



Building Cloud Native Applications on Cloud Foundry

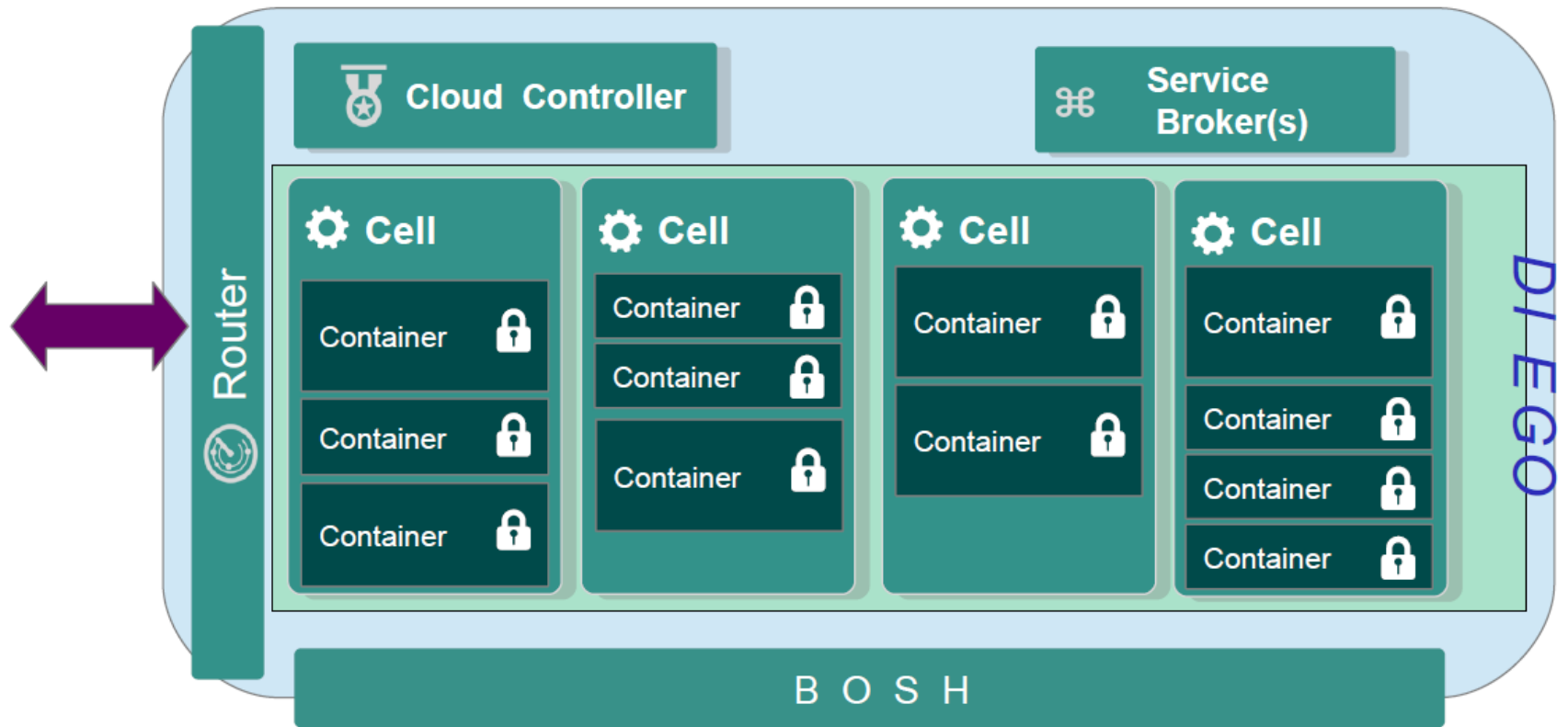
An in depth look at the microservices architecture pattern,
containers and Cloud Foundry

12: CFAR Overview

Objectives

- Explore the features of Cloud Foundry Application Runtime
- Define
 - Organization
 - Domain
 - Space
 - Route
 - Buildpack
 - Service
- Understand application deployment
- Experiment with the cf CLI

CFAR Architecture



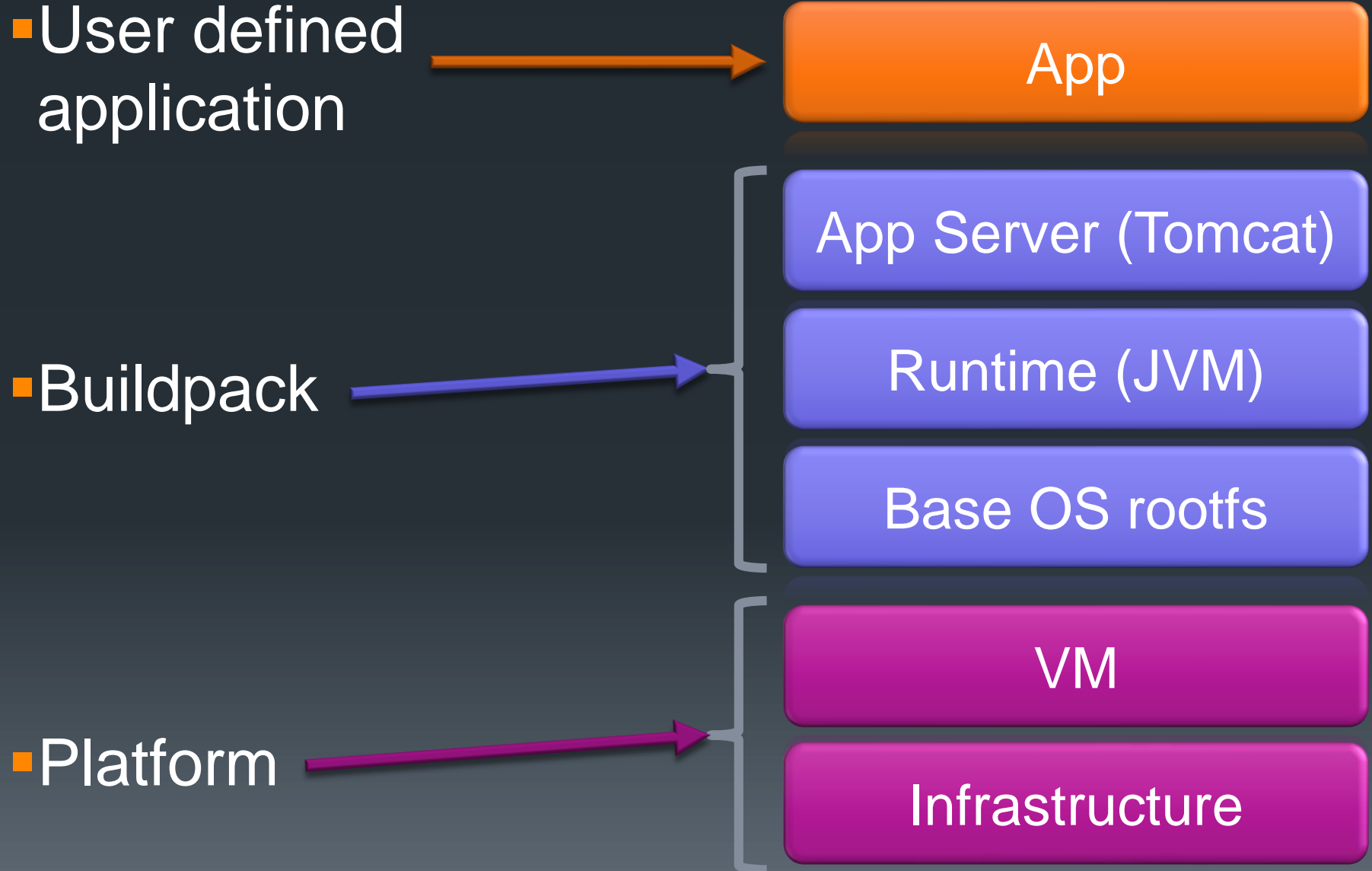
Cell = Execution Agent
= A VM where apps run



CFAR App Deployment

5

Copyright 2017 - 2018, RX-M LLC



Buildpacks

- Buildpacks define a specific **runtime environment**
- Each service you deploy must run in a buildpack
- **System Buildpacks** (built in to CloudFoundry):
 - Java, Python, Ruby, NodeJS, .Net Core, PHP, Go, Binaries, Static files
- **Community Buildpacks** (developed by other CloudFoundry users):
 - Haskell, Django, Perl, Jetty, JBoss, Websphere Liberty, Swift, etc.
- Many Heroku buildpacks work unmodified with CloudFoundry
- Using a build pack:
 - `$ cf push <app-name> -p artifact -b https://github.com/cloudfoundry/java-buildpack.git`

Manifests

7

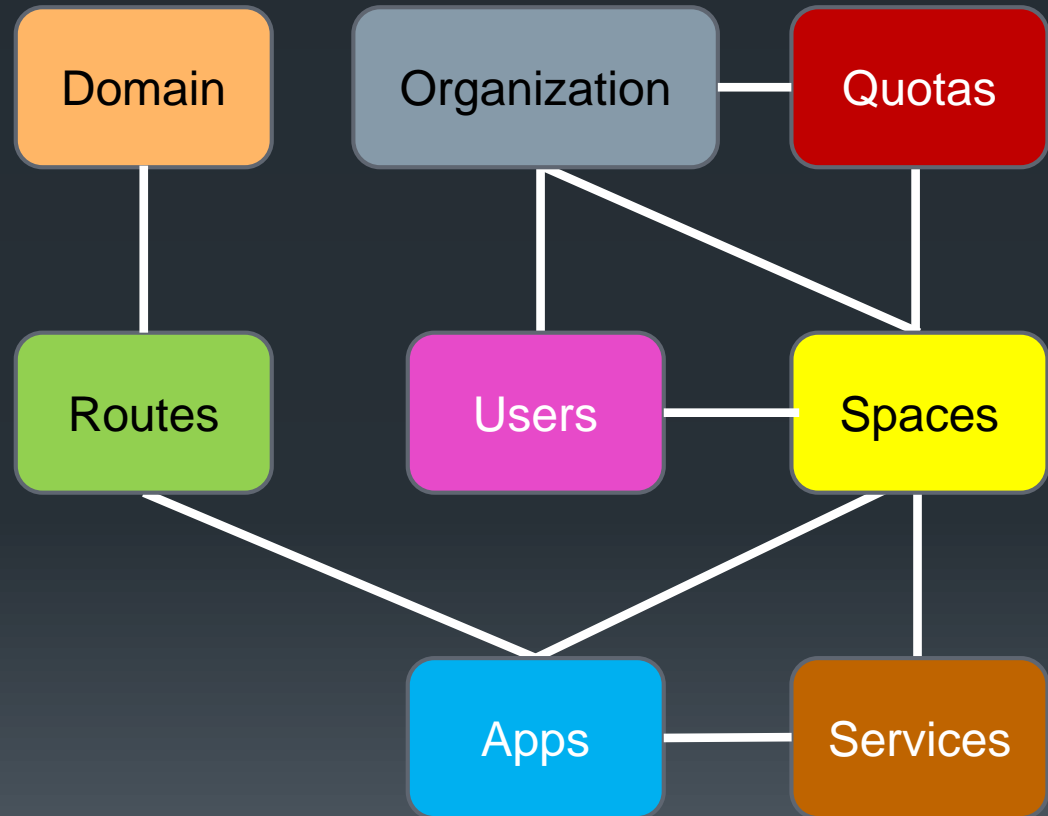
Copyright 2017 - 2018, RX-M LLC

- Manifests allow you to provide application deployment instructions to Cloud Foundry
- Manifests eliminate manual/imperative tasks
 - Defining all aspects of the deployment in code
 - Repeatable
 - Auditable
 - Can be checked into source code control

```
---
applications:
- name: my-python-mongo-app
  memory: 256M
  instances: 2
  path: .
  host: mpma
  domain: cfapps.io
  command: python mongo-app.py
  buildpack: https://github.com/rx-m/heroku-buildpack-python
  services:
  - my-mongo-db
```

Organizations, Users and Quotas

- An **organization** is an independently billable entity that an individual or multiple collaborators can share
- Collaborators access an organization with **user** accounts
- Collaborators in an org share a resource **quota** plan
- Organizations have one or more **spaces**, each of which provides an isolated namespace for services and applications
- Access to organizations and spaces is controlled with role based permissions

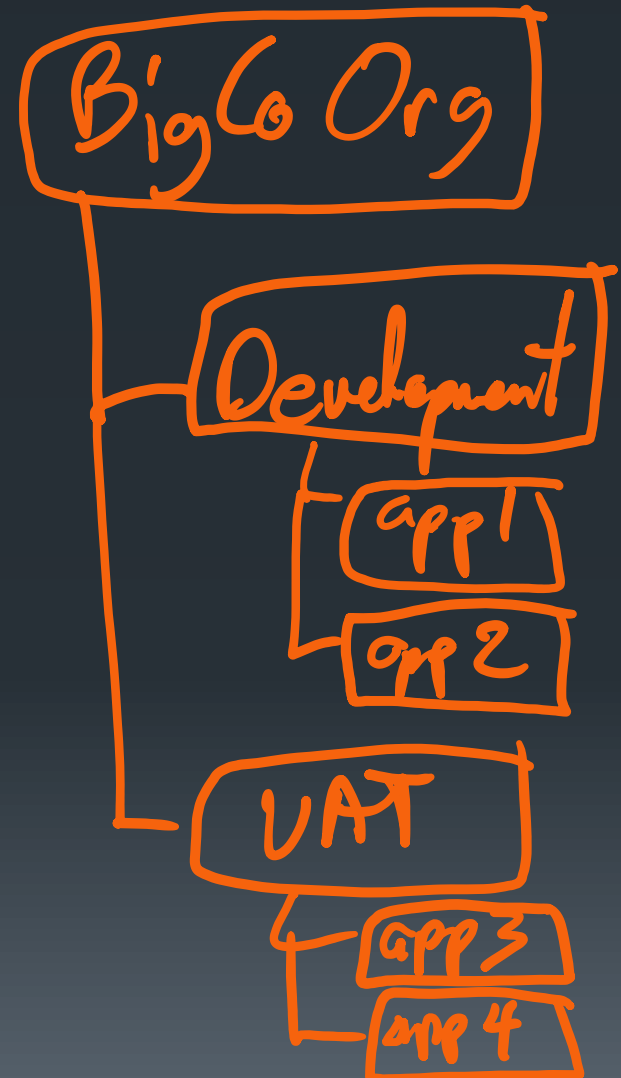


Org Roles and Permissions

- Org **Manager**
 - Managers or other users that administer the org
- Org **Auditor**
 - Auditors can view but not edit user information or quota usage information
- Org **User**
 - Users can view the list of other org users and their roles

Spaces

- Each Organization can container many spaces
 - The default PWS space is “development”
 - Administrators can create as many spaces as desired
- Users are given permissions on spaces
 - Developer privileges for creating and deploying applications
 - Administrative privileges for adding new users to the space
 - Auditing privileges for reading logs and examining configuration
- All applications and services live within some space



Space Roles

- Space **Manager**
 - Can invite/manage users, enable features for a given space
- Space **Developer**
 - Can create, delete, manage applications and services, full access to all usage reports and logs
- Space **Auditor**
 - View only access to all space information, settings, reports, logs

Domains

12

Copyright 2017 - 2018, RX-M LLC

- Deployed applications are associated with a URL
 - All requests to that URL redirect to the application
- Each Cloud Foundry instance has a default app domain
 - PWS has cfapps.io
- Custom Domains
 - You can register your own domain or use a preexisting domain with Cloud Foundry
- Subdomains
 - Each application has a unique subdomain
 - App URLs are subdomain.domain
 - E.g.: http://testweb.cfapps.io

```
user@ubuntu:~/cf$ curl webtest-1234.cfapps.io/status
```

```
Server A
```

```
user@ubuntu:~/cf$ cf apps
```

```
Getting apps in org rx-m.com / space development as randy.abernethy@rx-m.com...
```

```
OK
```

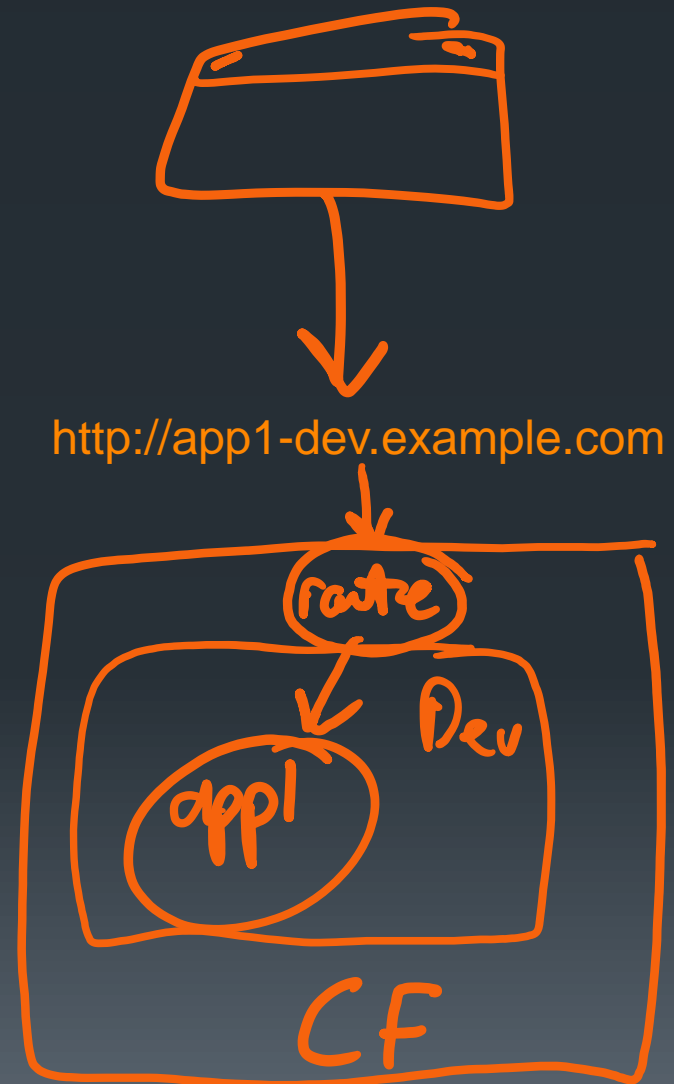
name	requested	state	instances	memory	disk	urls
webtest		started	1/1	512M	1G	webtest-1234.cfapps.io

Routes

13

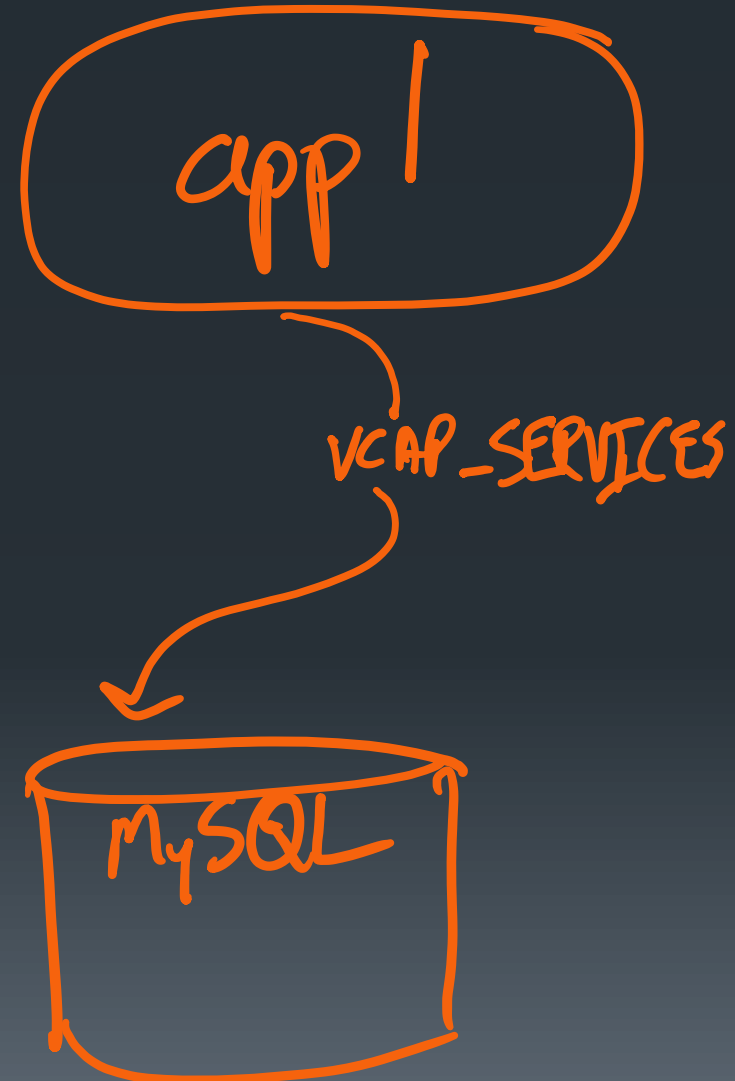
Copyright 2017 - 2018, RX-M LLC

- Routes provide a path from an Internet facing end point to a given application
 - A unique route exists to each application in each space
 - If multiple copies of a given application are running the route will load balance the instances as a group
- DNS domains can be shared by multiple spaces
 - App1 in the development space:
 - `http://app1-dev.example.com`
 - App1 in the UAT space:
 - `http://app1-uat.example.com`
 - You can use one or more of you own domain names in a given Cloud Foundry organization
 - PWS provides a DNS domain that you can use if you do not have your own:
 - `cfapps.io`
 - Shared by all users so prefixes must be unique



Services

- In Cloud Foundry a service is a feature provided by the platform that your application can consume
 - A message queue
 - A database
 - Etc.
- Services can be consumed by multiple applications
 - Connecting a service to an application is called “binding” in Cloud Foundry
 - Applications discover connection information and login credentials through the VCAP_SERVICES environment variable
 - All runtime specific application settings must be injected through environment variables
- Application state should generally be housed in platform services
 - Storage within an application container is ephemeral



- Pivotal Web Services (PWS)
 - Pivotal fully hosted and managed Cloud Foundry solution
 - <http://run.pivotal.io>
- Accounts can be created for free
 - Apps manager URL: console.run.pivotal.io
 - API URL: api.run.pivotal.io

The screenshot shows the Pivotal Web Services console interface. The left sidebar contains navigation links: ORG (rx-m.com), SPACES (development, UAT), Marketplace, Docs, Support, Tools, Blog, and Status. The main content area displays the 'development' space with a summary of app status: 1 RUNNING, 0 STOPPED, and 0 CRASHED. Below this, there are tabs for App (1), Services, Route (1), Members (2), and Settings. The 'App (1)' tab is active, showing a table of apps.

Status	Name	Instances	Memory	Last Push	Route
Running	webtest	1	512 MB	15 hours ago	https://webtest-1234.cfapps.io

Installing the CF CLI

16

Copyright 2017 - 2018, RX-M LLC

```
user@ubuntu:~$ wget -q -O - https://packages.cloudfoundry.org/debian/cli.cloudfoundry.org.key | sudo apt-key add -

user@ubuntu:~$ echo "deb http://packages.cloudfoundry.org/debian stable main" | sudo tee /etc/apt/sources.list.d/cloudfoundry-cli.list

deb http://packages.cloudfoundry.org/debian stable main

user@ubuntu:~$ sudo apt-get update

Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:6 https://download.docker.com/linux/ubuntu xenial InRelease
Ign:4 https://cf-cli-debian-repo.s3.amazonaws.com stable InRelease
Get:7 https://cf-cli-debian-repo.s3.amazonaws.com stable Release [1,797 B]
Get:8 https://cf-cli-debian-repo.s3.amazonaws.com stable Release.gpg [819 B]
Get:9 https://cf-cli-debian-repo.s3.amazonaws.com stable/main amd64 Packages [6,849 B]
Get:10 https://cf-cli-debian-repo.s3.amazonaws.com stable/main i386 Packages [6,851 B]
Fetched 16.3 kB in 1s (12.4 kB/s)
Reading package lists... Done

user@ubuntu:~$ sudo apt-get install cf-cli

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  cf-cli
0 upgraded, 1 newly installed, 0 to remove and 230 not upgraded.
Need to get 4,344 kB of archives.
After this operation, 17.3 MB of additional disk space will be used.
Get:1 http://packages.cloudfoundry.org/debian stable/main amd64 cf-cli amd64 6.34.1 [4,344 kB]
Fetched 4,344 kB in 4s (1,024 kB/s)
Selecting previously unselected package cf-cli.
(Reading database ... 125146 files and directories currently installed.)
Preparing to unpack .../cf-cli_6.34.1_amd64.deb ...
Unpacking cf-cli (6.34.1) ...
Setting up cf-cli (6.34.1) ...

user@ubuntu:~$ cf -version

cf version 6.34.1+bbdf81482.2018-01-17

user@ubuntu:~$
```


Help

- The cf help command provides general help:
 - cf help
- You can get help with specific commands using the form:
 - cf help <cmd>
- To get full help add the -a switch
 - cf help -a

```
user@ubuntu:~$ cf help push
NAME:
  push - Push a new app or sync changes to an existing app

USAGE:
  cf push APP_NAME [-b BUILDPACK_NAME] [-c COMMAND] [-f MANIFEST_PATH]
  [-i NUM_INSTANCES] [-k DISK] [-m MEMORY] [-p PATH] [-s STACK] [-t HEALTH_TIMEOUT]
  [--no-route | --random-route | --hostname HOST | --no-hostname]

  cf push APP_NAME --docker-image [REGISTRY_HOST:PORT/]IMAGE[:TAG]
  [-c COMMAND] [-f MANIFEST_PATH | --no-manifest] [--no-start]
  [-i NUM_INSTANCES] [-k DISK] [-m MEMORY] [-t HEALTH_TIMEOUT] [-u]
  [--no-route | --random-route | --hostname HOST | --no-hostname]

  cf push -f MANIFEST_WITH_MULTIPLE_APPS_PATH [APP_NAME] [--no-start]

ALIAS:
  p

OPTIONS:
  -b BUILDPACK_NAME Custom buildpack by name (e.g. my-buildpack)
  -f MANIFEST_PATH Path to a manifest file (e.g. 'https://github.com/cloudfoundry/java-buildpack')
  -i NUM_INSTANCES Number of instances to start
  -k DISK Disk space in MB
  -m MEMORY Memory in MB
  -p PATH Path to application source
  -s STACK Stack name
  -t HEALTH_TIMEOUT Health check timeout in seconds
  --no-route Do not create a route
  --random-route Create a random route
  --hostname HOST Hostname for the route
  --no-hostname Do not create a hostname
  -c COMMAND Startup command, set to null to reset
```

```
user@ubuntu:~$ cf help
cf version 6.35.2+88a03e995.2018-03-15, Cloud Foundry command line tool
Usage: cf [global options] command [arguments...] [command options]

Before getting started:
  config      login,l      target,t
  help,h      logout,lo

Application lifecycle:
  apps,a      run-task,rt   events
  push,p      logs           set-env,se
  start,st    ssh           create-app-manifest
  stop,sp     app
  restart,rs  env,e
  restage,rg  scale

Services integration:
  marketplace,m      create-user-provided-service,cups
  services,s         update-user-provided-service,uups
  create-service,cs  create-service-key,csk
  update-service    delete-service-key,dsk
  delete-service,ds service-keys,sk
  service          service-key
  bind-service,bs  bind-route-service,brs
  unbind-service,us unbind-route-service,urs

Route and domain management:
  routes,r      delete-route   create-domain
  domains      map-route
  create-route  unmap-route

Space management:
  spaces      create-space   set-space-role
  space-users delete-space   unset-space-role

Org management:
  orgs,o      set-org-role
  org-users   unset-org-role

CLI plugin management:
  plugins      add-plugin-repo   repo-plugins
  install-plugin list-plugin-repos

Commands offered by installed plugins:

Global options:
  --help, -h      Show help
  -V              Print API request diagnostics to stdout

Use 'cf help -a' to see all commands.
user@ubuntu:~$
```

Login

```
user@ubuntu:~$ cf login -a api.run.pivotal.io randy.abernethy@rx-m.com
API endpoint: api.run.pivotal.io

Email> randy.abernethy@rx-m.com

Password>
Authenticating...
OK

Targeted org rx-m.com

Targeted space development


API endpoint: https://api.run.pivotal.io (API version: 2.101.0)
User: randy.abernethy@rx-m.com
Org: rx-m.com
Space: development
user@ubuntu:~$
```

CF URLs

- PWS URLs

- System domain: run.pivotal.io
 - API Endpoint: api.run.pivotal.io
 - Apps Manager: console.run.pivotal.io
- Apps domain: cfapps.io

- On Prem Cloud Foundry (PCF)

- System domain: <your configured CF system domain>
 - API Endpoint: api.<your configured CF system domain>
 - Apps Manager: login.<your configured CF system domain>
- Apps domain: <your configured CF app domain>

CF state store

20

Copyright 2017 - 2018, RX-M LLC

- Cloud Foundry login data, configuration and plugins are stored in the .cf directory
- The cf target command displays current config

```
user@ubuntu:~$ ls -la ~/.cf
total 12
drwx-----  2 user user 4096 Feb  1 19:25 .
drwxr-xr-x 16 user user 4096 Feb  1 19:24 ..
-rw-----  1 user user 3494 Feb  1 19:29 config.json
user@ubuntu:~$ cf target
api endpoint:      https://api.run.pivotal.io
api version:       2.101.0
user:              randy.abernethy@rx-m.com
org:               rx-m.com
space:             development
user@ubuntu:~$
```

Displaying orgs and spaces

```
user@ubuntu:~$ cf orgs
Getting orgs as randy.abernethy@rx-m.com...

name
rx-m.com
user@ubuntu:~$ cf spaces
Getting spaces in org rx-m.com as randy.abernethy@rx-m.com...

name
development
user@ubuntu:~$
```

Creating and setting (targeting) spaces

```
user@ubuntu:~/cf$ cf create-space UAT
Creating space UAT in org rx-m.com as randy.abernethy@rx-m.com...
OK
Assigning role RoleSpaceManager to user randy.abernethy@rx-m.com in org rx-m.com / space UAT as randy.abernethy@rx-m.com...
OK
Assigning role RoleSpaceDeveloper to user randy.abernethy@rx-m.com in org rx-m.com / space UAT as randy.abernethy@rx-m.com...
OK
TIP: Use 'cf target -o "rx-m.com" -s "UAT"' to target new space
user@ubuntu:~/cf$
```

Deploying Applications

- Cloud Foundry apps are deployed with the “cf push” command
 - `$ cf push myapp`
 - “cf p” for short
- Push options:
 - -i number of instances
 - -m Memory limit (256M, 1024M, 1G)
 - -n Host Name (app subdomain)
 - -p Local path to app directory (jar, war, *.js, zip, etc.)
 - -b Custom buildpack URL (usually a github/bitbucket URL)
 - Not required for built-in buildpacks
 - --health-check-type, -u Health check type
 - Port (default)
 - None
 - Process
 - Http
 - Many others

A Node JS App

```
user@ubuntu:~/cf$ cat test.js
var express = require('express');
var http = require('http');

var app = express();

app.get('/status', function(req, res) {
  return res.send('Server A\n');
});

http.createServer(app).listen(process.env.PORT || 9090, function() {
  console.log('Listening on port 9090');
});

user@ubuntu:~/cf$
```


Dependencies

25

Copyright 2017 - 2018, RX-M LLC

```
user@ubuntu:~/cf$ ls -l
total 12
drwxrwxr-x 31 user user 4096 Feb  1 20:18 node_modules
-rw-rw-r--  1 user user  152 Feb  1 20:19 package.json
-rw-rw-r--  1 user user  282 Feb  1 20:09 test.js
user@ubuntu:~/cf$ cat package.json
{
  "name": "first",
  "version": "0.0.1",
  "author": "Demo",
  "dependencies": {
    "express": "3.4.8"
  },
  "engines": {
    "node": "4.8.6"
  }
}
user@ubuntu:~/cf$
```

cf push [1]

```
user@ubuntu:~/cf$ cf push webtest -i 1 -m 512M -c "node test.js" -n webtest-1234
Pushing app webtest to org rx-m.com / space development as randy.abernethy@rx-m.com...
Getting app info...
Updating app with these attributes...
  name:                webtest
  path:                /home/user/cf
  command:             node test.js
  disk quota:          1G
  health check type:   port
  instances:           1
  memory:              512M
  stack:               cflinuxfs2
  routes:
    webtest-1234.cfapps.io

Updating app webtest...
Mapping routes...
Comparing local files to remote cache...
Packaging files to upload...
Uploading files...
 387.34 KiB / 387.34 KiB [=====] 100.00% 1s

Waiting for API to complete processing files...

Staging app and tracing logs...
  Downloading binary_buildpack...
  Downloading nodejs_buildpack...
```

cf push [2]

27

```
Uploading droplet, build artifacts cache...
Uploading build artifacts cache...
Uploading droplet...
Uploaded build artifacts cache (278B)
Uploaded droplet (13.6M)
Uploading complete
Stopping instance 44b806c4-cfa4-4a90-89b3-d3d01d539ad2
Destroying container
```

Waiting for app to start...

```
name:                webtest
requested state:     started
instances:           1/1
usage:               512M x 1 instances
routes:              webtest-1234.cfapps.io
last uploaded:       Thu 01 Feb 20:19:58 PST 2018
stack:               cflinuxfs2
buildpack:           nodejs
start command:       node test.js
```

	state	since	cpu	memory	disk	details
#0	running	2018-02-02T04:20:28Z	0.0%	0 of 512M	0 of 1G	

```
user@ubuntu:~/cf$
```

Monitoring Apps

```
user@ubuntu:~/cf$ curl webtest-1234.cfapps.io/status
```

Server A

```
user@ubuntu:~/cf$ cf apps
```

Getting apps in org **rx-m.com** / space **development** as **randy.abernethy@rx-m.com...**

OK

name	requested state	instances	memory	disk	urls
webtest	started	1/1	512M	1G	webtest-1234.cfapps.io

```
user@ubuntu:~/cf$ cf logs webtest
```

Retrieving logs for app **webtest** in org **rx-m.com** / space **development** as **randy.abernethy@rx-m.com...**

```
2018-02-01T20:27:35.96-0800 [RTR/9] OUT webtest-1234.cfapps.io - [2018-02-02T04:27:35.957+0000] "GET /status HTTP/1.1" 200 0 9 "-" "curl/7.47.0" "10.10.2.158:52630" "10.10.149.157:61028" x_forwarded_for:"63.98.155.67, 10.10.2.158" x_forwarded_proto:"http" vcap_request_id:"26d48e0a-d488-4402-69bb-7e4e13fd6cf1" response_time:0.006506819 app_id:"8d01c6ce-1e9d-4280-9b01-0ee3f22f53db" app_index:"0" x_b3_traceid:"fe4727821b5bdcfe" x_b3_spanid:"fe4727821b5bdcfe" x_b3_parentspanid:"-"  
2018-02-01T20:27:35.96-0800 [RTR/9] OUT
```

Scaling Apps

- Changing the instance count (scaling horizontally):
 - `$ cf scale <app> -i <new-value>`
 - To scale myapp to 5 instances:
 - `$ cf scale myapp -i 5`
 - Will start or stop instances as necessary
- Changing instance memory allocation (scaling vertically)
 - `$ cf scale <app> -m <new-memory>`
 - To scale myapp to 1 gig
 - `$ cf scale myapp -m 1G`
 - The instance must be restarted before the new size will take effect

Controlling Apps

- `cf stop` Sends SIGTERM, 10 seconds later sends SIGKILL
- `cf start` Starts app
- `cf restart` Starts and stops an app
- `cf restage` Redeploys the app and starts it
(required to pick up config changes)

CF Web

31

Copyright 2017 - 2018, RX-M LLC

The screenshot shows the Pivotal Web Services console interface. The browser address bar displays the URL: `https://console.run.pivotal.io/organizations/aaa14634-a3f1-441e-9cae-7e237120db36/spaces/3ce1f344-0746-44c7-9140-ec6baa4b8...`. The page header includes the Pivotal Web Services logo, a search bar with the text "Search by App Name", and the user email "randy.abernethy@rx-m.com".

On the left sidebar, the "ORG" section shows "rx-m.com" selected. Under the "SPACES" section, "development" is selected. The sidebar also lists "Marketplace", "Docs", "Support", "Tools", "Blog", and "Status".

The main content area shows the "development" space summary with status indicators: RUNNING (1), STOPPED (0), and CRASHED (0). Below this, there are tabs for "App (1)", "Services", "Route (1)", "Members (2)", and "Settings". The "Apps" tab is active, displaying a table with the following data:

Status	Name	Instances	Memory	Last Push	Route
Running	webtest	1	512 MB	10 minutes ago	https://webtest-1234.cfapps.io

The footer contains the Pivotal logo, copyright notice "© 2018 Pivotal Software Inc. All rights reserved.", links for "Terms" and "Privacy", and a "GIVE FEEDBACK" button.



ORG

rx-m.com

SPACES

development

Marketplace

Docs

Support

Tools

Blog

Status

APP

webtest



● Running

VIEW APP

Overview

Services

Route (1)

Logs

Tasks

Settings

Buildpack: N/A

Events

Last Push: 08:20 PM 02/01/18

Scaling

CANCEL

SCALE APP



Started app

randy.abernethy@rx-m.com 02/02/2018 at
04:20:06 AM UTC

Stopped app

randy.abernethy@rx-m.com 02/02/2018 at
04:19:57 AM UTC

Started app

randy.abernethy@rx-m.com 02/02/2018 at
04:18:41 AM UTC

Stopped app

randy.abernethy@rx-m.com 02/02/2018 at
04:18:32 AM UTC

Stopped app

randy.abernethy@rx-m.com 02/02/2018 at
04:16:14 AM UTC

Stopped app

randy.abernethy@rx-m.com 02/02/2018 at
04:14:39 AM UTC

Mapped route to app

randy.abernethy@rx-m.com 02/02/2018 at
04:13:42 AM UTC

Created app

randy.abernethy@rx-m.com 02/02/2018 at
04:13:42 AM UTC

Instances

1

Memory Limit

512 MB



Disk Limit

1 GB

☐ Autoscaling

Instances

View in PCF Metrics

#	CPU	Memory	Disk	Uptime
0	0%	18.69 MB	44.69 MB	10 min

Lab 12

- Cloud Foundry Application Runtime

13: Garden

Objectives

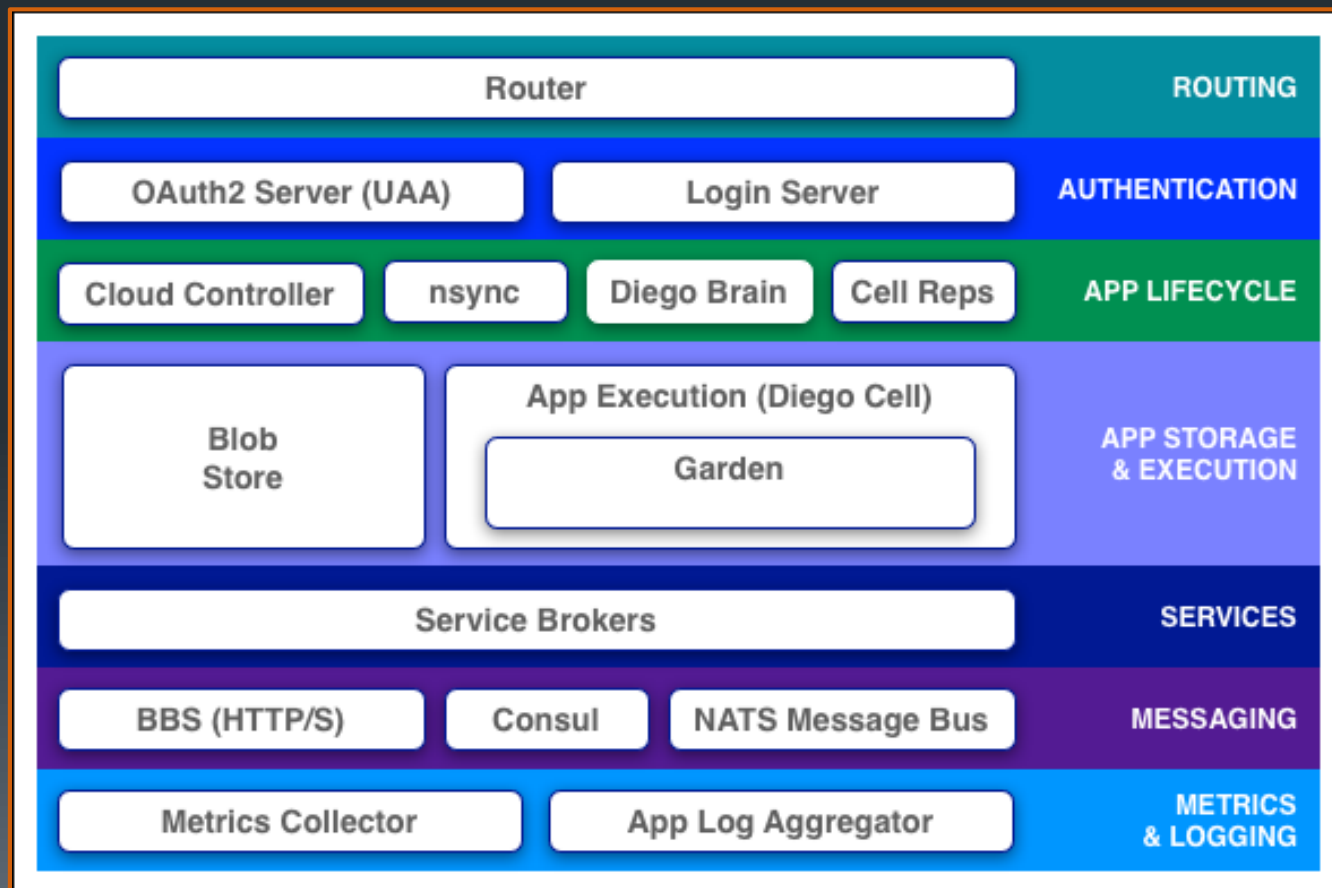
- Understand CF container tech and architecture
- Explain the role of Garden
- List the Linux kernel features used to isolate Garden containers
- Describe Garden images
- Map out the networking model of a garden container

Garden Containers

36

Copyright 2017 - 2018, RX-M LLC

- Garden containers are the self-contained environments used to encapsulate an app deployed to CF
 - The Garden container manager is a Go language rewrite of the old Ruby based Warden container manager
- Containers isolates processes, memory, and the filesystem using operating system namespaces and cgroups
- Each container includes
 - Process ID (PID) namespace
 - Network namespace
 - Mount namespace
- Resource control is managed using Linux control groups
 - Limits the amount of memory that the container may use
 - Requires the container to use a fair share of CPU

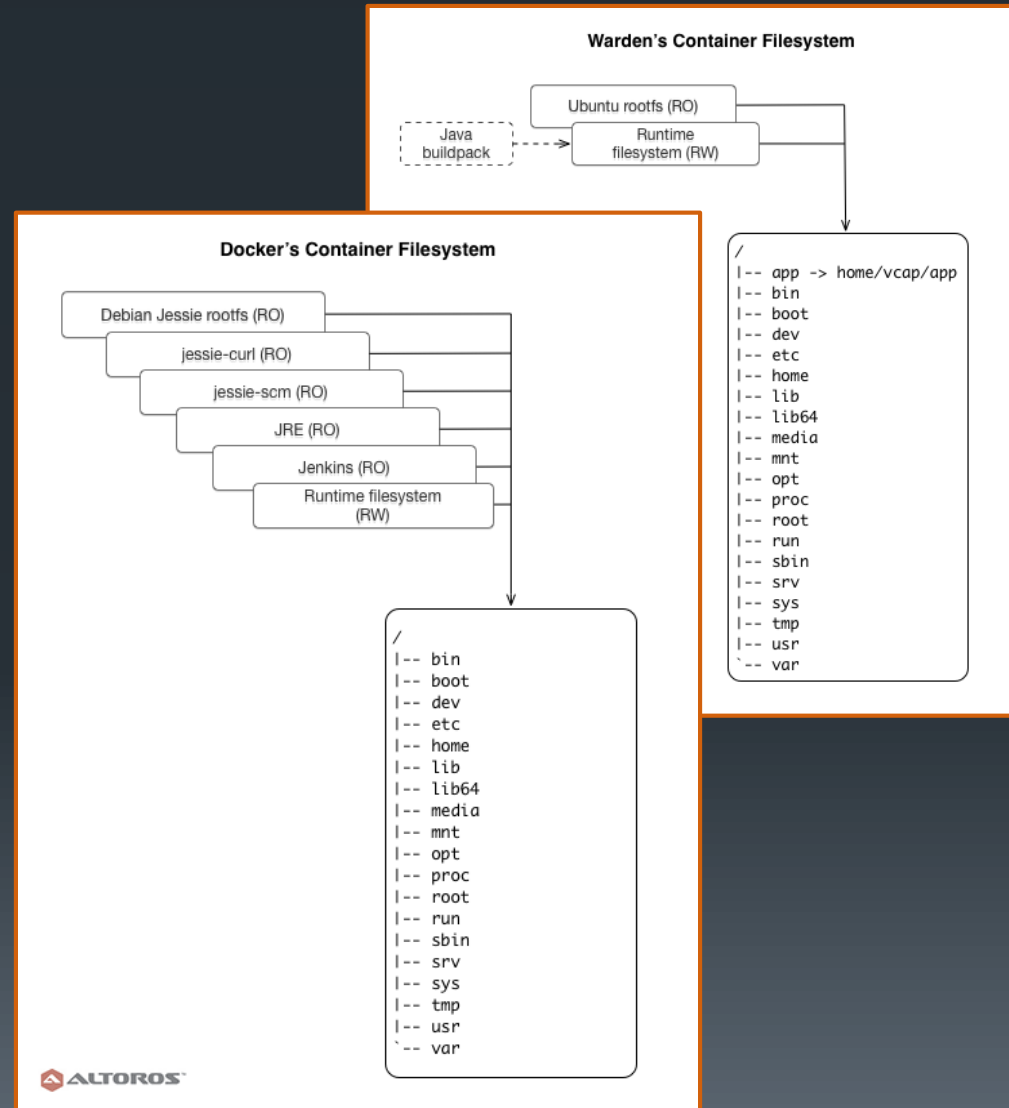


Docker versus CF Containers

37

Copyright 2017 - 2018, RX-M LLC

- The main distinction between Docker containers and CF containers is the image structure used
- CF is designed to run applications that get all their dependencies from buildpacks
- CF containers have two layers:
 - a read-only layer with an OS root file system (The buildpack)
 - e.g. Ubuntu 14.04
 - a nonpersistent, read/write layer for the application
 - Dependencies and temporary data
 - Only this image can be run
- Docker containers can have 1 or more layers and each layer can be used to start a container

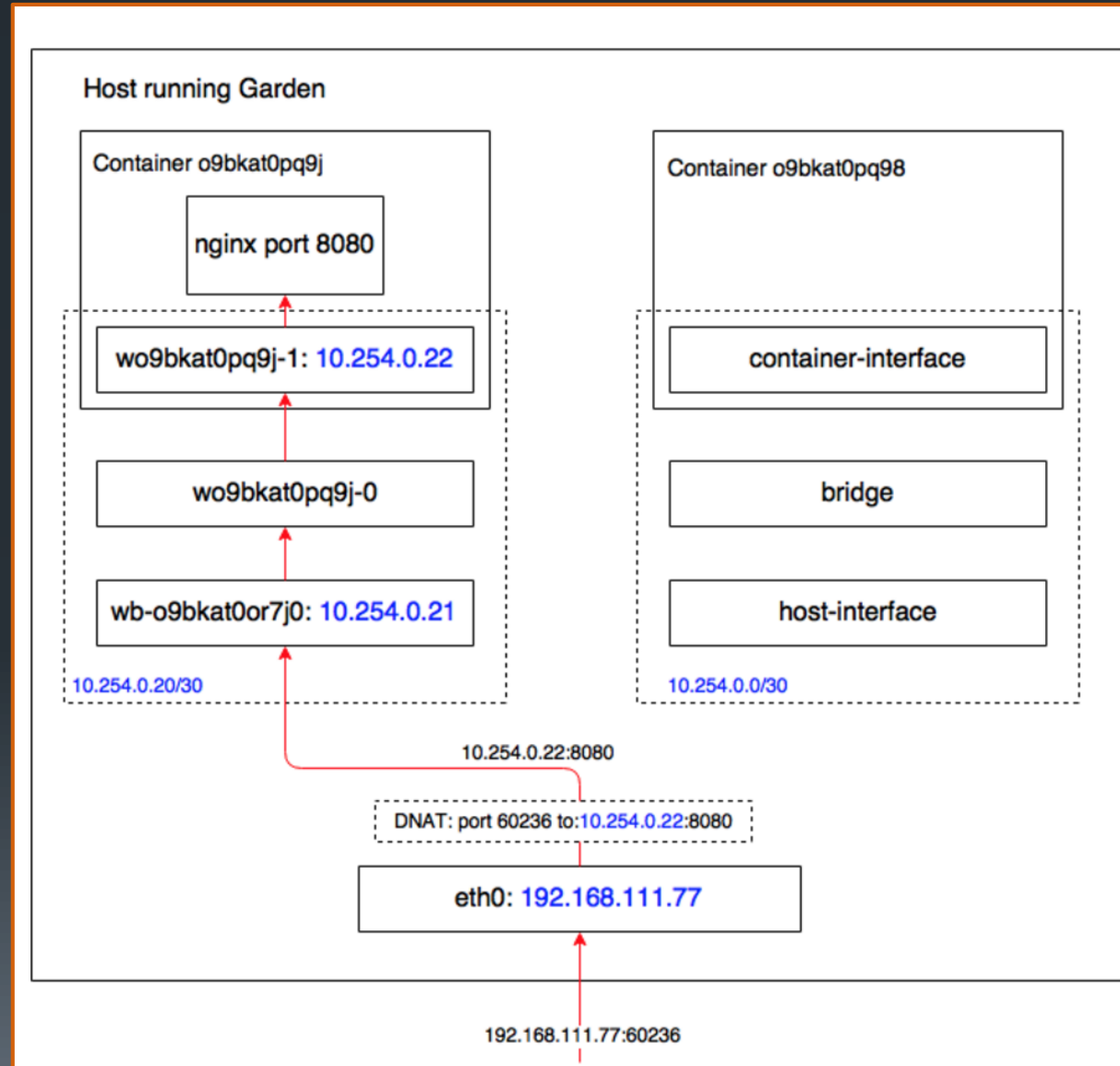


Garden Networking

38

Copyright 2017 - 2018, RX-M LLC

- Networks in containers are isolated from a host with the Linux network namespace
- Garden creates a dedicated /30 network for each container with two IP addresses:
 - One for the host
 - One for the container
- Garden uses “network address translation” (NAT) based on a port to forward traffic from the host to the container
- Garden randomly picks an unused port and then adds a rule to iptables to forward traffic to the container



Using Docker with CF

- Garden can run OCI compliant containers
- Garden can also pull and run Docker images if properly prepared

CF Docker Requirements

- To push apps with Docker, CF requires the following:
 - A Cloud Foundry (CF) deployment that has Docker support enabled
 - A Docker image that meets the following requirements:
 - Must contain an `/etc/passwd` file with an entry for the root user
 - The home directory and the shell for that root user must be present in the image file system
 - The total size of the Docker image file system layers must not exceed the disk quota for the app
 - The maximum disk allocation for apps is set by the Cloud Controller (default is 2048 MB)
 - If the total size of the Docker image file system layers exceeds the disk quota, the app instances will not start
 - The location of the Docker image on Docker Hub or another Docker registry
 - The registry must support Docker Registry API V2
 - The registry must present a valid certificate
- If you want to log in to your app container using the `cf ssh` command, a shell such as `sh` or `bash` must be available in the container

Ports

- When configuring a Docker image for Cloud Foundry the exposed port will set the CF PORT value
 - You can use the Dockerfile EXPOSE directive to set the container and CF port
- CF supports only one port per container

Pushing a docker image

- To deploy a Docker image from a Docker Hub
 - `$ cf push APP-NAME --docker-image ACCT/REPO:TAG`
- **APP-NAME**: The name of the app being pushed
- **ACCT**: The name of the account/org where the image is stored
- **REPO**: The name of the image repository in the account/org
- **TAG**: (Optional) The tag of the image (defaults to “latest”)
- For example, the following command pushes the my-image image from Docker Hub to a Cloud Foundry app:
 - `$ cf push my-app --docker-image cloudfoundry/my-image`

Lab 13

- Pushing container images to Cloud Foundry

The End

- Many thanks for attending!

Books & Pubs

- **Domain Driven Design**
 - Eric J. Evans, Addison-Wesley Professional
- **Building Microservices**
 - Sam Newman, O'Reilly
- **Accelerate: The Science of Lean Software and DevOps**
 - Building and Scaling High Performing Technology Organizations
 - Nicole Forsgren
- **Microservices Patterns**
 - Chris Richardson
- **The New Stack**
 - <https://thenewstack.io>
- **Microservices Weekly**
 - <https://microserviceweekly.com>

