IBM MobileFirst

Using Containers to Build and Deploy Scalable APIs in the Cloud

Raymond Camden, Developer Advocate for IBM

IBM

Interactivity Tips

1. Ask A Question

2. Download a PDF copy of today's presentation

3. Social Networking Tools

About your Presenter

+ Developer Advocate for IBM

+ MobileFirst, Bluemix, Cordova/PhoneGap, Node, and web stuff in general

+ Blogging at raymondcamden.com

+ Tweeting at @raymondcamden

3

Development in the Past (for me…)

+ Focused heavily on an app server

+ App server did everything (database access, generating HTML, etc)

+ Client-side was really limited

+ Ex:

  • Client: "I need a list of cats"

  • Server: "I'll ask the database for the list."

  • Server: "I'll take that list and generate a lot of HTML to render it"

  • Server: "I'll wrap that display in some site template with dynamic junk"

  • Server: "I'll return a big pile of HTML to you"

Development in the Present (for me…)

+ The Client (desktop, mobile, TV, etc) is a lot more powerful

+ The Server needs to do a lot less

+ Ex:

  • Client: "Give me a list of cats"

  • Server: "Here is a list in raw JSON"

  • Client: "Thanks, you're awesome, I'll render it"

The App Server in 2016

+ Smaller

+ Simpler

+ An API provider

The Components…

Node.js

+ Built on Chrome's V8 JS Engine

+ JavaScript on the server

+ Very lightweight and fast

+ Completely free and open source

+ Lots and lots and lots of libraries available

Express

+ Very minimal web framework

+ On /getCats do so and so

+ Makes it easy to load static assets (pictures, CSS, etc)

+ expressjs.com

Quick Example

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
    res.send('Hello World!');
});

app.listen(3000, function () {
    console.log('Example app listening on port 3000!');
});
```

LoopBack

+ Open source framework designed for APIs

+ Rapid creation of Models

+ Simple ORM system for CRUD

+ Support for JavaScript apps (Angular)

+ Very detailed/configurable security policies

+ loopback.io

StrongLoop

+ Graphical tool for working with LoopBack

+ Deployment

+ Management

+ Monitoring

+ Debugging

+ strongloop.com

Getting Started

Installation

+ Requires npm (Node Package Manager)

+ Install Node.js (nodejs.org)

+ npm install –g strongloop

+ CLI: slc

```
[→  bin  slc -v
strongloop v6.0.0 (node v4.2.6)
├── strong-arc@1.8.8 (fb756f2)
├── strong-build@2.1.0 (47dd24a)
├── strong-deploy@3.1.2 (be6180a)
├── strong-mesh-models@8.1.0 (62e539b)
├── strong-pm@5.2.0 (4197516)
├── strong-registry@1.1.5 (f46e58f)
├── strong-start@1.3.2 (1327018)
├┐ strong-supervisor@3.3.1 (1e39220)
│└── strong-agent@2.0.2 (4ea7ee9)
├── generator-loopback@1.15.1 (826d610)
├── node-inspector@0.7.4
└── nodefly-register@0.3.3
```

Creating an Application

+ slc loopback

+ Follow the prompts

+ Define models

+ Run the application

Demo

IBM **MobileFirst**

What Next?

+ Setup a datasource (example)

+ Setup security

+ Add custom methods

  • Unique stuff (/friendlyCats vs /api/Cats?filter[where][friendly]=true)

  • Customize how data is stored ("If friendly, change name to…")

+ Build your client

Demo

Deploying

Options

+ Your own server

  • FTP (or some other fancier way) it over and run it

+ Deploy to a Bluemix

  • Cloud Foundry Application

  • Container

+ Possibly add one of the *many* services

+ Reminder – free to try!

Cloud Foundry Apps
4.375 GB/8 GB Used

CREATE APP

Containers
256 MB/2 GB
1/2 Public IPs Requested | 1 Used

START CONTAINERS

BETA **Virtual Machines**
0 B/0 B | 0/0 Public IPs

RUN VIRTUAL MACHINES

Data & Analytics

WORK WITH DATA

100+ **Services & APIs**
26/40 Used

USE SERVICES OR APIS

Compute // Start with Cloud Foundry or Docker images

Container Images

Create containers from IBM images or add your own

ibm-mobilefirst-starter
IBM

ibm-node-strong-pm
IBM

ibmliberty
IBM

ibmnode
IBM

Add your own
My Org

ibm-node-strong-pm

IBM

TAG / VERSION

latest

Copy Image URL

VIRTUAL SIZE
510.29 MB

CREATED DATE
01/04/2016

TYPE
Container Image

VIEW DOCS

TERMS

Single Container	Scalable Group

Use single container deployment for short-term processes. You can make your container accessible to the internet by assigning a public route.

Space:

My Space ▼

Container name:

MyTest

Size:

Small (1 GB Memory, 64 GB Storage) ▼

Public IP address: Public ports: ⓘ

Request and Bind Public IP ▼   22/tcp, 3001/tcp, 3002/tcp, 3003/tcp, 3004/tcp, 3

Advanced Options   ⌃

Projected Usage

Memory:

USED	QUOTA
1.500 GB	2 GB

Public IP Addresses:

USED	REQUESTED	QUOTA
2	2	2

Vulnerability Assessment

🛑 Deploy with Caution

⚠ You authorized an override to deploy this vulnerable image.

View the vulnerability report for this image

Manage your org's policies

CREATE

Then what…

+ Keep working on your app…

+ Use StrongLoop Arc to deploy

+ Repeat

+ Demo!

Wrap Up

IBM MobileFirst

Conclusion

+ LoopBack/StrongLoop

  - Built on Node.js/Express

  - Rapid API development and prototyping

  - Easy to use on the client by following REST

+ Bluemix + Containers

  - Easy to use PaaS

  - Access to numerous services

  - Containers (esp. with StrongLoop) dramatically simplify deployment

Contacting Me

+ Email: rcamden@us.ibm.com

+ Blog: www.raymondcamden.com

+ Twitter: @raymondcamden

Thank You

IBM **MobileFirst**

QUESTIONS?