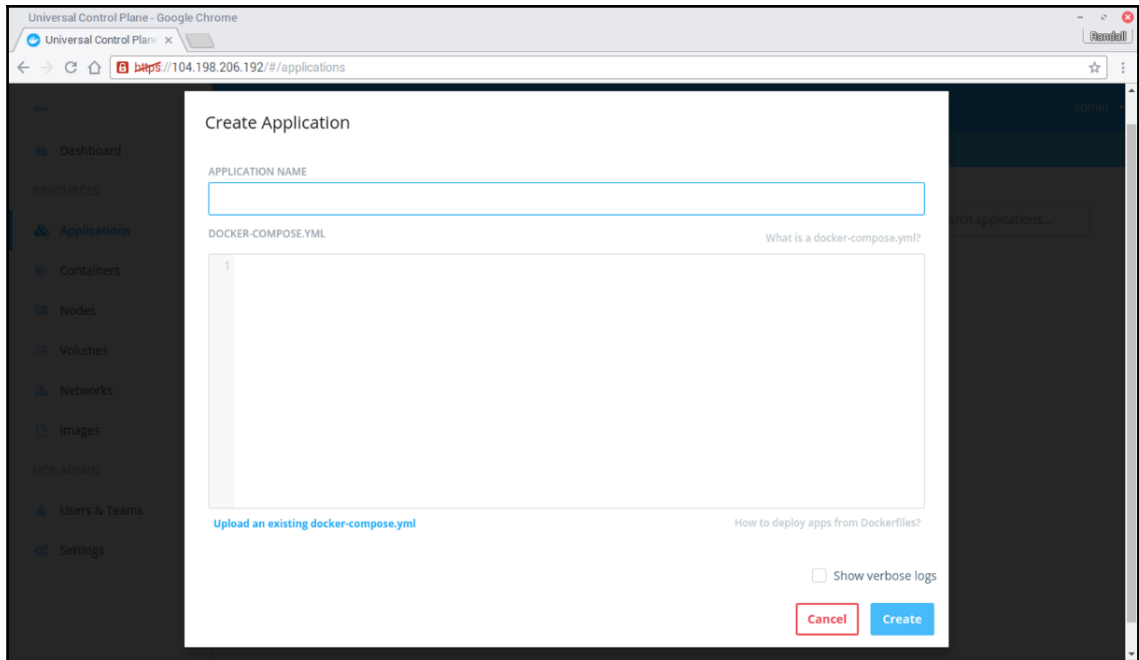
Chapter 3: Cluster Building Blocks – Registry, Overlay Networks, and Shared Storage



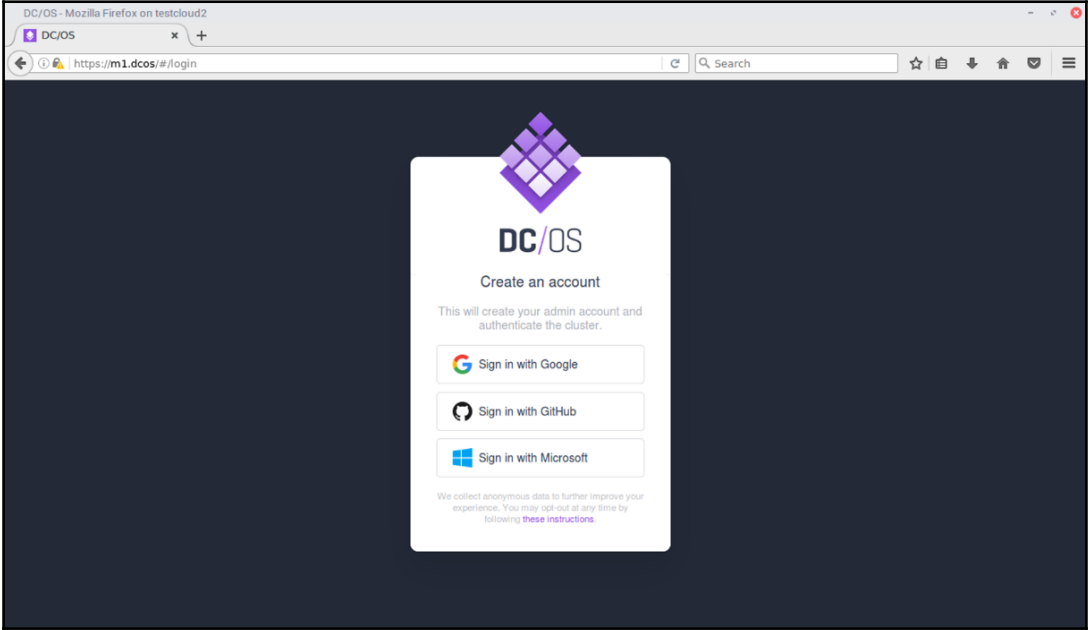


Chapter 4: Orchestration with Docker Swarm





Chapter 6: Working with Mesosphere



DC/OS - Mozilla Firefox on testcloud2

DC/OS

https://m1.dcos/#/dashboard/

dcos-vagrant
192.168.65.90

Dashboard
Services
Jobs
Network
Nodes
Universe
System

CPU Allocation
Memory Allocation
Disk Allocation

Install DC/OS CLI

Installation

Choose your operating system and follow the instructions. For any issues or questions, please refer to our [documentation](#).

Windows OS X Linux

Copy and paste the code snippet into the terminal:

```
curl -fLs5 --retry 20 -Y 100000 -y 60 https://downloads.sudo mv dcos /usr/local/bin &&sudo chmod +x /usr/local/bin/dcos &&dcos config set core.dcos_url https://m1.dcos &&dcos
```

Close

No Services Running

0%
0 B of 11 GiB

Component Health

Admin Router Ag... Healthy
Admin Router M... Healthy
Admin Router Re... Healthy

DC/OS - Google Chrome

DC/OS

https://104.197.113.232/#/services/%2Fnginx/

cluster_name
10.128.0.3

Dashboard
Services
Jobs
Network
Nodes
Universe
System

Edit Service

JSON mode

General

Container Settings

Configure your Docker Container. You can configure your Docker volumes in the Volumes tab and your Docker ports in the Network tab.

Container Image

nginx:1.11

Extend runtime privileges Force pull image on launch

Docker Parameters

Key	Value
<input type="text"/>	<input type="text"/>

+ Add Parameter

Cancel Deploy Changes

Scale Edit More

UPDATED VERSION

7 minutes ago

7 minutes ago 10/14/2015 7:04:30 PM

DC/OS - Google Chrome
https://104.197.113.232/#/services/%2Fnginx/

cluster_name
10.128.0.3

- Dashboard
- Services
- Jobs
- Network
- Nodes
- Universe
- System

Edit Service

JSON mode

General

Container Settings

Network

Environment Variables

Labels

Health Checks

Volumes

Optional

Network

Configure the networking for your service. We will automatically generate a Service Address to connect to for each of your load balanced endpoints. [Read more about load balancing.](#)

Network Type

Virtual Network: dcos

Service Endpoints

Container Port	Name	Protocol	
80	web	tcp	<input checked="" type="checkbox"/> Load Balanced X

+ Add an endpoint

Clients can access your service at these Service Addresses

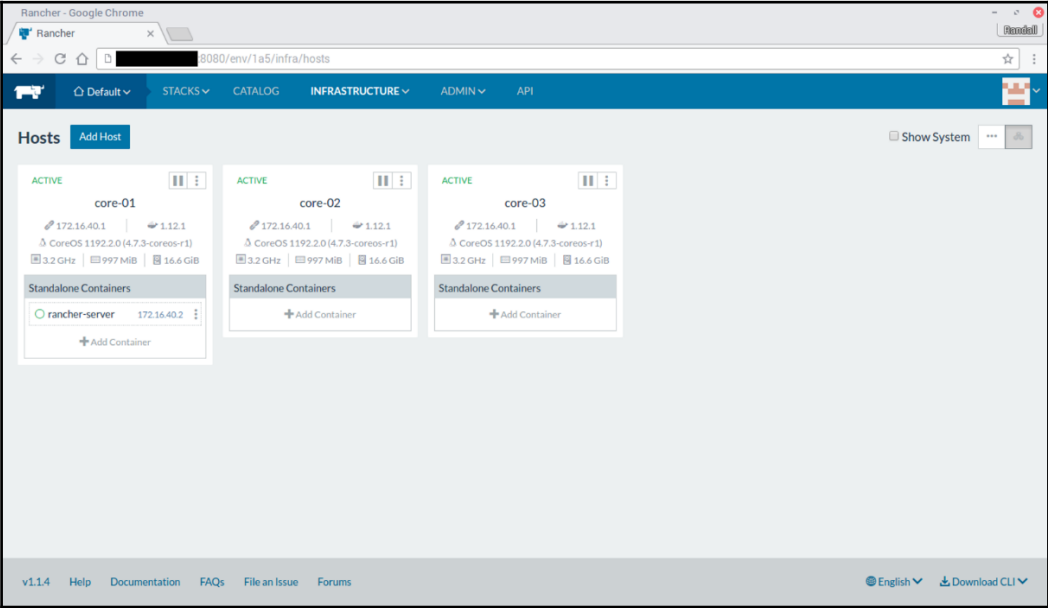
- nginx.marathon.4lib.thisdcos.directory:80

Cancel Deploy Changes

Scale Edit More

IP	UPDATED	VERSION
IP	3 minutes ago	10.142.0.11 17:00:45 PM
IP	3 minutes ago	
IP	4 minutes ago	

Chapter 7: Using Simpler Orchestration Tools – Fleet and Cattle



Rancher - Google Chrome

Rancher

8080/env/1a5/infra/hosts/add?driver=custom

- 1 Start up a Linux machine somewhere and install the latest version of Docker on it.
- 2 Make sure any security groups or firewalls allow traffic:
 - From and To all other hosts on UDP ports 500 and 4500 (for IPsec networking)
- 3 Optional: Add labels to be applied to the host.
- 4 Optional: Specify the public IP that should be used for this host. This is required if you're trying to add the host the `rancher/server` container is on.
- 5 Copy, paste, and run the command below to register the host with Rancher:

```
sudo docker run -d --privileged -v /var/run/docker.sock:/var/run/docker.sock -v /var/lib/rancher:/var/lib/rancher rancher/agent:v1.0.2 http://172.17.0.101:8080/v1/scripts/781697f518071810563:1477180800000:K8gZUqgoc3eFFCgkQeLAB0NEXFz
```
- 5 Click close below. The new host should pop up on the Hosts screen within a minute.

Rancher - Google Chrome

Rancher

0080/env/1a5/api

Default STACKS CATALOG INFRASTRUCTURE ADMIN API

Environment API Keys Add Environment API Key

Environment API keys are tied to this specific Environment (**Default**) and can only manipulate resources within there. Other accounts with access to this Environment can also manage these keys.

Endpoint: <http://0080/v1/projects/1a5>

State	Name	Description	Access Key	Created
Active	demo	None	C7CEF9002676375EBASA	Last Monday at 8:36 PM

ADVANCED OPTIONS

v1.1.4 Help Documentation FAQs File an Issue Forums English Download CLI

Rancher - Google Chrome

Rancher

0080/env/1a5/apps/add-balancer?environmentId=1e2

Default STACKS CATALOG INFRASTRUCTURE ADMIN API

Add Load Balancer

Scale

Run 1 container Always run one instance of this container on every host

Name: Description:

Listening Ports

Source IP/Port*	Protocol	SSL	Default Target Port	Access
<input type="text" value="On Host, e.g. 80 or 1.2.3.4:80"/>	<input type="text" value="http"/>	<input type="checkbox"/>	<input type="text" value="In Container, e.g. 8080"/>	<input type="text" value="Public"/>

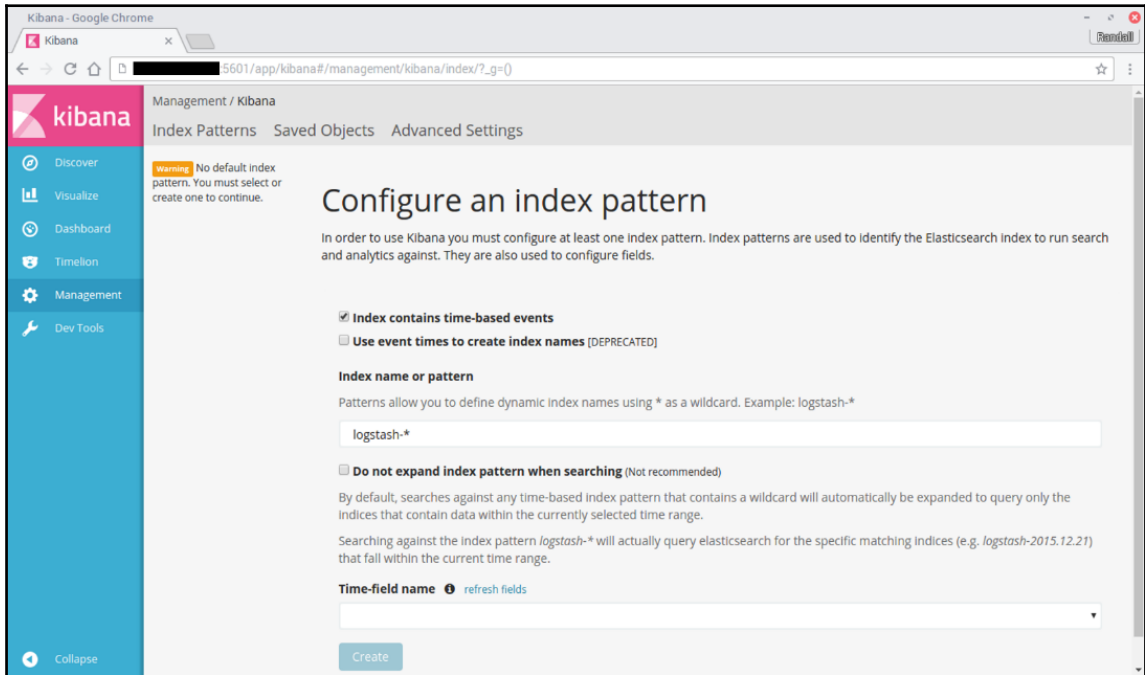
Targets

Target Service*:

Show advanced routing options - Direct requests to different services based on port, Host header, or request path

SSL Termination Stickiness Custom haproxy.cfg Labels Scheduling

Chapter 8: Monitoring Your Cluster



Kibana - Google Chrome

5 hits
container_name:bb1

logstash-*
October 29th 2016, 00:00:00.000 - October 29th 2016, 23:59:59.999 — by 30 minutes

Count

Time -

_source

```

October 29th 2016, 13:55:56.234 container_name: bb1 source_host: 172.17.8.102 level: 6 created: October 29th 2016, 13:55:56.234 message: / # / # [jls version: 1.1 command: /w/w sh image_name: busybox @timestamp: October 29th 2016, 13:55:58.838 host: core-02 @version: 1 tag: image_id: sha256:e02e811dd08fd49e7f6032625495118e63f597eb150403d0e23238af1df240ba container_id: b810acb266e0826242c8d07a1d21b374cd3645c8c7305132574f2abdf3e3e65 id: AVgS
October 29th 2016, 13:55:56.234 container_name: bb1 source_host: 172.17.8.102 level: 6 created: October 29th 2016, 13:55:56.234 message: [1;34mbin[0m [1;34mdev[0m [1;34metc[0m [1;34mhome[0m [1;34mproc[0m [1;34mroot[0m [1;34msys[0m [1;34mtmp[0m [1;34msur[0m [1;34mvar[0m [1;34m[0m version: 1.1 command: /w/w sh image_name: busybox @timestamp: October 29th 2016, 13:55:58.838 host: core-02 @version: 1 tag: image_id: sha256:e02e811dd08fd49e7f6032625495118e63f59
October 29th 2016, 13:55:56.234 container_name: bb1 source_host: 172.17.8.102 level: 6 created: October 29th 2016, 13:55:56.234 message: Hello world version: 1.1 command: /w/w sh image_name: busybox

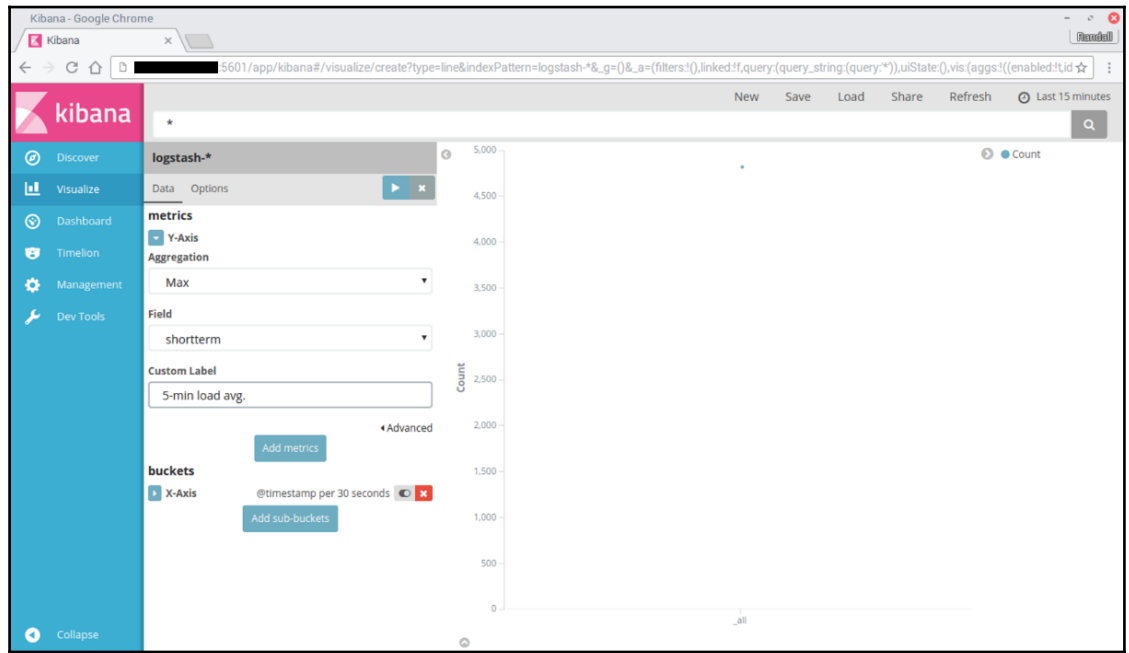
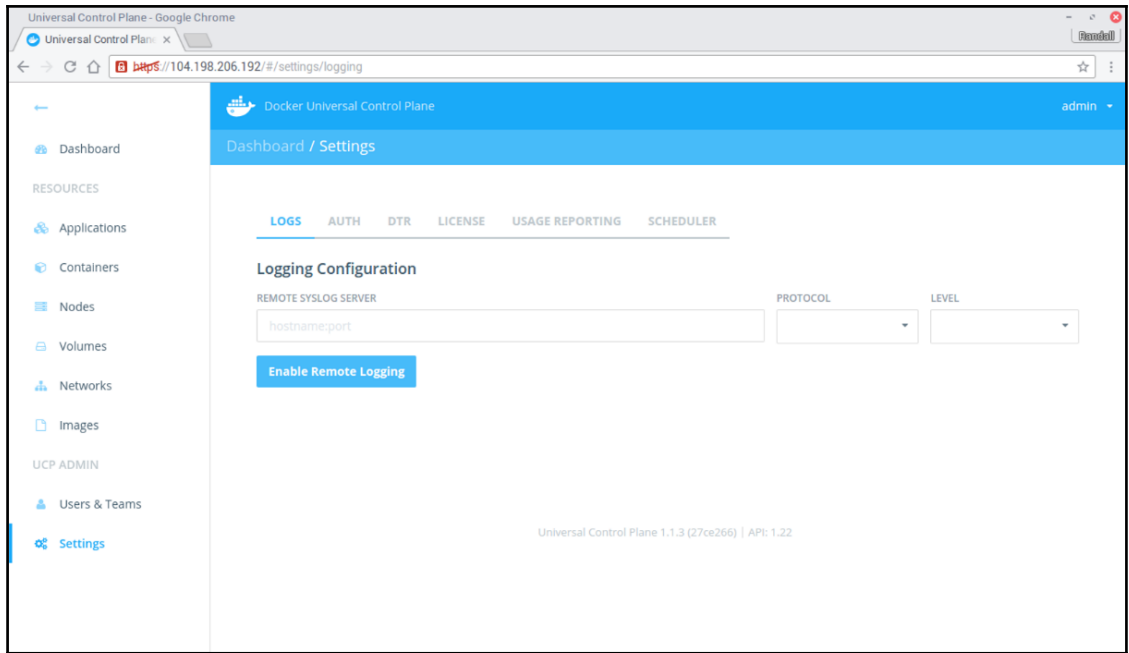
```

Kibana - Google Chrome

Doc: logstash-2016.11.03/logs/AVgqOxiPDDC6P_LqbsF8

Table JSON

@timestamp	November 3rd 2016, 06:47:45.372
t @version	1
t _id	AVgqOxiPDDC6P_LqbsF8
t _index	logstash-2016.11.03
# _score	1
t _type	logs
t command	/w/w /docker-entrypoint.sh kibana
t container_id	ad8e622068885260eaf84742de15b92d5fbf1850b860315433d88b6e8ef89943
t container_name	kibana
o created	November 2nd 2016, 18:55:43.718
t host	core-01
t image_id	sha256:57381b34e3ea8bc58c7d7051ce27ed25e1e9ff1072f499b143e96509d4f5b4a8
t image_name	kibana
# level	6
t message	{ "type": "response", "@timestamp": "2016-11-03T12:47:44Z", "tags": [], "pid": 10, "method": "post", "statusCode": 200, "req": { "url": "/elasticsearch/_msearch", "method": "post", "headers": { "host": "162.220.127.21:5601", "connection": "keep-alive", "content-length": "762", "accept": "application/json, text/plain, /*", "origin": "http://162.220.127.21:5601", "kbn-version": "5.0.0", "user-agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36", "content-type": "application/x-ldjson", "dnt": "1", "referrer": "http://162.220.127.21:5601/app/kibana", "accept-encoding": "gzip, deflate", "accept-language": "en-US,en;q=0.8", "remoteAddress": "10.0.2.2", "userAgent



Kibana - Google Chrome

Kibana

5601/app/kibana#/visualize/create?type=line&indexPattern=logstash-*&_g=()&_a=(filters:(),linked:!,query:(query_string:(query:*)),uiState:(),vis:(aggs:!((enabled:!,id *

New Save Load Share Refresh Last 15 minutes

kibana

Discover

Visualize

Dashboard

Timelion

Management

Dev Tools

Collapse

logstash-*

Data Options

metrics

Y-Axis

Add metrics

Count

buckets

X-Axis

Aggregation

Date Histogram

Field

@timestamp

Interval

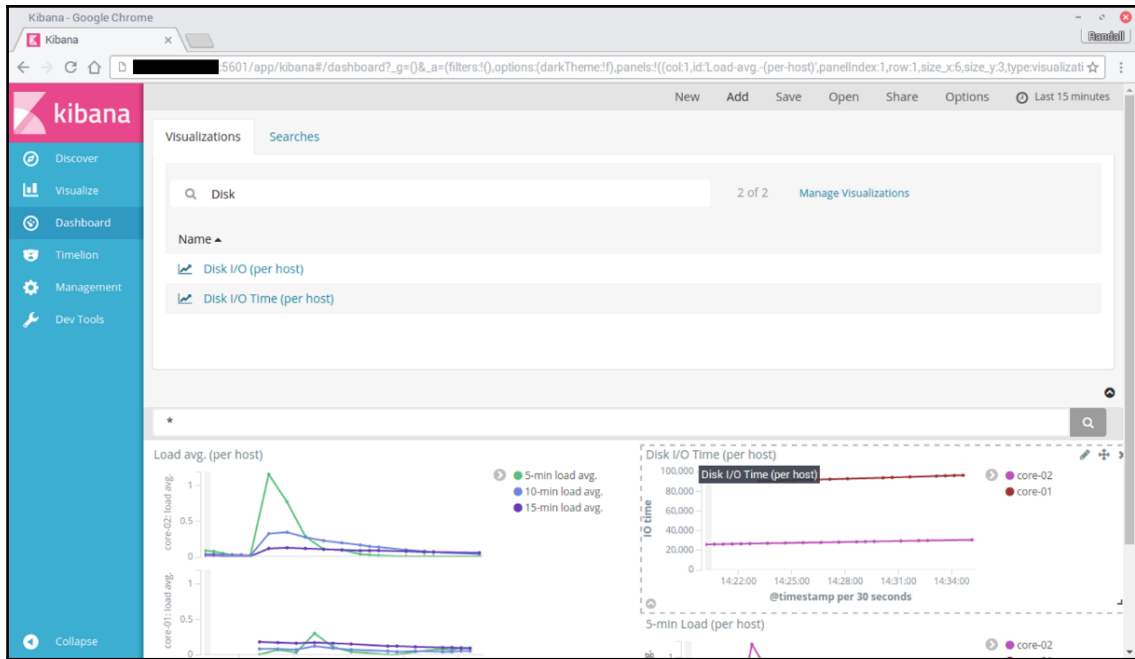
Auto

Custom Label

Advanced

Add sub-buckets

Count



Grafana - Home - Google Chrome

Grafana - Home 3000/datasources/new

Data Sources

Add data source

Name Default

Type

Http settings

Uri

Access

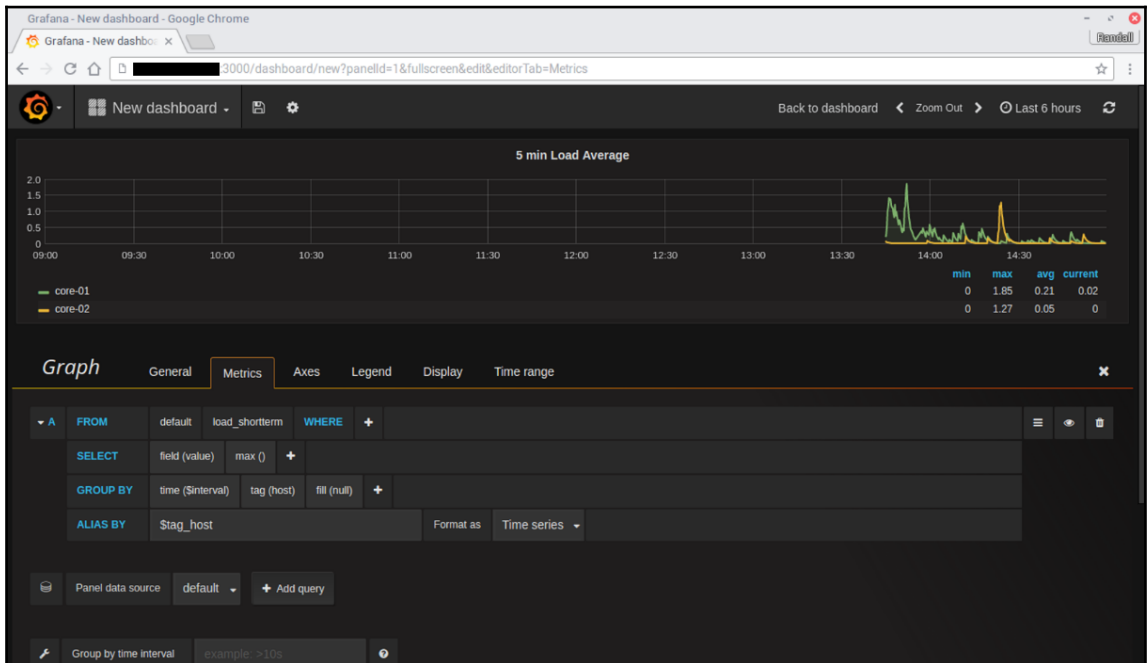
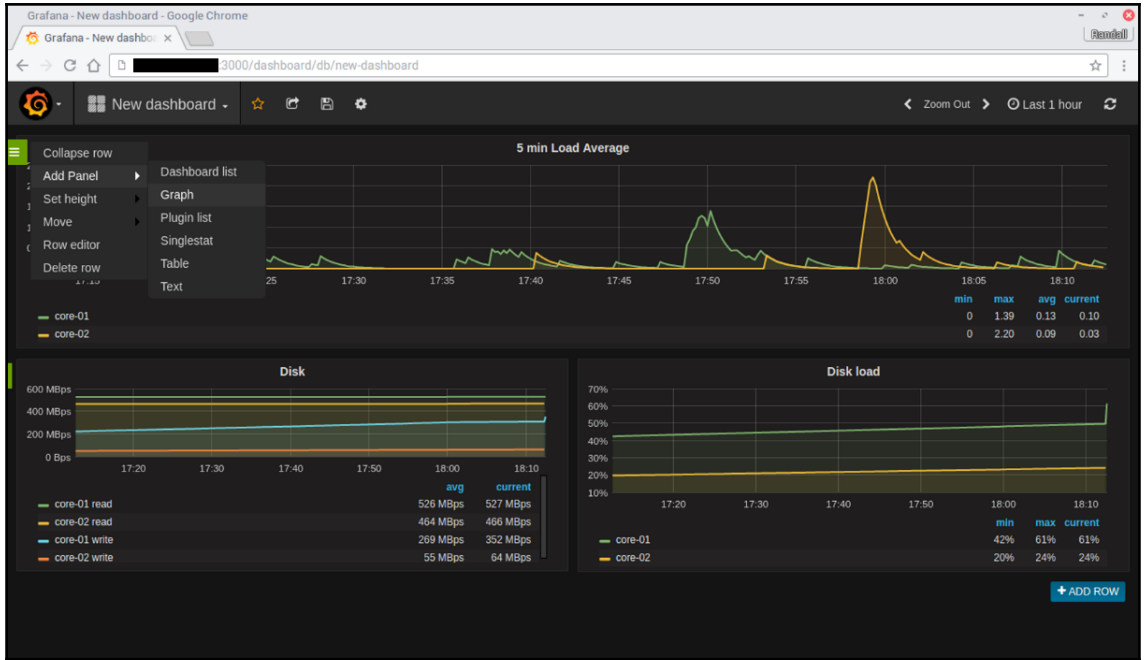
Http Auth With Credentials

InfluxDB Details

Database

User Password

Default group by time



```
Terminal - docker orch:2:ssh - "root@sysdig: /"
Viewing: Containers For: whole machine
Source: Live System Filter: container.name != host
CPU   PROCS  THREADS  VIRT  RES  FILE  NET  ENGINE  IMAGE  ID  NAME
5.50  0      6        509M  97M  0     0.00  docker  influxdb  ee6a3a9ff82a  influxdb
1.50  2      2        88M   25M  0     0.00  docker  sysdig/sysdig  3562529772a3  sysdig
0.00  1      6        192M  8M   0     0.00  docker  weaveworks/plugin:1.7.2  a6bc8f942c2a  weaveplugin
0.00  1      6        556M  27M  0     0.00  docker  grafana/grafana  1673d9ef3dd4  naughty_leavitt
0.00  1      7        208M  8M   0     3.11K  docker  weaveworks/weaveexec:1.7.2  0c1e6fa6e8a8  weaveproxy
0.00  1      10       349M  48M  0     1.20K  docker  weaveworks/weave:1.7.2  a2464df29437  weave
0.00  1      11       635M  3M   0     0.00  docker  registry.gitlab.com/peristalk  d0f2eb89d004  collectd-core-02

F1:Help F2:Views F4:Filter F5:Echo F6:Dig F7:Legend F8:Actions F9:Sort F12:Spectro CTRL+F:Search Pause 2/7(28.6%)
0:zsh 1:testcloud2- 2:ssh+ 3:core@core-02: 21:21
```

Chapter 9: Using Continuous Integration to Build, Test, and Deploy Containers

The screenshot shows the GitLab web interface for the 'Runners' settings of a project named 'ci-demo'. The browser address bar shows the URL `https://gitlab.com/perstalker/ci-demo/runners`. The page title is 'Runners - Settings - Randall Smith / ci-demo - GitLab - Google Chrome'. The navigation bar includes 'Project', 'Activity', 'Pipelines', 'Registry', 'Issues', 'Merge Requests', and 'Wiki'. A search bar is present with the text 'This project Search'. A settings menu is open on the right, listing options like 'Members', 'Groups', 'Deploy Keys', 'Webhooks', 'Services', 'Protected Branches', 'Runners', 'Variables', 'Triggers', 'CI/CD Pipelines', 'Push Rules', 'Mirror Repository', 'Pages', 'Audit Events', and 'Edit Project'. The main content area explains that a 'Runner' is a process which runs a build and can be in one of two states: 'active' (green) or 'paused' (red). It provides instructions on how to start serving builds by adding specific runners or using shared runners. Under 'Specific Runners', there is a section titled 'How to setup a specific Runner for a new project' with four steps: 1. Install a Runner compatible with GitLab CI (check out the [GitLab Runner section](#) for information on how to install it). 2. Specify the following URL during the Runner setup: `https://gitlab.com/ci`. 3. Use the following registration token during setup: `2f`. 4. Start the Runner! Under 'Shared Runners', it states that shared runners on GitLab.com run in autoscale mode, are on DigitalOcean, and autoscaling means reduced waiting times to start a build for each project, thus maximizing security. A link 'Read more information' is provided. There is a button 'Disable shared Runners' for this project. At the bottom, it shows 'Available shared Runners : 2' with a green dot and the ID '38dcea4b'.

Runners - Settings - Randall Smith / ci-demo - GitLab - Google Chrome

Runners - Settings - Randall Smith / ci-demo

<https://gitlab.com/perstalker/ci-demo/runners>

Runners - Settings - Randall Smith / ci-demo

Project Activity Pipelines Registry Issues Merge Requests Wiki

A 'Runner' is a process which runs a build. You can setup as many Runners as you need. Runners can be placed on separate users, servers, and even on your local machine.

Each Runner can be in one of the following states:

- active** - Runner is active and can process any new builds
- paused** - Runner is paused and will not receive any new builds

To start serving your builds you can either add specific Runners to your project or use shared Runners

Specific Runners

How to setup a specific Runner for a new project

1. Install a Runner compatible with GitLab CI (check out the [GitLab Runner section](#) for information on how to install it).
2. Specify the following URL during the Runner setup: `https://gitlab.com/ci`
3. Use the following registration token during setup: `2f`
4. Start the Runner!

Shared Runners

Shared Runners on GitLab.com run in autoscale mode, are on DigitalOcean. Autoscaling means reduced waiting times to start a build for each project, thus maximizing security. [Read more information](#) on how shared Runners are configured for GitLab.com.

[Disable shared Runners](#) for this project

Available shared Runners : 2

● 38dcea4b

Admin Area - GitLab - Google Chrome

Admin Area - GitLab

https://gitlab.example.com/admin/runners

Admin Area

Overview Monitoring Messages System Hooks Applications Abuse Reports 0

Overview Projects Users Groups Builds Runners

To register a new Runner you should enter the following registration token. With this token the Runner will request a unique Runner token and use that for future communication.
Registration token is `34`

You can reset runners registration token by pressing a button below.

Reset runners registration token

A 'Runner' is a process which runs a build. You can setup as many Runners as you need. Runners can be placed on separate users, servers, even on your local machine.

Each Runner can be in one of the following states:

- shared** - Runner runs builds from all unassigned projects
- specific** - Runner runs builds from assigned projects
- locked** - Runner cannot be assigned to other projects
- paused** - Runner will not receive any new builds

Runner description or token Search

Runners with last contact less than a minute ago: 1

Type	Runner token	Description	Projects	Builds	Tags	Last contact
------	--------------	-------------	----------	--------	------	--------------

Container Registry - Randall Smith / ci-demo - GitLab - Google Chrome

Container Registry - Randall Smith / ci-demo

https://gitlab.com/perlstalker/ci-demo/container_registry

Randall Smith / ci-demo

This project Search

Project Activity Repository Pipelines Registry Graphs Issues 0 Merge Requests 0 Wiki

A 'container image' is a snapshot of a container. You can host your container images with GitLab.
To start using container images hosted on GitLab you first need to login:

```
docker login registry.gitlab.com
```

Then you are free to create and upload a container image with build and push commands:

```
docker build -t registry.gitlab.com/perlstalker/ci-demo .
docker push registry.gitlab.com/perlstalker/ci-demo
```

Name	Image ID	Size	Created
build	9be08492f	68.2 MB · 4 layers	-
deploy	f0d49a964	83.6 MB · 6 layers	-
latest	f0d49a964	83.6 MB · 6 layers	-
master	b52987008	68.2 MB · 4 layers	-

Pipelines - Randall Smith / ci-demo - GitLab - Google Chrome

https://gitlab.com/perstalker/ci-demo/pipelines

Randall Smith / ci-demo

Project Activity Repository **Pipelines** Registry Graphs Issues 0 Merge Requests 0 Wiki

Pipelines Builds Environments Cycle Analytics

All 35 Running 0 Branches Tags Run pipeline CI Lint

Status	Pipeline	Commit	Stages	Time	Age	Actions
passed	#4969779 by latest	✓ deploy -> 9e40359f Proper "when" syntax	✓ ✓ ✓ ✓	05:05	2 days ago	▶
failed	#4969715 by yamll invalid	✓ deploy -> 58a00488 Add manual deployment			2 days ago	
passed	#4969656 by	✓ deploy -> 9901fc8d Use ./bin/sh	✓ ✓ ✓ ✓	04:23	2 days ago	
failed	#4969591 by	✓ deploy -> c32a494e Make deploy.sh executable	✓ ✓ ✓ ✗	04:36	2 days ago	⌵
failed	#4969519 by	✓ deploy -> 937f4ccc Add ./ to deploy script	✓ ✓ ✓ ✗	04:31	2 days ago	⌵
failed	#4969452 by	✓ deploy -> 1da524d7 Remove unnecessary pull	✓ ✓ ✓ ✗	05:15	2 days ago	⌵

Environments - Randall Smith / ci-demo - GitLab - Google Chrome

https://gitlab.com/perstalker/ci-demo/environments

Randall Smith / ci-demo

Project Activity Repository **Pipelines** Registry Graphs Issues 0 Merge Requests 0 Wiki

Pipelines Builds **Environments** Cycle Analytics

Available 2 Stopped 0 New environment

Environment	Last Deployment	Build	Commit	Time	Age	Actions
development	#2 by	deploy-to-dev (#6151182)	✓ deploy -> 9e40359f Proper "when" syntax		about a minute ago	▶ Re-deploy
production	#3 by	deploy-to-prod (#6151183)	✓ deploy -> 9e40359f Proper "when" syntax		less than a minute ago	Re-deploy

Pipeline - Randall Smith / ci-demo - GitLab - Google Chrome

https://gitlab.com/perlstalker/ci-demo/pipelines/4969779

Randall Smith / ci-demo

Project Activity Repository **Pipelines** Registry Graphs Issues 0 Merge Requests 0 Wiki

Pipelines Builds Environments Cycle Analytics

Pipeline #4969779 with 5 builds for `deploy` in 4 minutes 48 seconds (queued for 2 seconds) passed

Authored by **Randall Smith** 5 minutes ago

Commit `9e40359fda24f3dd3c995e341e74a9e1a42624b`

Proper "when" syntax

Hide pipeline graph

```
graph LR; subgraph Build; B1[build-image]; end; subgraph Test; T1[test-image]; end; subgraph Release; R1[release-image]; end; subgraph Deploy; D1[deploy-to-dev]; D2[deploy-to-prod]; end; B1 --> T1; T1 --> R1; R1 --> D1; R1 --> D2;
```

Status	Build ID	Name
passed		Build

Docker Cloud - Google Chrome

https://cloud.docker.com/app/perlstalker/repository/create

BUILD

- Repositories
- APPLICATIONS

 - Stacks
 - Services
 - Containers

- INFRASTRUCTURE

 - Node Clusters
 - Nodes

- SETTINGS

 - Cloud Settings

Description

```
new-repo: tagname  
$ docker push new-repo: tagname
```

Make sure to change tagname with your desired image repository tag.

Visibility

Using 0 of 1 private repositories. [Get more](#)

Public Anyone can see this repository

Private Only you can see this repository

Build Settings (optional)

Autobuild triggers a new build with every **git push** to your source code repository [Learn more](#)

Connected Disconnected

PerStalker x ci-demo x

Click here to customize the build settings

Cancel Create

Docker Cloud - Google Chrome

PerIStalker/ci-demo

https://cloud.docker.com/app/peristalker/repository/docker/peristalker/ci-demo/builds/edit

DOCKER CLOUD Back to classic UI + Get Help peristalker

Repositories / peristalker / ci-demo / Builds / Edit

General Tags Builds Timeline

Build configurations

SOURCE REPOSITORY

NOTE: Changing your source repository may affect your existing build rules.

BUILD LOCATION

Build on my own nodes

Build on Docker Cloud's infrastructure using a node ⓘ

AUTOTEST

Off

Internal Pull Requests

Internal and External Pull Requests

BUILD RULES +

Docker Cloud - Google Chrome

PerIStalker/ci-demo

https://cloud.docker.com/app/peristalker/service/wizard/config?image=peristalker/ci-demo

DOCKER CLOUD Back to classic UI + Get Help peristalker

Services / Wizard

General settings

IMAGE

SERVICE NAME

NICKNAME

ADD TO STACK

CONTAINERS

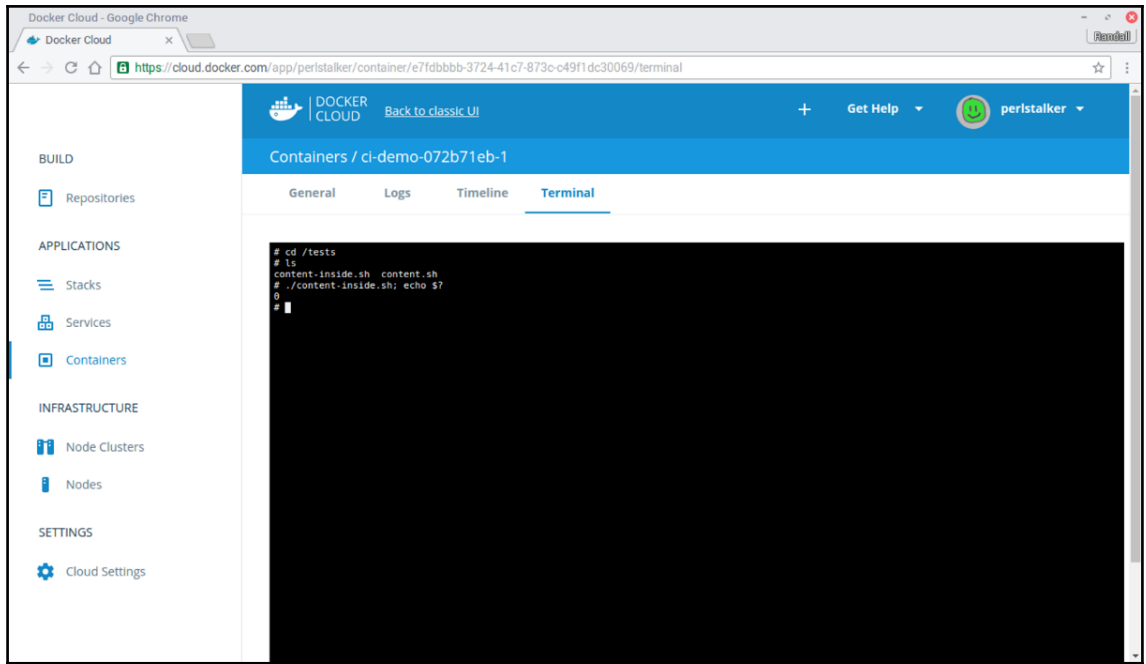
DEPLOYMENT STRATEGY

DEPLOYMENT CONSTRAINTS

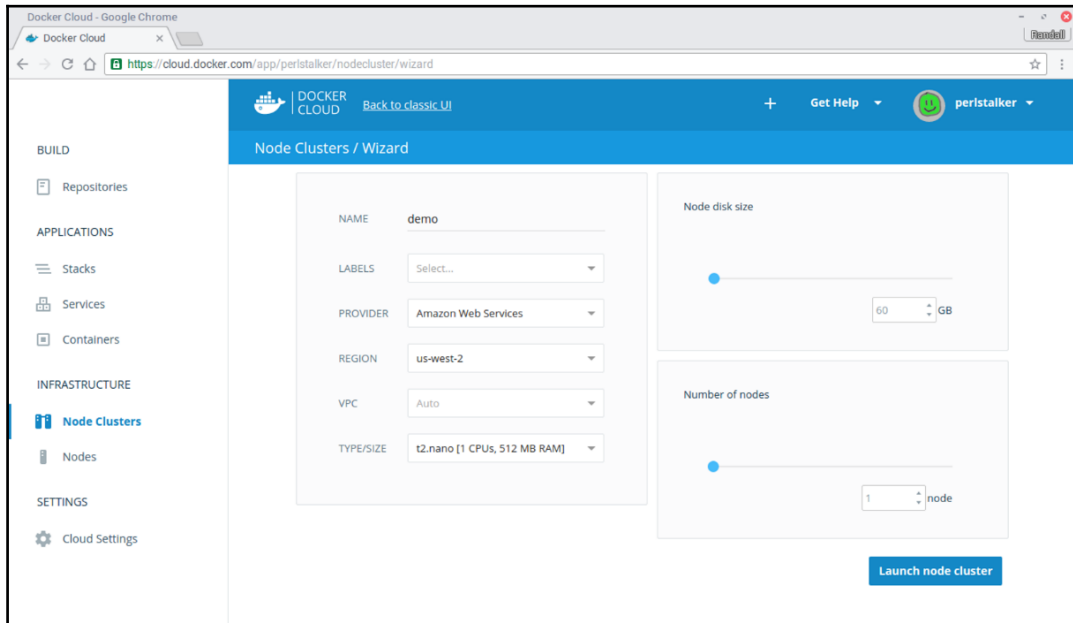
SUMMARY

- General settings
- Container configuration
- Ports
- Links
- Environment variables
- Volumes

Create & Deploy



Chapter 10: Why Stop at Containers? Automating Your Infrastructure



Docker Cloud - Google Chrome
Docker Cloud x
https://cloud.docker.com/app/perstalker/nodecluster/1064c71c-c323-4f79-a642-488b89a89da6/general

DOCKER CLOUD Back to classic UI + Get Help 2 perstalker

Node Clusters / demo Scale

General Timeline

Repositories

Stacks

Services

Containers

Node Clusters

Nodes

Cloud Settings

DEMO

DEPLOYED

a minute ago

REGION us-west-2

CPU 1

MEMORY 512 MB

DISK SPACE 0 GB

Nodes

b7fdf462-6ae1-4868-95f1-6... a minute ago

Terminate Deploy