



# IBM Bluemix

Hands on Workshop  
Section 1 - Overview

<Presenter name>

# Bluemix: IBM's Cloud Platform

Build, run, scale, manage, integrate & secure applications in the cloud

## Developer experience

- Rapidly deploy and scale applications in any language.
- Compose applications quickly with useful APIs and services and avoid tedious backend config.
- Realize fast time-to-value with simplicity, flexibility and clear documentation.



Built on a foundation of open technology.

## Enterprise capability

- Securely integrate with existing on-prem data and systems.
- Choose from flexible deployment models.
- Manage the full application lifecycle with DevOps.
- Develop and deploy on a platform built on a foundation of open technology.

## Bluemix service categories

- DevOps
- Big Data
- Mobile
- Watson
- Business Analytics

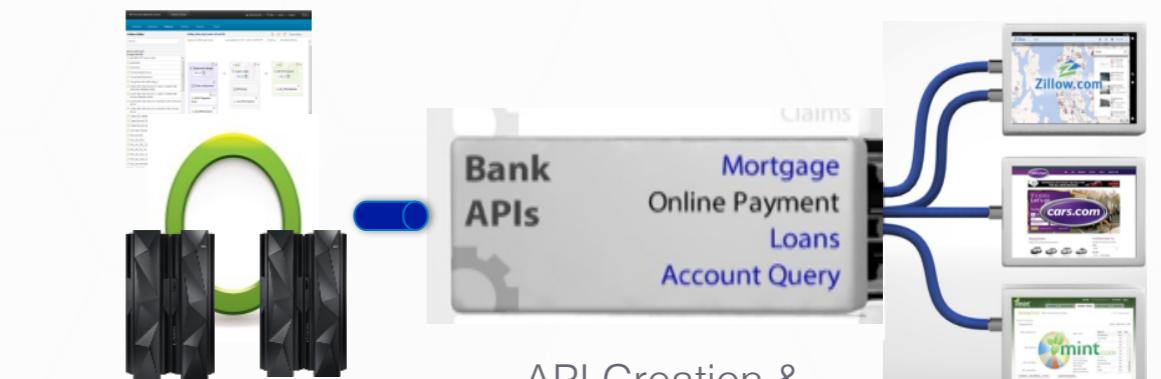
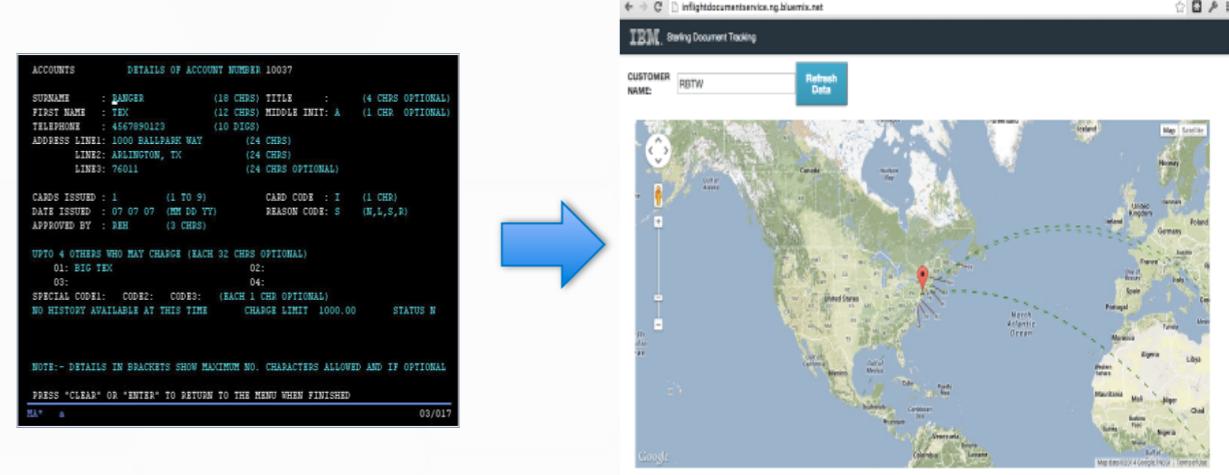
- Database
- Web and application
- Security
- Internet of Things
- Cloud Integration

- API management & Integration



# Addressing Many Types of Use Cases

- Extend existing applications
  - Add user experience such as mobile, social
  - Add new capabilities integrating other services/APIs
  - Rapid experimentation for new capabilities
- API enable applications
  - Scalable API layer on top of existing services
  - Simplify how composite service capabilities are exposed via APIs
- New applications
  - Systems of Engagement
  - Different state management models
  - 12-factor applications



Backend Systems and

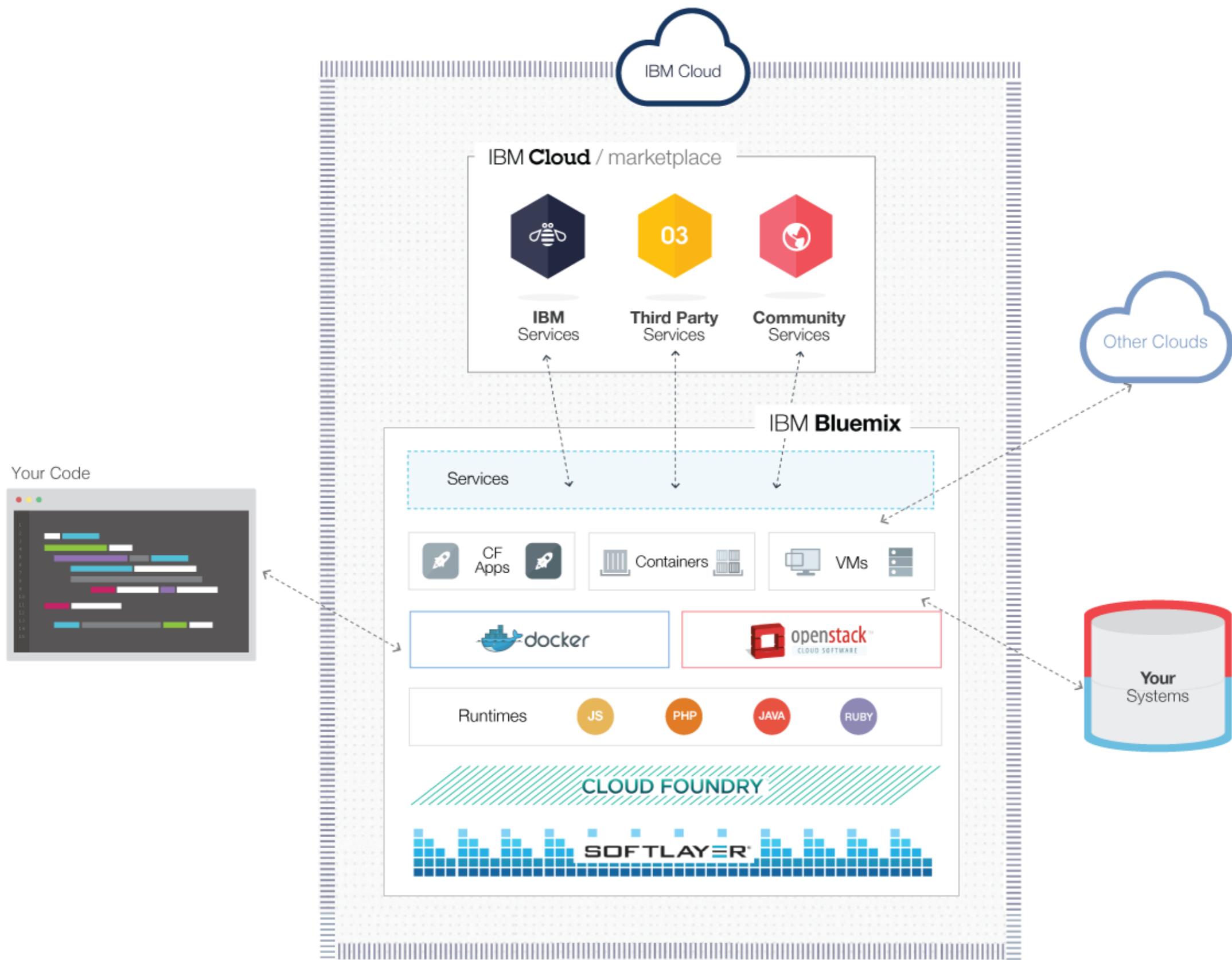
Integration

API Creation &  
Management

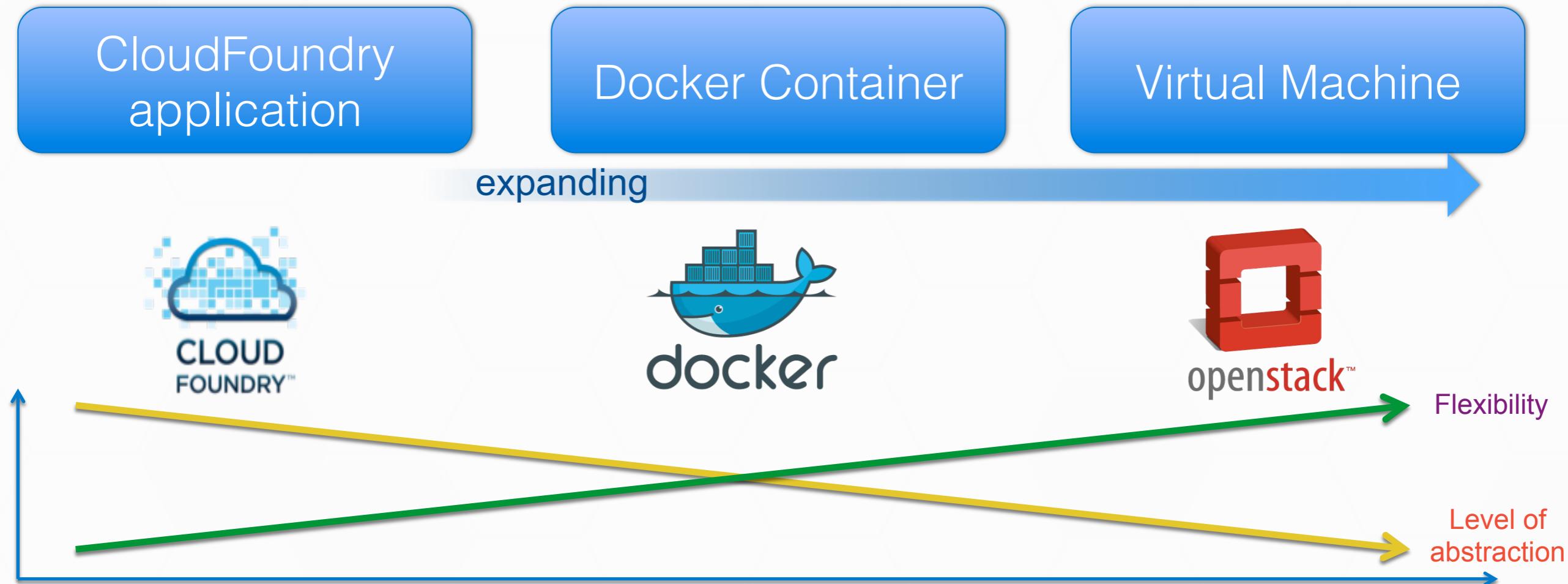
New Channels &  
Opportunities



# Bluemix structure



# Bluemix offers different compute models to “run your code”



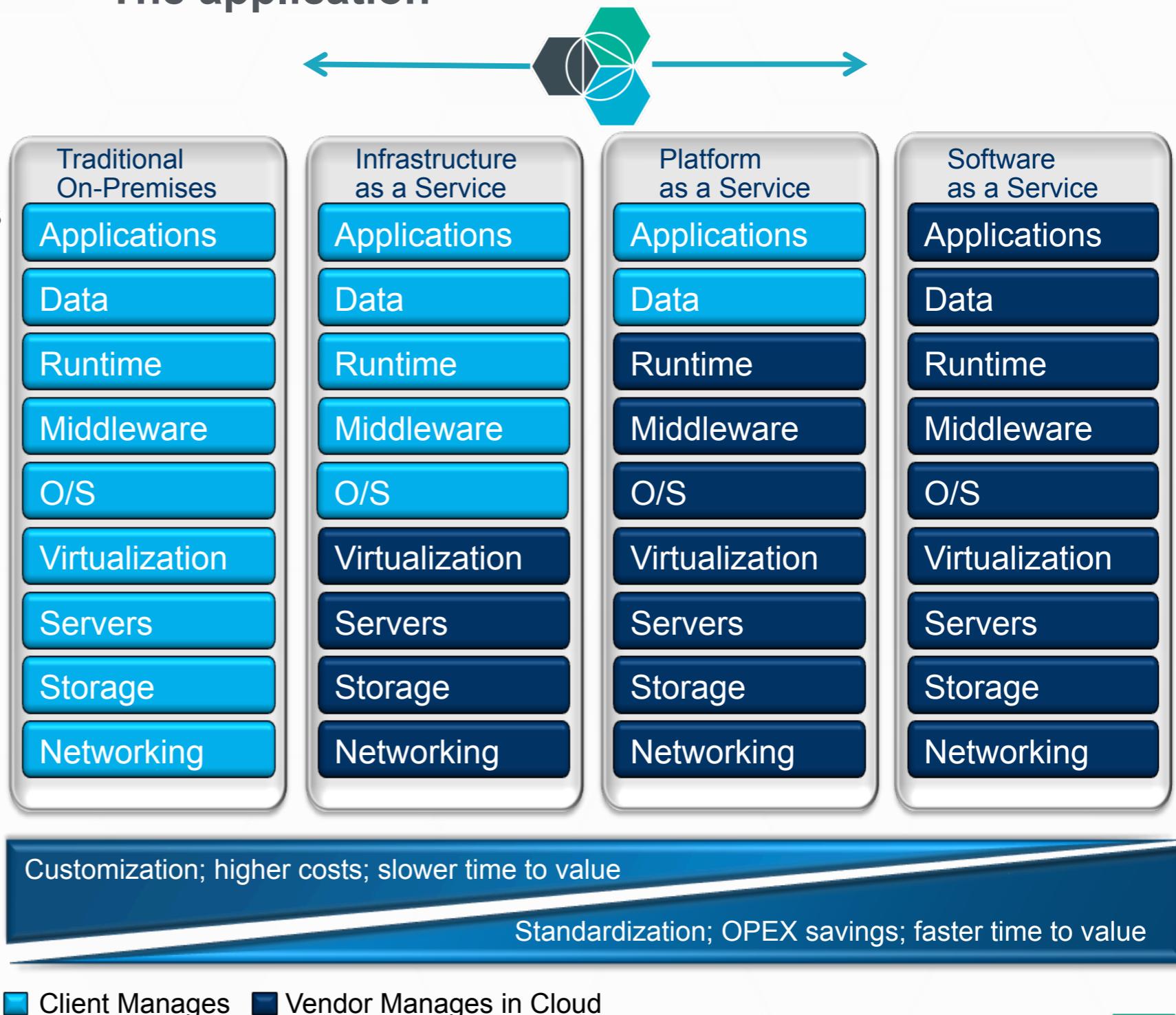
Consistent experience:

- Common service binding & consumption model
  - Common user ID & permissions model
  - Ability to hook into common routing layer



# A PaaS let's the developer focus on what's most important to them: The application

- Bluemix eliminates / dramatically simplifies various tasks:
- OS patching
- OS security hardening
- Deployment
- Load-balancing
- Scaling
- Health management



# Bluemix: Foundational Concepts

Account – Anchor point for billing

Organisations – enables team collaboration

## Spaces

- Logical grouping of apps and service instances
  - per-user permissions

Apps

Containers

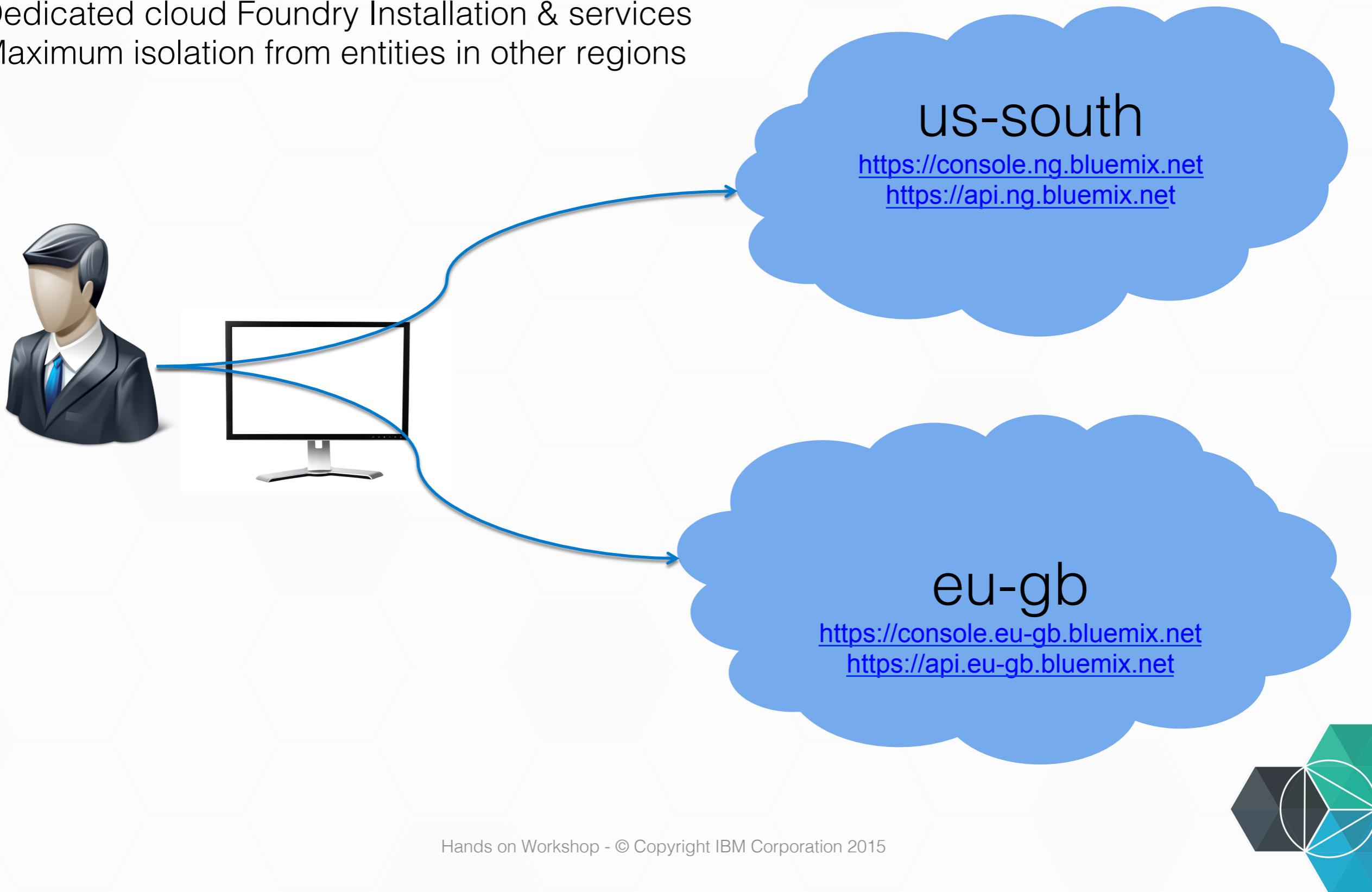
VMs

Service  
Instances

User

# So what's a Region ?

- Dedicated cloud Foundry Installation & services
- Maximum isolation from entities in other regions



# Building increasingly flexible deployment models

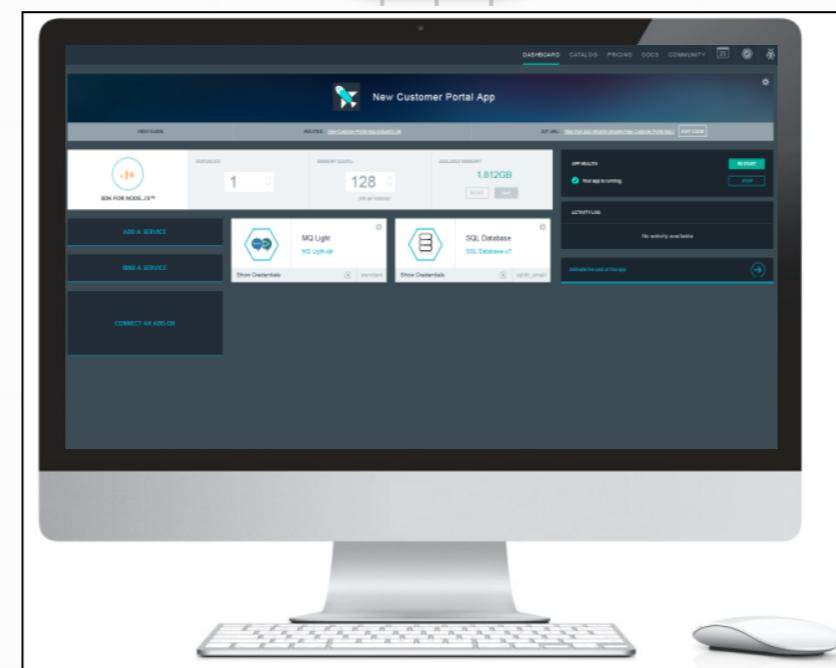
Public



Dedicated



On-premises





# IBM Bluemix

## Hands on Workshop Section 2 – Overview of Bluemix



# IBM Bluemix

## Hands on Workshop

### Section 3 – Deploying your first application

# Boilerplates

- Provide a fast way to get an application started
- Package of sample application code and services

**Starters // Choose a package of sample code and services, or start from scratch**

**Boilerplates**  
Get started with a new app, now

 Java Cache Web Starter <b>IBM</b>	 Java Cloudant Web Starter <b>IBM</b>	 Java DB Web Starter <b>IBM</b>	 Mobile Cloud <b>IBM</b>	 iOS 8 MobileFirst Services Starter <b>IBM</b>	 Node.js Cache Web Starter <b>IBM</b>
 Node.js Cloudant DB Web Starter <b>IBM</b>	 Personality Insights Java Web Starter <b>IBM</b>	 Personality Insights Node.js Web Starter <b>IBM</b>	 Node-RED Starter Community	 Ruby Sinatra Community	 Vaadin Rich Web Starter Community





## Exercise 3.a

Deploy your first application

# Accessing sample code

- Sample code for the application is available to download once a runtime or boilerplate has been deployed

The screenshot shows the IBM Bluemix dashboard for an application named "BI-MyFirstDeploy-BI". On the left sidebar, under the "Start Coding" section, there is an orange oval around the "Start Coding >" link. A red callout bubble with the text "1.Click" points to this link. In the main content area, there are three coding options: "Eclipse Tools for Bluemix", "CF Command Line Interface", and "GIT with IBM DevOps". Below these options, a section titled "Start coding with Cloud Foundry command line interface" provides instructions and links for setting up the CF Command Line Interface. A red callout bubble with the text "2.Click" points to the "Download Starter Code" button. At the bottom of the page, there is a terminal window showing the command "\$ cd directory\_name". The footer of the page reads "Hands on Workshop - © Copyright IBM Corporation 2015".

How do you want to start coding?

Eclipse Tools for Bluemix  
Develop, integrate, and push applications to Bluemix using Eclipse.

CF Command Line Interface  
Run your code locally.  
Manually push to Bluemix.

GIT with IBM DevOps  
When you commit to GIT, push to Bluemix with DevOps Services.

Start coding with Cloud Foundry command line interface

You can use the Cloud Foundry command line interface to deploy and modify applications and service instances.

**S** Setup: Before you begin, install the Cloud Foundry command line interface.

[Download CF Command Line Interface](#)

**1** Download and extract your starter code to set up your development environment.

[Download Starter Code](#)

**2** Change to your new directory.

```
$ cd directory_name
```

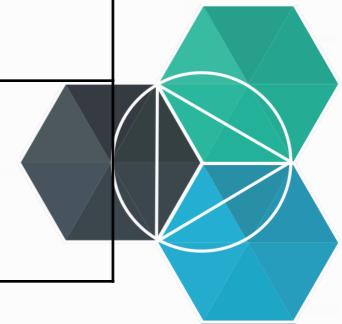
Hands on Workshop - © Copyright IBM Corporation 2015



# Command Line Interface

- Bluemix uses the CloudFoundry command line interface (CLI) – cf
- cf help – provides help page showing all the commands
- cf help <command> - provides help for specific command

<b>cf l</b>	login	<b>cf ds</b>	delete service
<b>cf t</b>	target space or organization	<b>cf bs</b>	bind service to application
<b>cf a</b>	list apps in current space	<b>cf st</b>	start app
<b>cf app</b>	display status for specific app in current space	<b>cf sp</b>	stop app
<b>cf p</b>	push (deploy or update) app	<b>cf d</b>	delete app
<b>cf s</b>	show service info	<b>cf scale</b>	scale app
<b>cf cs</b>	create service	<b>cf logs</b>	tail or show logs for app



# Sample cf commands

- To login to Bluemix:
  - cf l -a <https://api.ng.bluemix.net> -u <email> -p <password> -o <email> -s dev
  - cf l -a <https://api.eu-gb.bluemix.net> -u <email> -p <password> -o <email> -s dev
    - This will login to Bluemix, set the organisation to the user's own organisation and the space to dev
- To check what space you are logged onto or to change the space:
  - cf t
  - cf t -s test
    - The first option will print the current target organisation and space
    - The second option will switch to the test space
- To check what spaces exist in an organisation:
  - cf spaces
    - Displays the spaces available in the current organisation
  - cf space dev
    - Displays information about the dev space in the current organisation



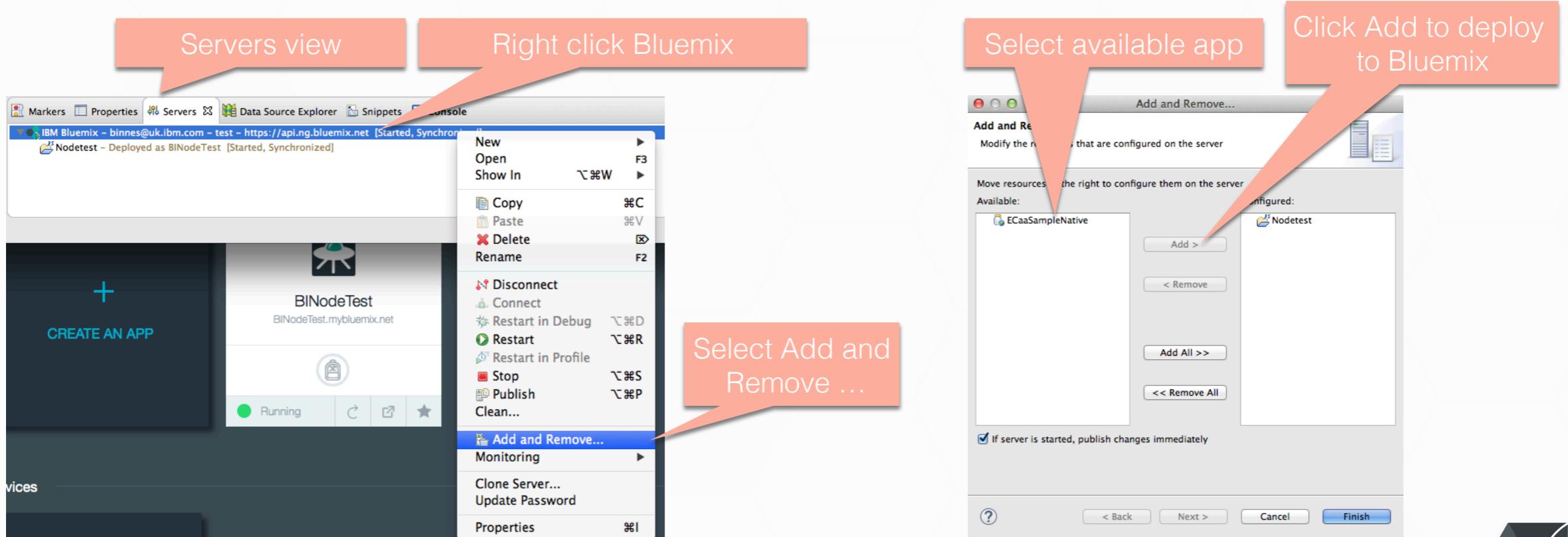


## Exercise 3.b

Deploy then update an application using the CLI

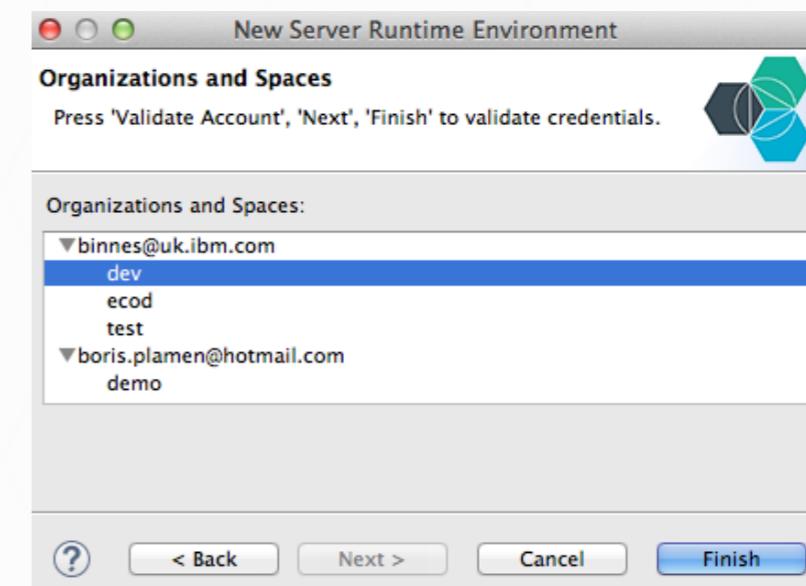
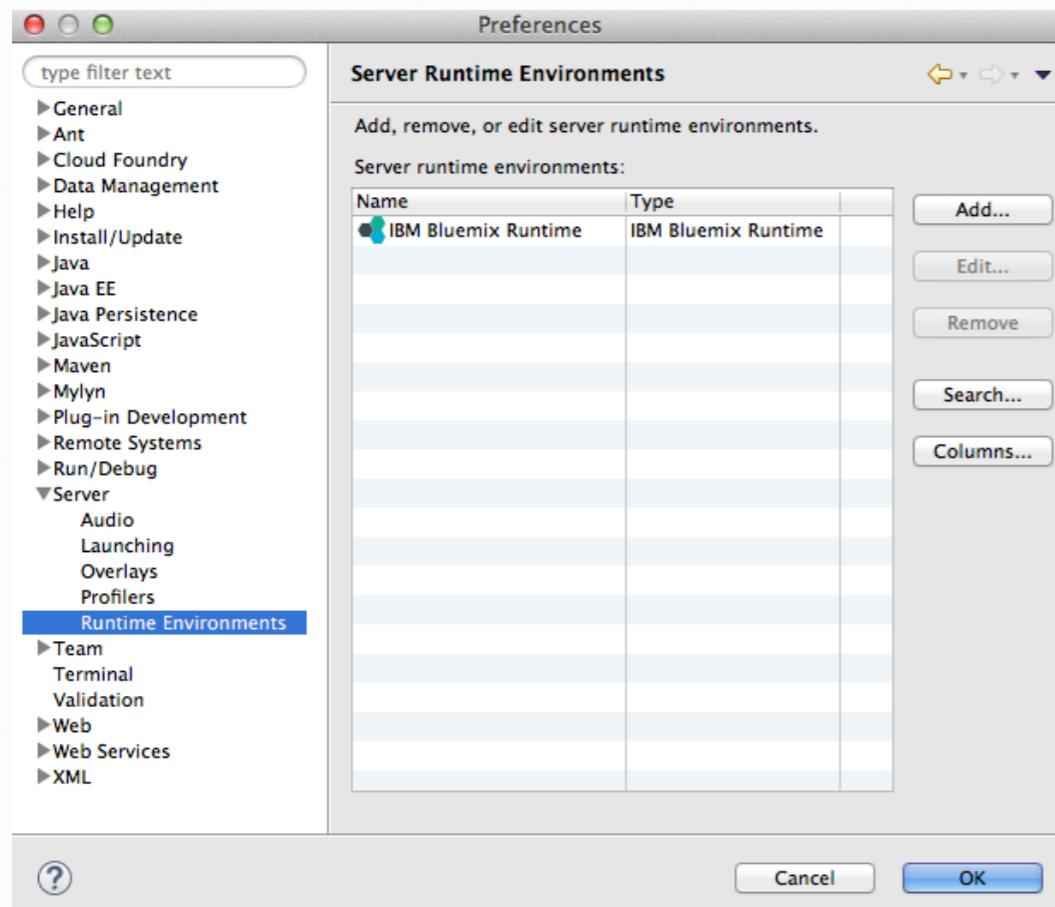
# Eclipse plugin for Bluemix

- Available in the Eclipse marketplace
- Enables developers to develop in Eclipse then deploy to Bluemix.
  - Java and JavaScript supported



# Need to define your Bluemix server connection in Eclipse

- In Eclipse preferences select ‘Server’ then ‘Runtime Environments’
- Need to add a server definition for each Bluemix space you want to deploy to

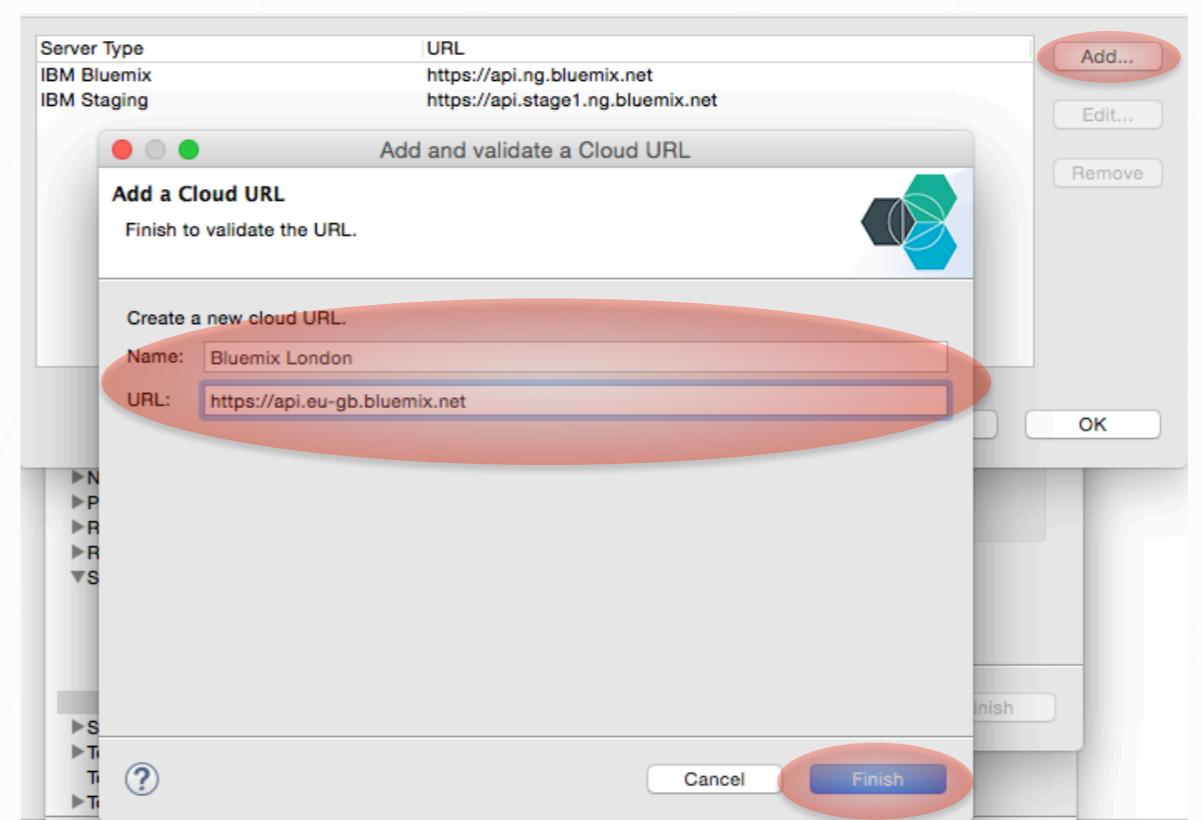
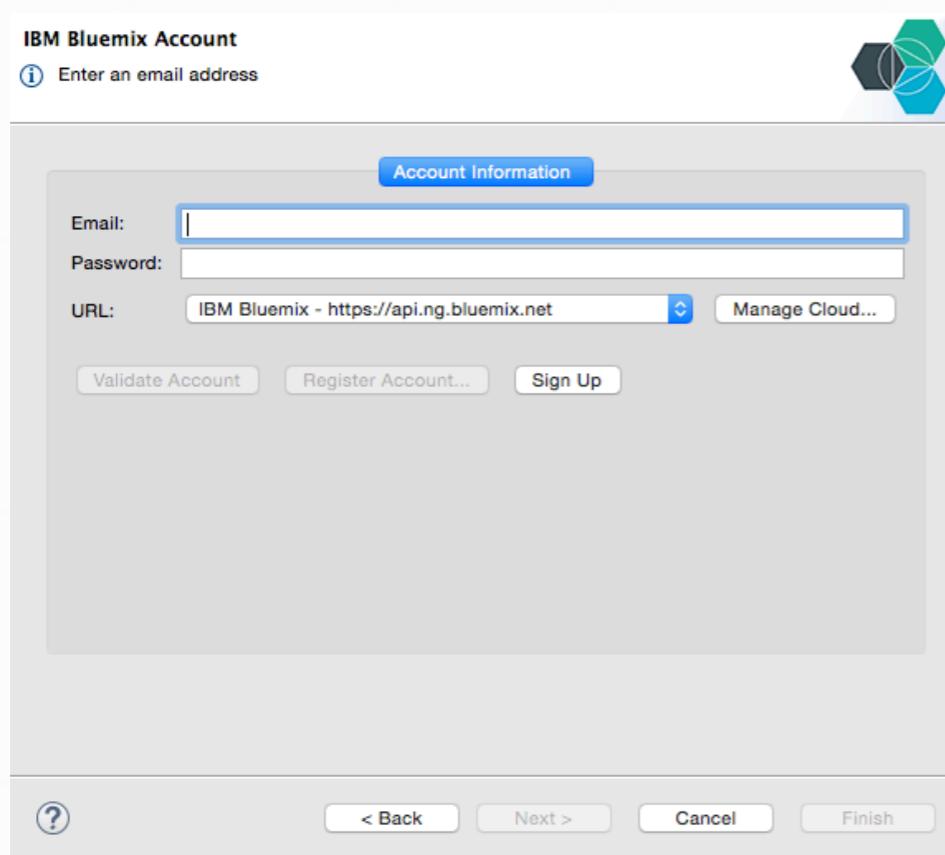


# Adding additional Bluemix Regions to Eclipse

1. Select 'Manage Cloud ...' from the Add Bluemix Account dialog box
2. Select 'Add' to add an addition Bluemix cloud
3. Enter the details of the Bluemix region by providing the name and endpoint URL
4. Select 'Finish'

You can find details of region endpoint URLs here:

[https://www.ng.bluemix.net/docs/#overview/overview.html#ov\\_intro\\_reg](https://www.ng.bluemix.net/docs/#overview/overview.html#ov_intro_reg)



# Environment Variables

- Bluemix sets environment variables when deploying an application to inform the application about the environment it is running in.
  - Application may need to use this information to set configuration – such as the hostname and port to listen for incoming requests.
  - VCAP\_APP\_HOST – specifies the hostname on which your application should listen
  - VCAP\_APP\_PORT – specifies the port number on which your application should listen
  - You can also define custom environment variables to provide configuration to your application



# Manifest file

- Allows you to specify the parameters for an application deployment
- manifest.yml file is used at deploy time if found in directory application is being pushed from

```
---  
applications:  
- name: bluemix-todo-node  
  host: bluemix-todo-node-${random-word}  
  command: node app.js  
  memory: 128M  
services:  
- todo-db
```

```
---  
applications:  
- name: Myphpmyadmin  
  memory: 128M  
  instances: 1  
  host: Myphpmyadmin  
  path: .  
  buildpack: https://github.com/dmikusa-pivotal/cf-php-build-pack.git  
services:  
- mysql_BlueMixLab
```

<http://docs.cloudfoundry.org/devguide/deploy-apps/manifest.html>



# .cignore

- When you push an application all content of the current directory and all sub directories are pushed to the server – this is not the behaviour you always want
- .cignore allows control of what is sent to the server by listing the files and directories you do not want to send to the server
- Sample .cignore file:  
.git  
node\_modules  
tmp  
lib-src





## Exercise 3.c

Working with Eclipse and  
Bluemix

# Summary

- Bluemix provides a number of options for developing applications giving you the ability to integrate Bluemix into the developer tools of your choice
- Bluemix provides starter packs to get you up and running quickly
- Start applications from scratch





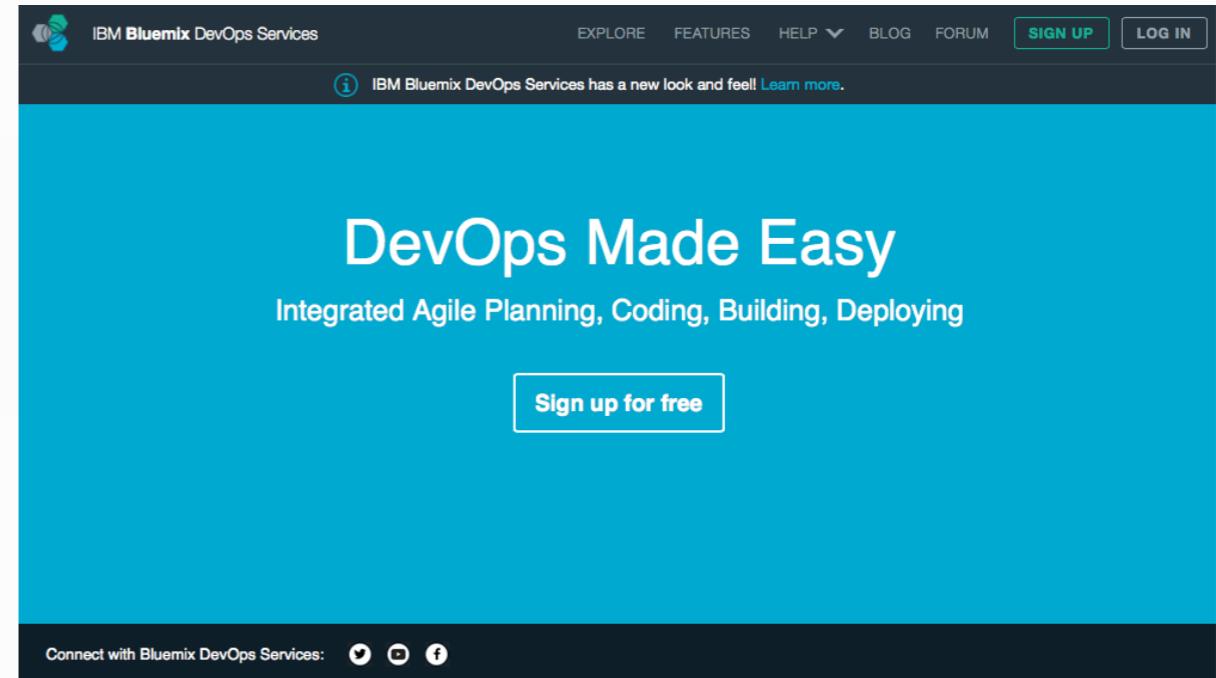
# IBM Bluemix

Hands on Workshop  
Section 4 – DevOps and Bluemix

# IBM Bluemix™ DevOps Services

*An Open, Integrated Rapid Development Experience that Scales*

- **A Premier DevOps Service for IBM Cloud Platform**
  - Promotes incremental frictionless adoption of DevOps Services for Bluemix
- **An Integrated Developer Experience**
  - End-to-end DevOps Solution in the cloud for developing applications.
  - Integrated task tracking, agile planning, source control with auto deploy
  - Complementary mobile quality and application performance monitoring
  - Use your favourite tools or work from the Web IDE
  - Free public and fee-based private projects (free for now now)
- **Scalable, secure, enterprise-ready**
  - Runs on SoftLayer Infrastructure



We bring the tools. You bring the code.



# DevOps Services available today



## Easy Access

Get started for free. With Git hosting and the built-in Web IDE, it's zero to code in seconds.



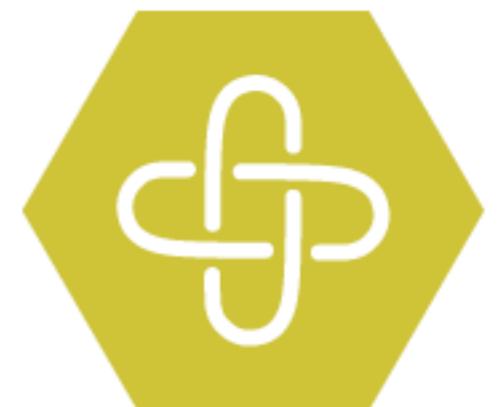
## Code Now

Use the built-in Web IDE, Eclipse, Visual Studio, or your tool of choice.



## Build & Deploy

Automatically build and deploy your application to IBM's cloud platform, Bluemix.



## Team Collaboration

Share your work and collaborate through expert tools for Agile Development.





## Exercise 4.a

Bluemix integration with DevOps  
Services

# Get productive with Bluemix™ DevOps Services in minutes

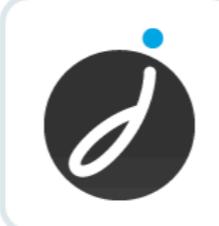
- Register at [www.jazzhub.com](http://www.jazzhub.com)
- Answer a few questions
  - Which SCM (Git, GitHub, Jazz SCM)?
  - Do you want your project to be public or private?
  - Do you want to practice agile software development?
  - Do you want to deploy on IBM Bluemix?
- Start coding

Create a project

binnes |

URL: <https://hub.jazz.net/project/binnes/>

Choose where your code will be (required)

 Use Jazz SCM     Create a Git repository     Connect to an external GitHub repository  
or

Private  
By default, projects are public. Select this to make it private so that you can share with project members only. [Learn more](#)

Add features for Scrum development (This option can only be added at project creation time.)  
Select this if you're familiar with Scrum and plan to deliver software on regular sprints. [i](#)

Make this a Bluemix Project  
Select this if you want to deploy your application to the IBM Bluemix cloud platform. [Find out how](#) [i](#)

[CANCEL](#) [CREATE](#)



# Collaborate seamlessly on public or private projects

- **Easily invite team members**
- **Access from anywhere**
- **Built for collaboration from the ground up**
- **Choose who sees your project, and how you engage with broader communities**

The screenshot shows the IBM Bluemix DevOps Services interface. At the top, there's a navigation bar with links for DASHBOARD, MY PROJECTS, EXPLORE, HELP, BLOG, and FORUM. Below that is a project header for "BI-MyFirstDeploy" with tabs for EDIT CODE, TRACK & PLAN, BUILD & DEPLOY, and a settings gear icon. The main area has sections for OVERVIEW, MEMBERS (1), and GIT LOG. A modal window titled "Invitations for binnes | BI-MyFirstDeploy" is open, containing a text input field for email addresses and a green "INVITE" button. To the right of the modal, there's a sidebar with "Members (1 of 1)" and a link to "About this project". A red circle highlights the "INVITE" button, and another red circle highlights the "Members (1 of 1)" section.

Invite other people to your project

Enter comma separated list of email addresses



# Agile development in the Cloud is easy with Bluemix™ DevOps Services for Bluemix™

- Built-in agile process support
- Work items to track and plan project activities
- Agile tools for the product backlog, releases, and sprints.
- Dashboard charts for project status

General + Add Widget

Select Catalog: Change and Configuration Management (/ccm10)

Search...

Current Iteration: 3.0 M7

Plans

Process Description

Performance Health Check

Plan View

Open Work Items by Type

Plan Statistics

Add External Widgets

Add OpenSocial Gadget

General

All

Build

Feeds

General

Help

Plans

Project/Team

Reports

Source Control

Work Items

Unassigned

Markus Kent

Jason Mitchell

Bill Cassavelli

assertEquals throws NPE while comparing null etc

assertArrayEquals misses differences

testCount hard-coded to 1 for childless Descriptors

Based on the assertThat syntax we should provide

Improve documentation for 4.4

assertEquals array comparison doesn't handle nu

javacnc updates for @ignore in 4.3

Tests on protected methods fail

Effective Estimate Priority

3 weeks High

2 hours High

— Low

2 days Unass

— Unass

1 day Unass

1 hour Unass

— Unass

37 - 42 of 65

The screenshot shows the 'Change and Configuration Management' catalog page. It features several widgets: a performance health check with latency and database latency charts; a plan view showing tasks for Markus Kent, Jason Mitchell, and Bill Cassavelli; and a process description detailing Scrum methodology with core practices like backlog, iterative development, and sprint reviews. A sidebar on the left lists categories like All, Build, Feeds, General, Help, Plans, Project/Team, Reports, Source Control, and Work Items, along with options to add external widgets or an OpenSocial gadget.



# Choose how you code with Bluemix™ DevOps Services

- **Browser-based Integrated Development Environment**
- **Full support for local development with Eclipse or Visual Studio**
- **Built-in support for Jazz Source Control**
- **Hosted Git repository**
- **Got GitHub? No problem**

The screenshot shows the IBM Bluemix DevOps Services interface. At the top, there's a navigation bar with links for DASHBOARD, MY PROJECTS, EXPLORE, HELP, BLOG, and FORUM. Below the navigation is a project overview for 'binnes | SentimentAnalysisApp'. On the left, a file tree shows files like .git, public, and app.js (which is selected). The main area is a code editor with the following content:

```
/*jshint node:true*/
var port = (process.env.VCAP_APP_PORT || 3000);
var express = require("express");
var sentiment = require('sentiment');
var twitter = require('ntwitter');

var app = express();
// Configure the app web container
app.configure(function() {
  app.use(express.static(__dirname + '/public'));
});

// Sample keys for demo and article - you must get your own keys if you clone this application!
// Create your own app at: https://dev.twitter.com/apps
var tweeter = new twitter({
  consumer_key: 'VdidGOWCQLMKWP2e6orPtFgYq',
  consumer_secret: 'Mjdz2ICdhzvh0obu0l0XNsX14fs7rvXIp21VJSN0QpTQk5e0jZ7',
  access_token_key: '1950798056-cDW1QdUE0bPas8PDONMU841QbwIxUK0AyC6h4YjnW',
  access_token_secret: 'jxFHWA1AfuFyraKLH9CuKbsXnE0u1AGzQAh10p1Q03rt'
});

app.get('/twitterCheck', function (req, res) {
  tweeter.verifyCredentials(function (error, data) {
    res.send("Hello, " + data.name + ". I am in your twitters.");
  });
});

var tweetCount = 0;
var tweetTotalSentiment = 0;
var monitoringPhrase;
```

Below the code editor, there's a section titled "Choose where your code will be" with three options:

- Use Jazz source control hosted at JazzHub (represented by a stylized 'j' icon)
- Create a Git repository hosted at JazzHub (represented by a red diamond icon with a white branch)
- Connect to an existing GitHub repository (represented by a GitHub logo with a checkmark)

A text input field at the bottom allows users to "Copy and paste URL (<https://github.com/username/project-name.git>)".



# Live Sync for Node.js

- Enables Node.js code changes to be pushed live without the need to redeploy the application
- Available in IBM Bluemix DevOps Services as the Live Edit feature
- Available on MS Windows and OS X platforms from the command line using the Live Sync CLI

The screenshot shows the IBM Bluemix Developer Console interface. At the top, there's a toolbar with icons for File, Edit, View, Tools, and a dropdown for the application 'BI-TestApp-4 (running: live edit)'. A 'Live Edit' toggle switch is turned on. Below the toolbar, the application name 'BI-TestApp-4' and its route 'Route: BI-TestApp-4.eu-gb.mybluemix.net' are displayed. A message says 'Your app is running.' with buttons for 'SUSPEND' and 'RESTART'. Two buttons at the bottom right are 'Open Shell' and 'Open Debugger'. The main area is a code editor with tabs for 'app.js' and 'index.js'. The 'app.js' tab is active, showing the following code:

```
1 (function (exports, require, module, __file) {
2   /*
3    * GET home page.
4    */
5   exports.index = function(req, res){
6     res.render('index');
7   };
8 });
9 
```

To the right of the code editor is a 'Call Stack' panel with the following entries:

- No Watch Expressions
- Call Stack
- exports.index index.js:8
- callbacks index.js:161
- param index.js:135
- pass index.js:142
- Router\_dispatch index.js:170
- router index.js:33
- next proto.js:190
- methodOverride.js:37
- methodOverride
- next proto.js:190
- multipart multipart.js:64
- (anonymous function) bodyParser.js:57
- urlencoded urlencoded.js:51
- (anonymous function) bodyParser.js:55
- json json.js:53
- bodyParser bodyParser.js:53
- next proto.js:190
- logger logger.js:156
- next proto.js:190
- favicon favicon.js:77
- next proto.js:190

At the bottom left, it says 'Hands on Workshop - © Copyright IBM Corporation 2015'. On the far right, there's a decorative graphic of overlapping geometric shapes.



## Exercise 4.b

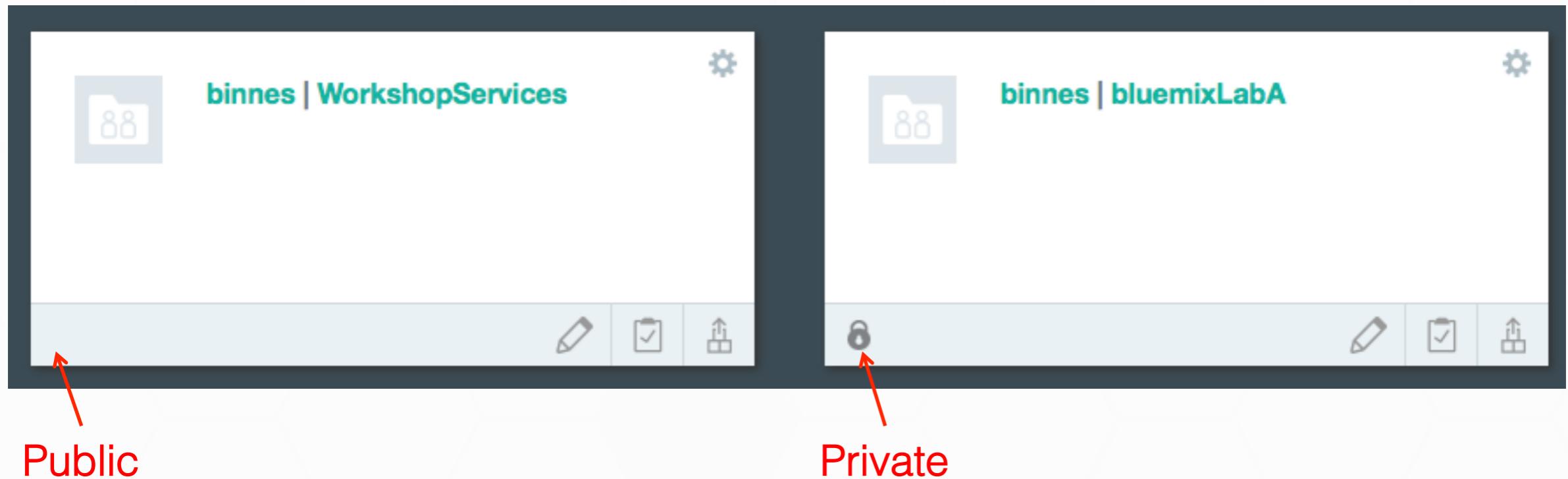
Working in DevOps Services

**Leverage the power of social coding. With public projects, it's easy to learn and share work with a broader audience.**

With public projects, it's easy to learn and share work with a broader audience.

**Private**

By default, projects are public. Select this to make it private so that you can share with project members only. [Learn more](#)



# Summary

- IBM Bluemix DevOps Services:
  - Provides an online set of tooling to plan, manage, develop, and deploy your application
  - Can use the build in code repository or choose to use Git Hub
  - Allows developers to collaborate on a project
  - Allows developers to work online or locally on the same project





# IBM Bluemix

Hands on Workshop  
Section 5 – Maximizing the value  
of Bluemix

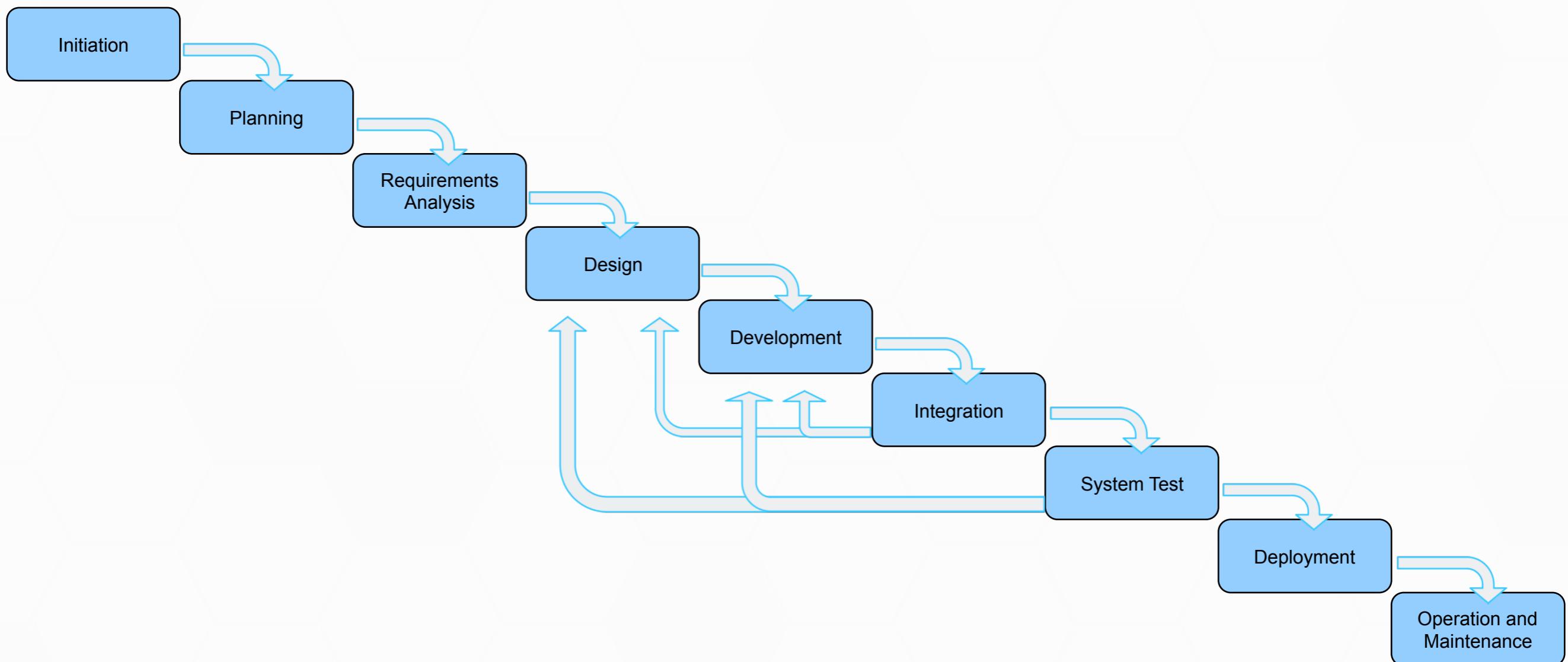
# Business challenge & developer expectations

## Client Business Challenge: Developers' expectations:

- Time to market for new applications is too long
- Speed and innovation are needed to capture new business opportunities
- Remove blockage from IT deployment
- Competitive threat from new "born on the web" companies
- The client is looking to enter into the API economy. Need environment to share or sell software assets they build/own
- Reduce operational cost and limit capital investments as well as remove the need to manage and procure assets and services

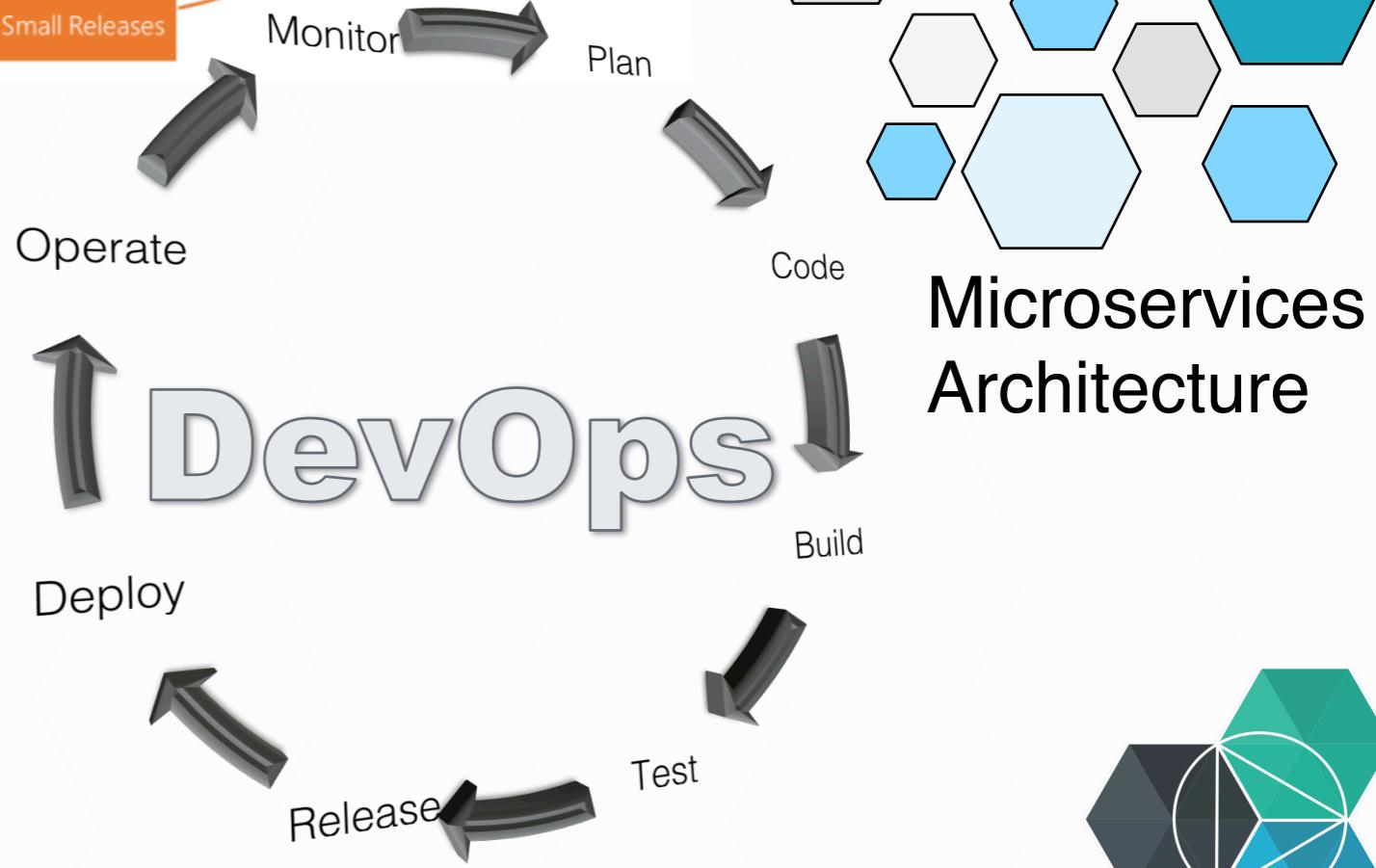


# Bluemix can fit into a software development process, but will it deliver the promised value?

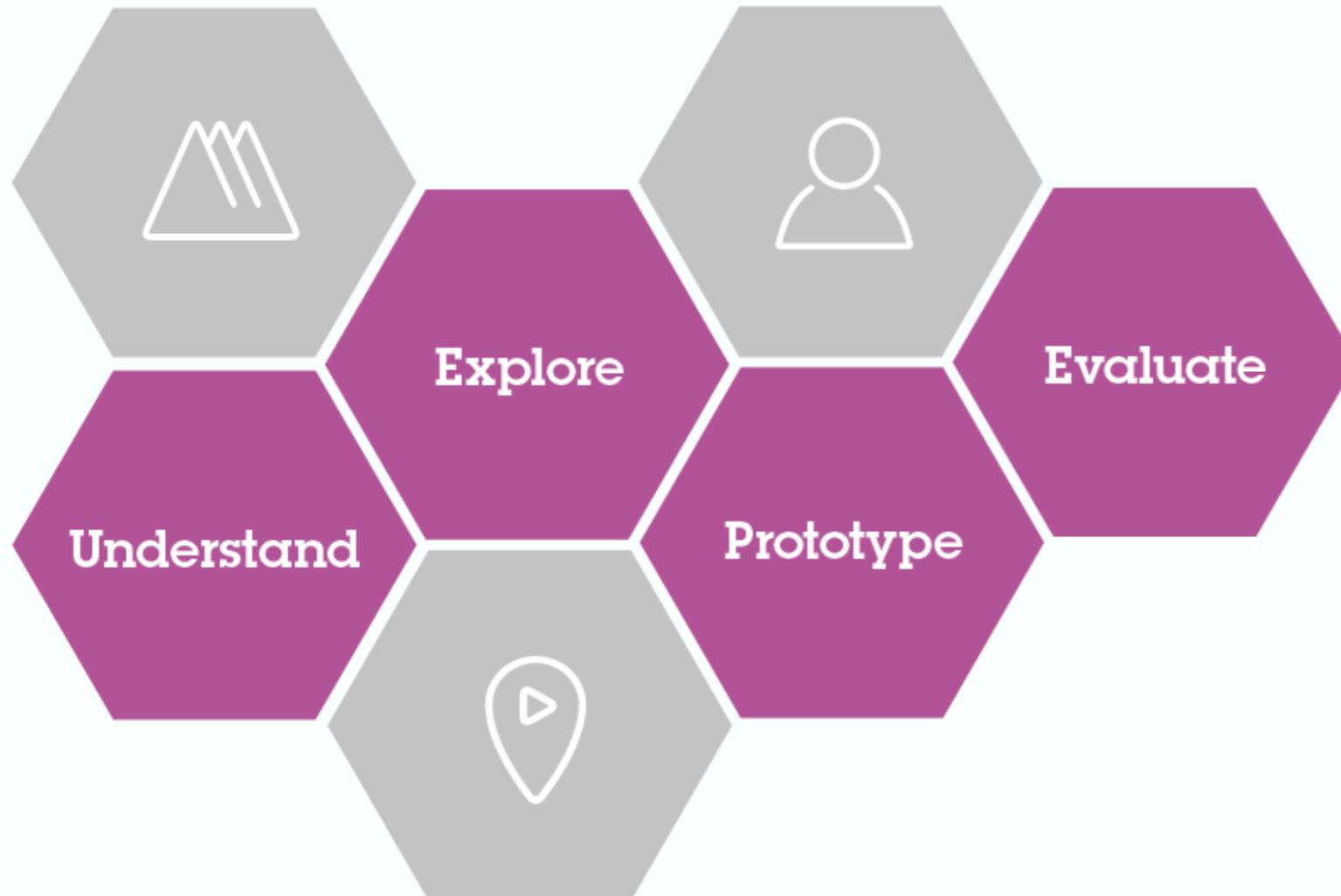


# What does it take to deliver the promised value?

- User-centered design
- Agile methodology
- DevOps processes and tooling
- Architect for cloud



# User-centred design



Hills: who, what, WOW

Sponsored users: design for real users, not for imagined needs

Playbacks: collaborate to tell a great story

Design thinking methods help you envision your user experience.



# Agile methodology

From Wikipedia:

## Agile principles

The Agile Manifesto is based on 12 principles:

- Customer satisfaction by rapid delivery of useful software
- Welcome changing requirements, even late in development
- Working software is delivered frequently
- Close, daily cooperation between business people and developers
- Projects are built around motivated individuals, who should be trusted
- Face-to-face conversation is the best form of communication (co-location)
- Working software is the principal measure of progress
- Sustainable development, able to maintain a constant pace
- Continuous attention to technical excellence and good design
- Simplicity—the art of maximizing the amount of work not done—is essential
- Self-organizing teams
- Regular adaptation to changing circumstance

## Agile practices

Agile development is supported by several concrete practices, covering areas such as requirements, design, modeling, coding, testing, project management, process, and quality. Some notable agile practices include:

- Acceptance test-driven development (ATDD)
- Agile modeling
- Backlogs (product and sprint)
- Behavior-driven development (BDD)
- Cross-functional team
- Continuous integration (CI)
- Domain-driven design (DDD)
- Information radiators (scrum board, task board, visual management board, burndown chart)
- Iterative and incremental development (IID)
- Pair programming
- Planning poker
- Refactoring
- Scrum events (sprint planning, daily scrum, sprint review, and retrospective)
- Test-driven development (TDD)
- Agile testing
- Timeboxing
- Use case
- User story
- Story-driven modeling
- Retrospective
- Velocity tracking



# DevOps

From Wikipedia:

## Goals

The specific goals of a DevOps approach span the entire delivery pipeline. They include improved deployment frequency, which can lead to faster time to market, lower failure rate of new releases, shortened lead time between fixes, and faster mean time to recovery in the event of a new release crashing or otherwise disabling the current system.

Simple processes become increasingly programmable and dynamic, using a DevOps approach, which aims to maximize the predictability, efficiency, security, and maintainability of operational processes. Very often, automation supports this objective.

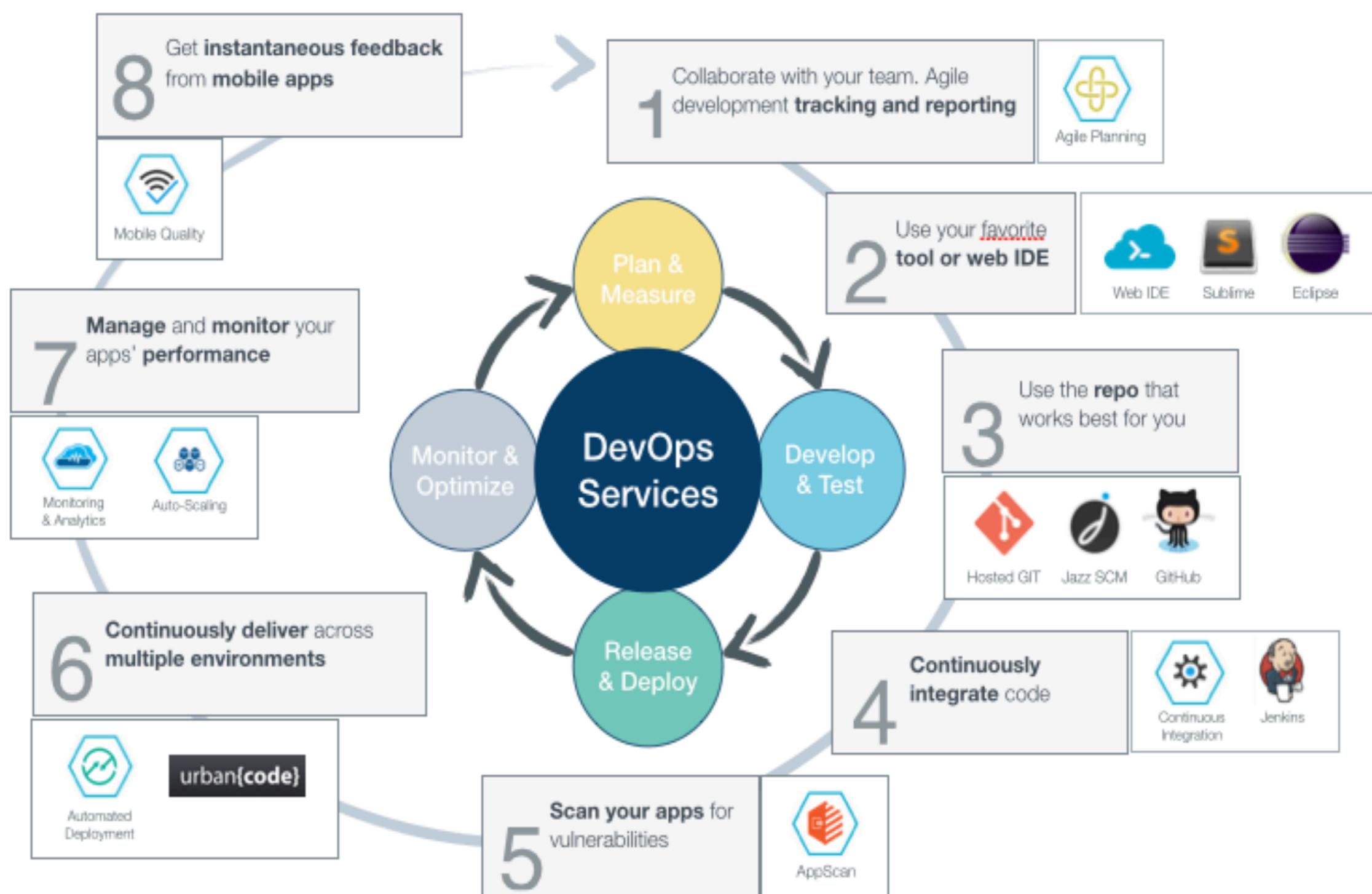
## Adoption

The adoption of DevOps is being driven by factors such as:

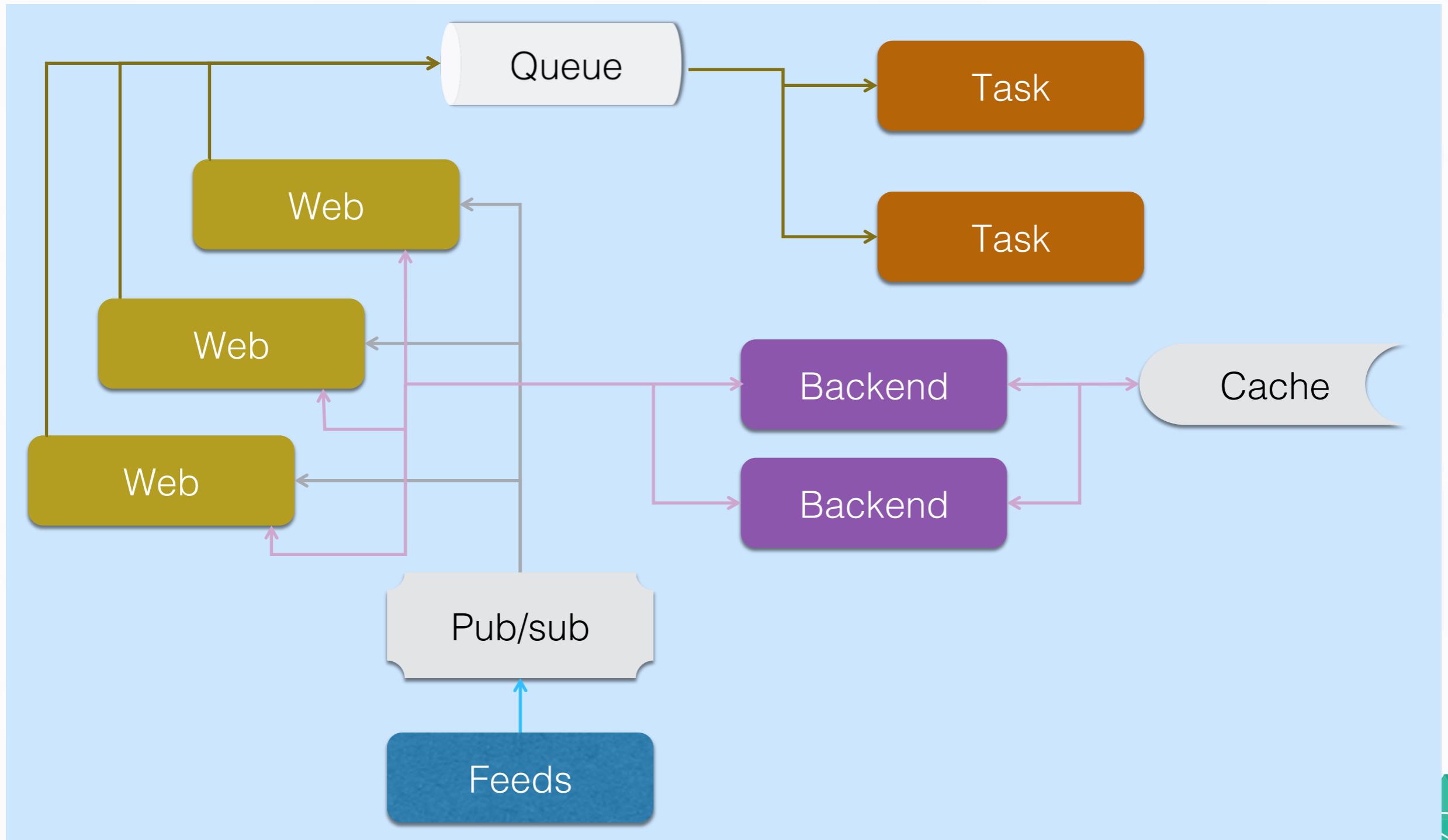
- Use of agile and other development processes and methodologies
- Demand for an increased rate of production releases from application and business unit stakeholders
- Wide availability of virtualized and cloud infrastructure from internal and external providers
- Increased usage of data center automation and configuration management tools



# Bluemix unified DevOps experience (roadmap)



# Application architecture



# The Twelve-Factor App

(<http://12factor.net>)

In the modern era, software is commonly delivered as a service: called *web apps*, or *software-as-a-service*.

The twelve-factor app is a methodology for building software-as-a-service apps that:

- Use declarative formats for setup automation, to minimize time and cost for new developers joining the project;
- Have a clean contract with the underlying operating system, offering maximum portability between execution environments;
- Are suitable for deployment on modern cloud platforms, obviating the need for servers and systems administration;
- Minimize divergence between development and production, enabling continuous deployment for maximum agility;
- And can scale up without significant changes to tooling, architecture, or development practices.

The twelve-factor methodology can be applied to apps written in any programming language, and which use any combination of backing services (database, queue, memory cache, etc).

## [I. Codebase](#)

One codebase tracked in revision control, many deploys

## [II. Dependencies](#)

Explicitly declare and isolate dependencies

## [III. Config](#)

Store config in the environment

## [IV. Backing Services](#)

Treat backing services as attached resources

## [V. Build, release, run](#)

Strictly separate build and run stages

## [VI. Processes](#)

Execute the app as one or more stateless processes

## [VII. Port binding](#)

Export services via port binding

## [VIII. Concurrency](#)

Scale out via the process model

## [IX. Disposability](#)

Maximize robustness with fast startup and graceful shutdown

## [X. Dev/prod parity](#)

Keep development, staging, and production as similar as possible

## [XI. Logs](#)

Treat logs as event streams

## [XII. Admin processes](#)

Run admin/management tasks as one-off processes





## Exercise 5.a

Creating a new project and code repository

# Best tools for the job

- Importance of having a good developer setup
- Lint tooling (static analysis is useful for code quality)
- Syntax highlighting, autocomplete, etc.
- Quality Assurance needs to be part of the development process
  - Continuous testing during development
  - Test framework – Mocha, Chai, Sinon for JavaScript
  - Static code analysis to help spot errors





## Exercise 5.b

Setting up for static analysis and  
test driven development

# Test driven development

- No code is written unless it is to pass a test
- When one function calls another function try to ensure that unit tests are not duplicating tests done by the tests for the called function
  - Test frameworks provide ‘spy’ functionality to verify functions are called correctly rather than testing what the called function returns
- Where ever possible try to ensure tests cover problematic areas of the implementation language
  - E.g. JavaScript string vs number comparison  
“5” > “11” but  $5 < 11$  type coercion sometimes delivers the wrong comparison





## Exercise 5.c

Test Driven development

# DevOps pipeline in Bluemix

- IBM Bluemix DevOps Services can be configured to automatically run tests and if the tests pass deploy code to Bluemix

The screenshot shows the IBM Bluemix DevOps Pipeline interface with three stages: Build, Test, and Deploy.

- Build Stage:** STAGE PASSED. Last input was a commit by binnes example commit for demo 5 days ago. A successful Build job ran 5 days ago. The last execution result is Build 16.
- Test Stage:** STAGE FAILED. Last input was Build 16. A failed Test job ran 5 days ago. The last execution result is No results.
- Deploy Stage:** STAGE PASSED. Last input was Stage: Build / Job: Build. A successful Deploy job ran 6 days ago. The last execution result is fizzbuzz-w3 (fizzbuzz-w3.eu-gb.mybluemix.net) with a green status indicator, and a successful Build 15 job.





## Exercise 5.d

Adding a the REST API,  
deploying to Bluemix and setting  
up the DevOps pipeline

# Summary

- A modern cloud platform alone does not deliver the agility and speed businesses require today
- A modern approach to application creation which delivers:
  - Innovation
  - Agility
  - Quality
  - Best user experiencerequires an appropriate methodology, set of processes, tooling and architecture to be able to deliver.
- Many companies are adopting a bimodal approach to manage existing applications and enable innovative new application to be created



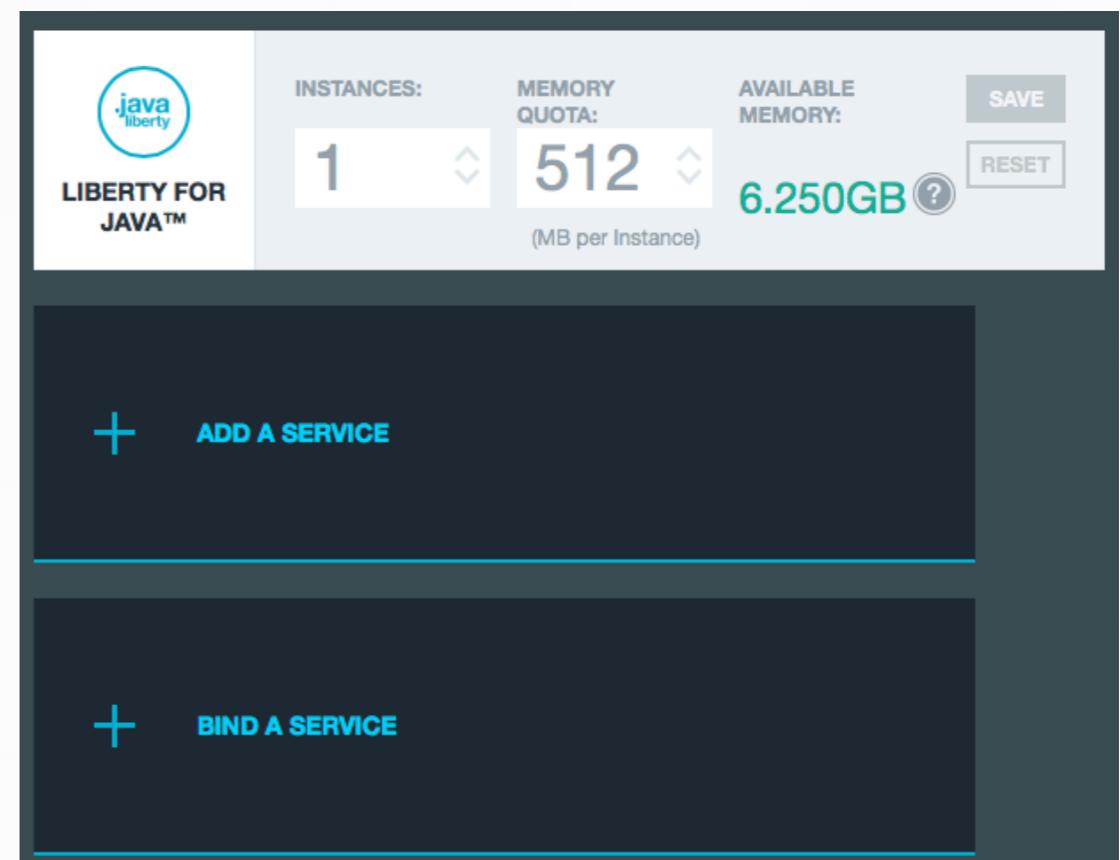


# IBM Bluemix

## Hands on Workshop Section 6 - Services

# Using services in Bluemix

- Bluemix provides services that can be used by applications without requiring you to manage the setup and operation of those services
- Available services are listed in the catalog in the Web UI and can also be obtained using cf marketplace.
- You need to bind a service to your application to get use of the service using Web UI or cf bs.



# Accessing a service

- When you bind a service to your application Bluemix will add details about the service to an environment variable `VCAP_SERVICES`.
- `VCAP_SERVICES` can be inspected in the web UI
- Applications should parse `VCAP_SERVICES`.

The screenshot shows the Bluemix web interface. On the left, there's a sidebar with links like 'Back to Dashboard', 'Overview', 'Liberty for Java™', 'Files and Logs', 'Environment Variables' (which is highlighted with a red oval and a 'Click' callout), 'Start Coding', and 'SERVICES'. Under 'SERVICES', there's a 'Cloudant NoSQL DB' entry. The main panel is titled 'Environment Variables' and shows a JSON representation of the `VCAP_SERVICES` environment variable. The JSON includes a 'cloudantNoSQLDB' entry with a single service definition. The 'VCAP\_SERVICES' tab is selected. At the bottom right of the main panel is an 'EXPORT' button. The footer of the page says 'Hands on Workshop - © Copyright IBM Corporation 2015'.

```
{ "cloudantNoSQLDB": [ { "name": "BI-MyFirstDeploy-BI-cloudantNoSQLDB", "label": "cloudantNoSQLDB", "plan": "Shared", "credentials": { "username": "22619f6f-dae8-4791-b7a9-3ef43e8eb788-bluemix", "password": "a715950065d0f25d3c7cf0ea9156cc395d4ba89bbc0bff5a994e4e40791e252a", "host": "22619f6f-dae8-4791-b7a9-3ef43e8eb788-bluemix.cloudant.com", "port": 443, "url": "https://22619f6f-dae8-4791-b7a9-3ef43e8eb788-bluemix:a715950065d0f25d3c7cf0ea9156cc395d4ba89bbc0bff5a994e4e40791e252a@22619f6f-dae8-4791-b7a9-3ef43e8eb788-bluemix.cloudant.com" } } ] }
```



# Parsing VCAP\_SERVICES

- Node

```
if (process.env.VCAP_SERVICES) {  
    var env = JSON.parse(process.env.VCAP_SERVICES);  
    var credentials = env['mysql-5.5'][0].credentials;  
    ...  
}
```

- Ruby

```
mysql_dbs = JSON.parse(ENV['VCAP_SERVICES'])["mysql-5.5"]  
credentials = mysql_dbs.first["credentials"]
```

- Java

```
String vcap_services = System.getenv("VCAP_SERVICES");  
if (vcap_services != null && vcap_services.length() > 0) {  
    JdomParser root = new JdomParser().parse(vcap_services);  
    JsonNode mysqlNode = root.getNode("mysql-5.5");  
    JsonNode credentials = mysqlNode.getNode(0).getNode("credentials");  
    ...  
}
```

- Note Java buildpack parses VCAP\_SERVICES and can auto configure bound services – see [Bluemix Liberty for Java documentation](#)



# Unit testing when services are used

- Unit tests should not include calls to remote systems
- Service APIs provide a ‘contract’ between the service consumer and service provider.
- Unit tests can ensure that code under test is using the API correctly according to the contract.
  - Assumes the service provider has also tested the service implements the contract defined by the API
- Test frameworks provide facilities to ‘stub’ and ‘mock’ external calls, but ensure that the API is being called correctly when running unit tests.
  - These facilities are often called ‘test doubles’





## Exercise 6.a

Adding a service to an application

# Become a service provider

- In addition to consuming services it is possible to create new services on Bluemix
  - Private services are only available to your organisation
    - `cf cups biTestService -p "host, port, username, password"`
    - Specify the ‘properties’ you want to make available to applications binding to the service.
  - Public services, that can become an additional revenue stream, can only be added through the IBM Cloud Marketplace



# Extend your reach through the IBM Cloud marketplace

Cloud customer buying behaviors are evolving, and we're evolving the way we sell as a result

## Paving New Avenues of Growth

ibm.com/marketplace

## New Markets + New Buyers = Growth

- **Marketplace** is IBM's digital channel for all things cloud
  - 400+ IBM and business partners and Services
  - Delivers a self-service experience
  - Guides users to discover, learn, try and buy
- **Strategic access** is IBM's competitive advantage
  - IBM and partner services in a single user experience
  - Trusted, Enterprise-grade SaaS, PaaS, IaaS
  - IBM expertise focusing on solutions so customers can innovate quickly and easily
- Multiple integration and capabilities – SoftLayer, patterns, Bluemix.



# Business Partners and IBM Cloud marketplace

## Qualified business partners can provide services that:

- Run on or be deployable to SoftLayer, or integrate with an IBM premium platform service, or integrate with or extend our SaaS offerings
  - Such services require additional IBM review, as services for IBM premium platform services must meet specific criteria which could include additional technical integration and/or certification.
- Are enterprise class, in production and supporting paying customers
- Are single or multi-tenant, as-a-service cloud applications; or virtual machine images; or add-on, plug-in or extension to a virtual machine environment; or virtual configured IT environment that includes application code and is deployed on the cloud (e.g. Patterns, or private cloud deployment)
- Align to IBM strategic imperatives

<http://ibm.biz/newway4partners>



# How do Business Partners participate in IBM Cloud marketplace?

The screenshot shows the 'IBM Cloud marketplace business partner nomination form'. It includes fields for 'Tell us about you' (First name, Last name, Email address, Phone number), 'Manage Profile' (Splash title, Splash description, Overview image), and a file upload section for 'Browse...'. A note at the top states: 'We welcome your company's request to be a marketplace business partner on the IBM Cloud Marketplace, and provide your services for sale to customers in the marketplace. We will review your nomination and contact you if your service is approved to be listed in the marketplace. If your service is approved, we will contact you to discuss the marketplace terms and conditions of IBM's partners.'

## 1. Apply to Become a Business Partner

Complete and submit the nomination form <https://www.marketplace.ibmcloud.com/joinnow/> to be considered for partnership in the IBM Cloud marketplace.

## 2. Qualify for the IBM Cloud marketplace

We will review your form and reply as quickly as possible to discuss your service and next steps. Some replies take up to two weeks, depending on your service. To qualify for premium programs (e.g. Bluemix), services must meet additional criteria.

## 3. Accept the terms and conditions

Once you are invited to include your service, you will need to a) register for marketplace and b) accept the IBM Cloud Marketplace Business Partner terms > <https://developer.ibm.com/marketplace/docs/vendor-guide/business-partner-agreements/>

## 4. Integrate and price your service to marketplace (3 weeks)

Within 90 days of signing the IBM Cloud Marketplace Business Partner agreement, you must integrate your service to the IBM Cloud marketplace APIs. Services designed for Bluemix must support an additional call, and integration and quality review for Bluemix may take several additional weeks. Details in the Business Partner Guide here > <https://developer.ibm.com/marketplace/docs/vendor-guide/>

## 5. Qualify for integration to premium options (e.g. Bluemix) - Can be in parallel with above

Prospective Bluemix, PureApplication Service on SoftLayer, or HPC Analytics services must meet certain additional criteria.

## 6. Promote and merchandise your offering

Once published, reference the IBM Cloud marketplace marketing kit to see how you can promote your offering with IBM > <https://developer.ibm.com/marketplace/docs/marketing-kit/>





## Exercise 6.b

Creating a user-provided service

# Summary

- The `VCAP_SERVICES` environment variable allows your application to discover bound services and provides the information required to access the service.
- Bluemix allows new services to be added through the IBM Cloud Market Place.
- SoftLayer can be used to host services.
- User-provided services (private to an organization) can be created or accessed the same way as a Bluemix provided service.





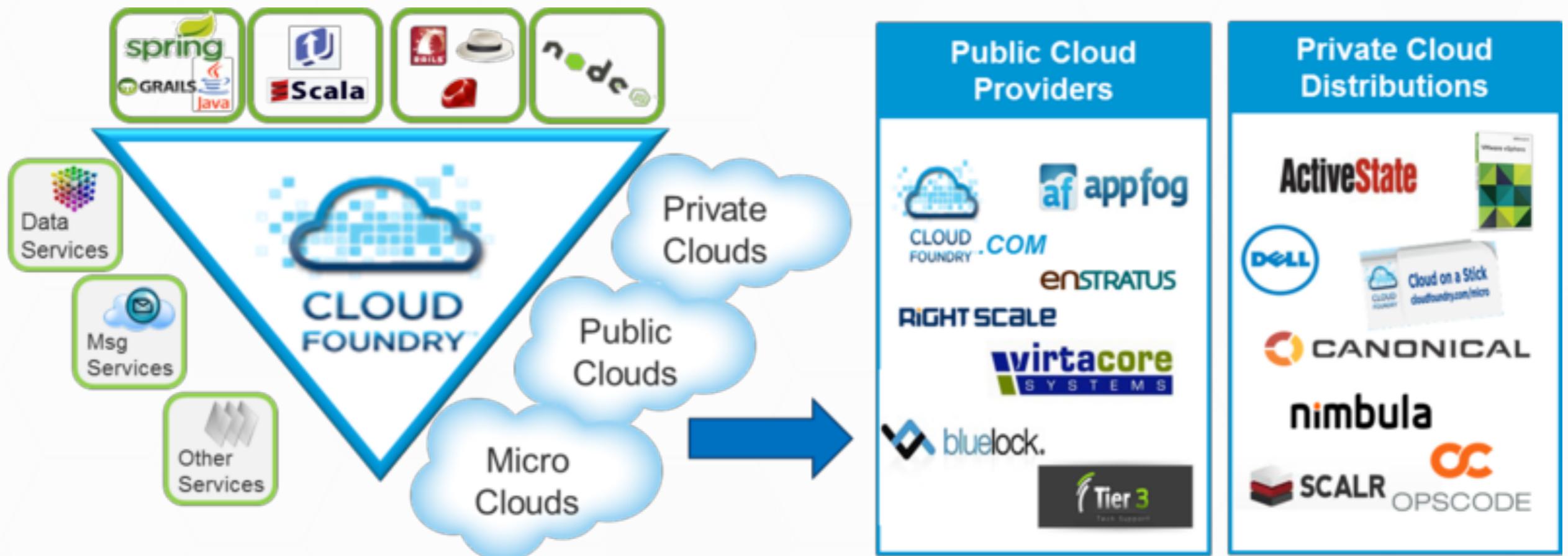
# IBM Bluemix

## Hands on Workshop

### Section 7 – How it works

# What is Cloud Foundry?

An open platform-as-a-service (**PaaS**). The system supports **multiple** frameworks, **multiple** application infrastructure services and deployment to **multiple** clouds.

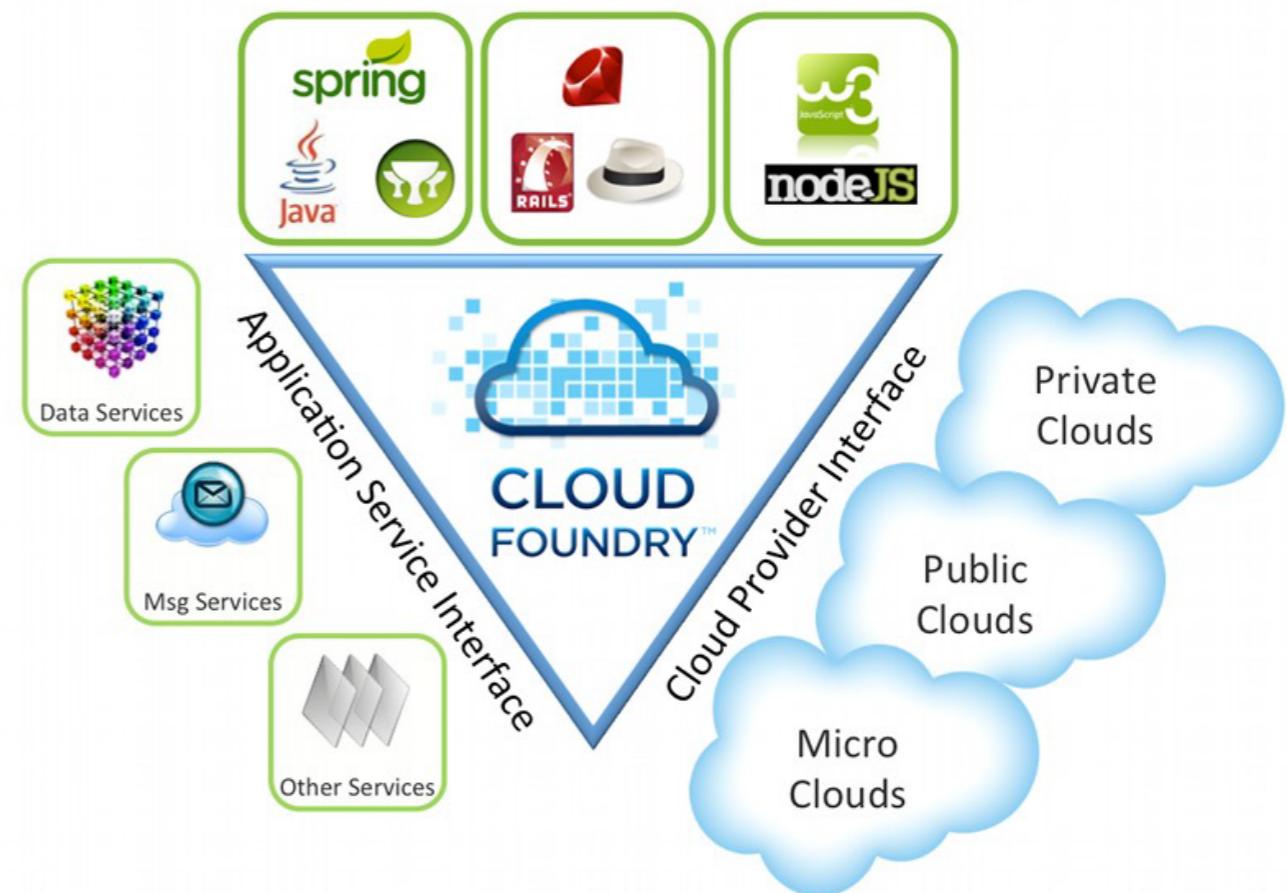


**Making Multi-Cloud a Reality**

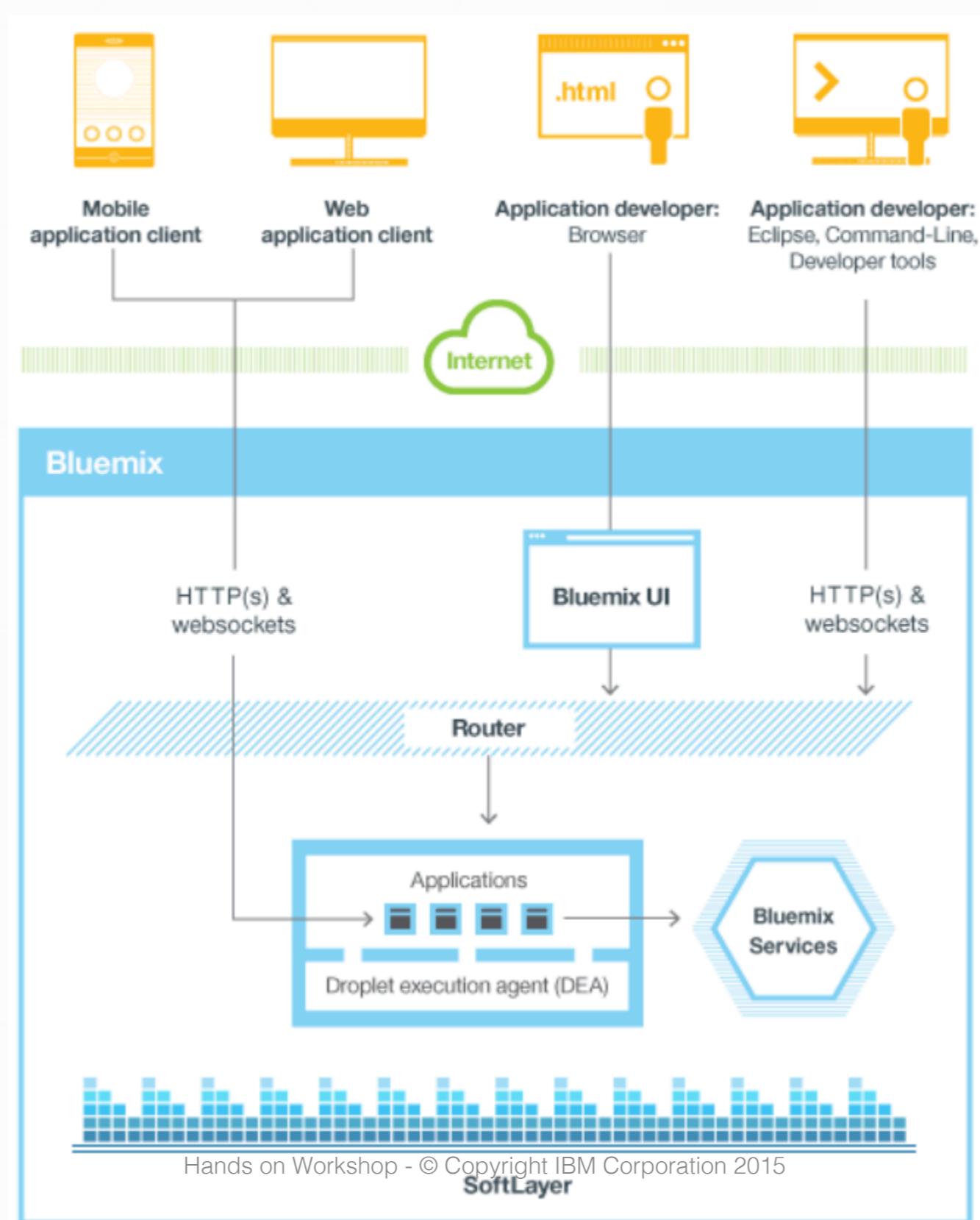


# Languages/Frameworks/Service

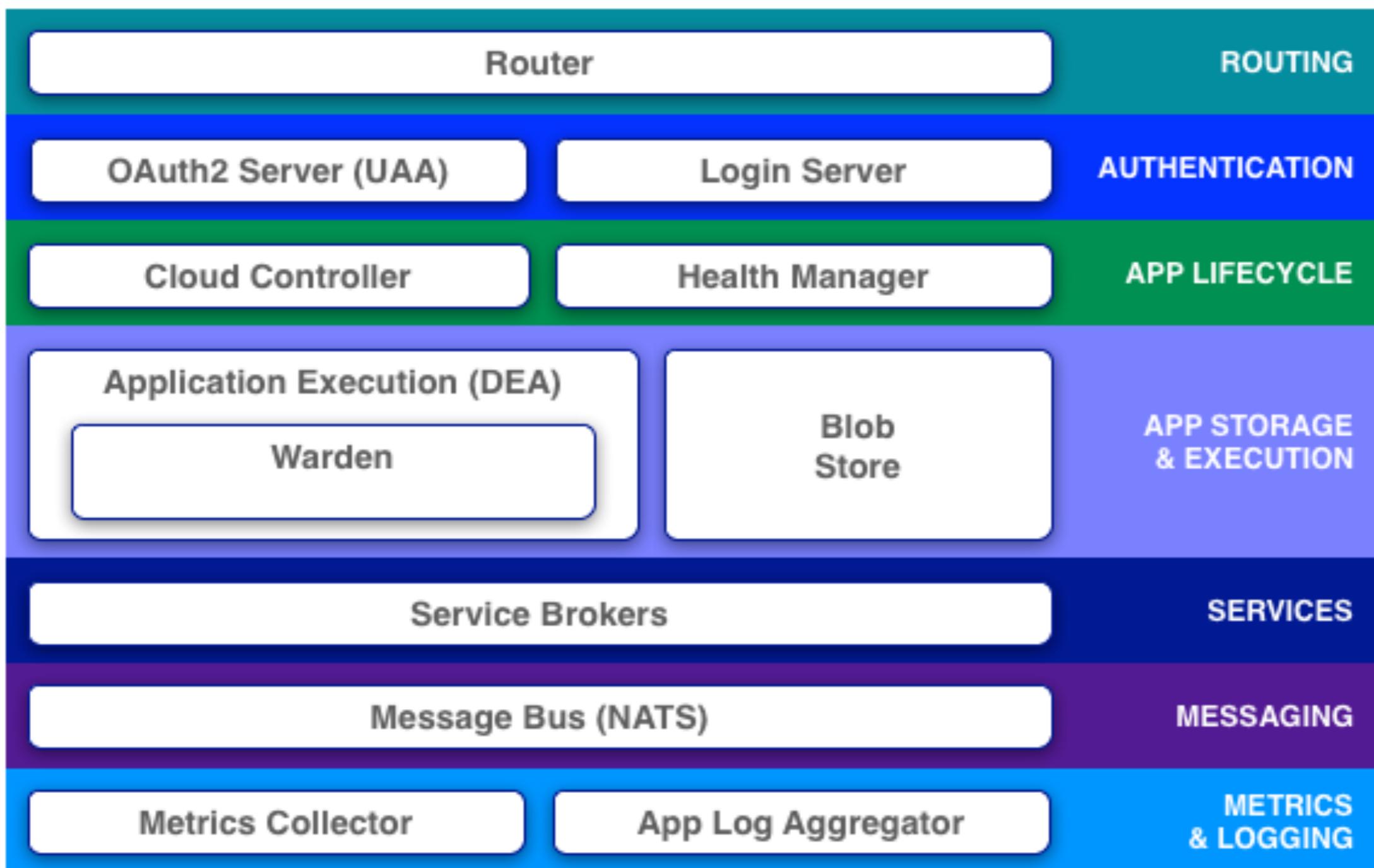
- Multi-Language
  - Ruby, Java, Scala, Node.js, Erlang, Python, PHP..
- Multi-Framework
  - Rails, Sinatra, Spring, Grails, Express, Lift
- Multi-Services
  - MySQL, Postgres, MongoDB, Redis, RabbitMQ
- Multi-Cloud, Multi-IaaS
  - Public Cloud, MicroCloud, Private Cloud



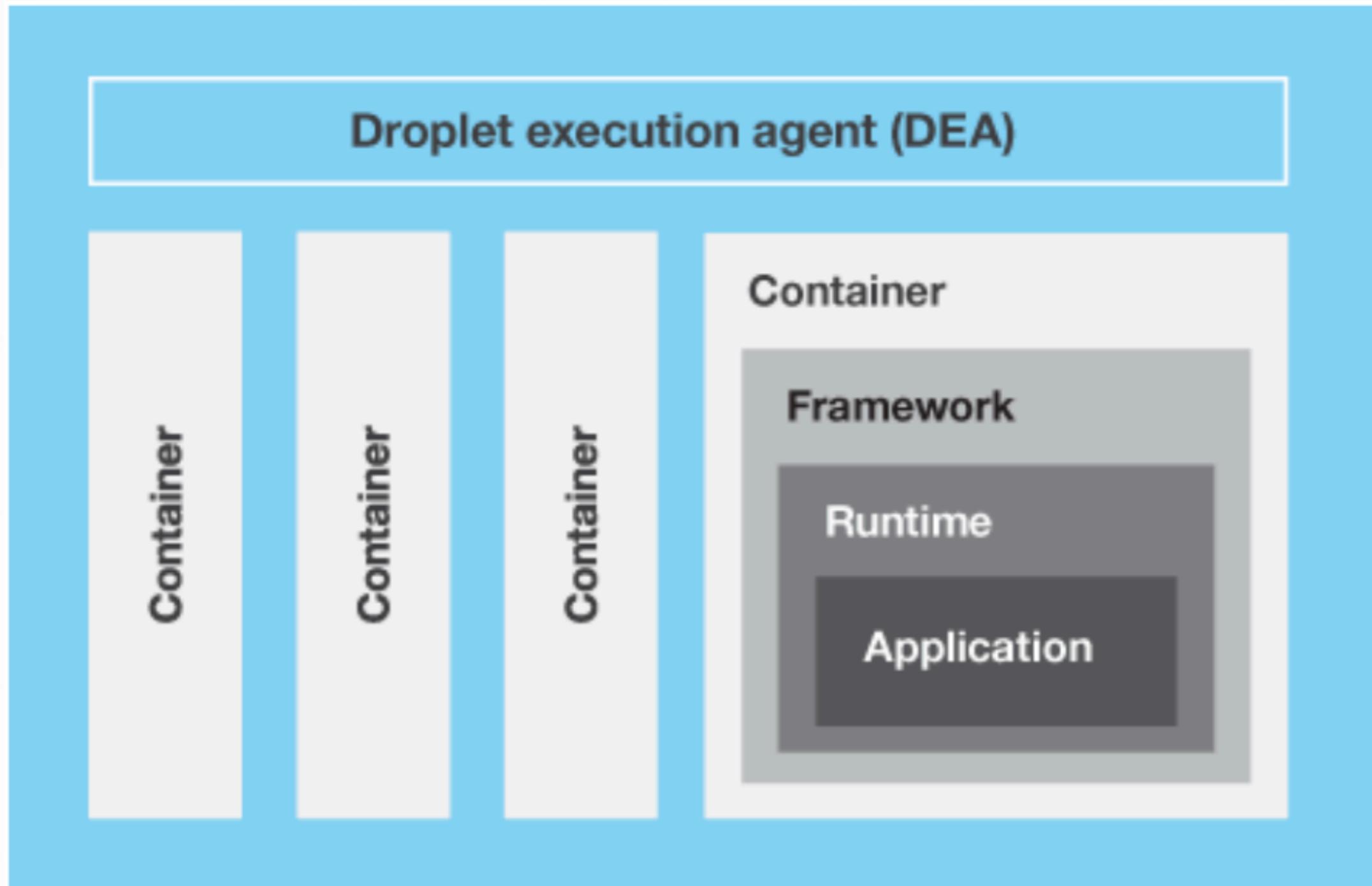
# Bluemix Architecture



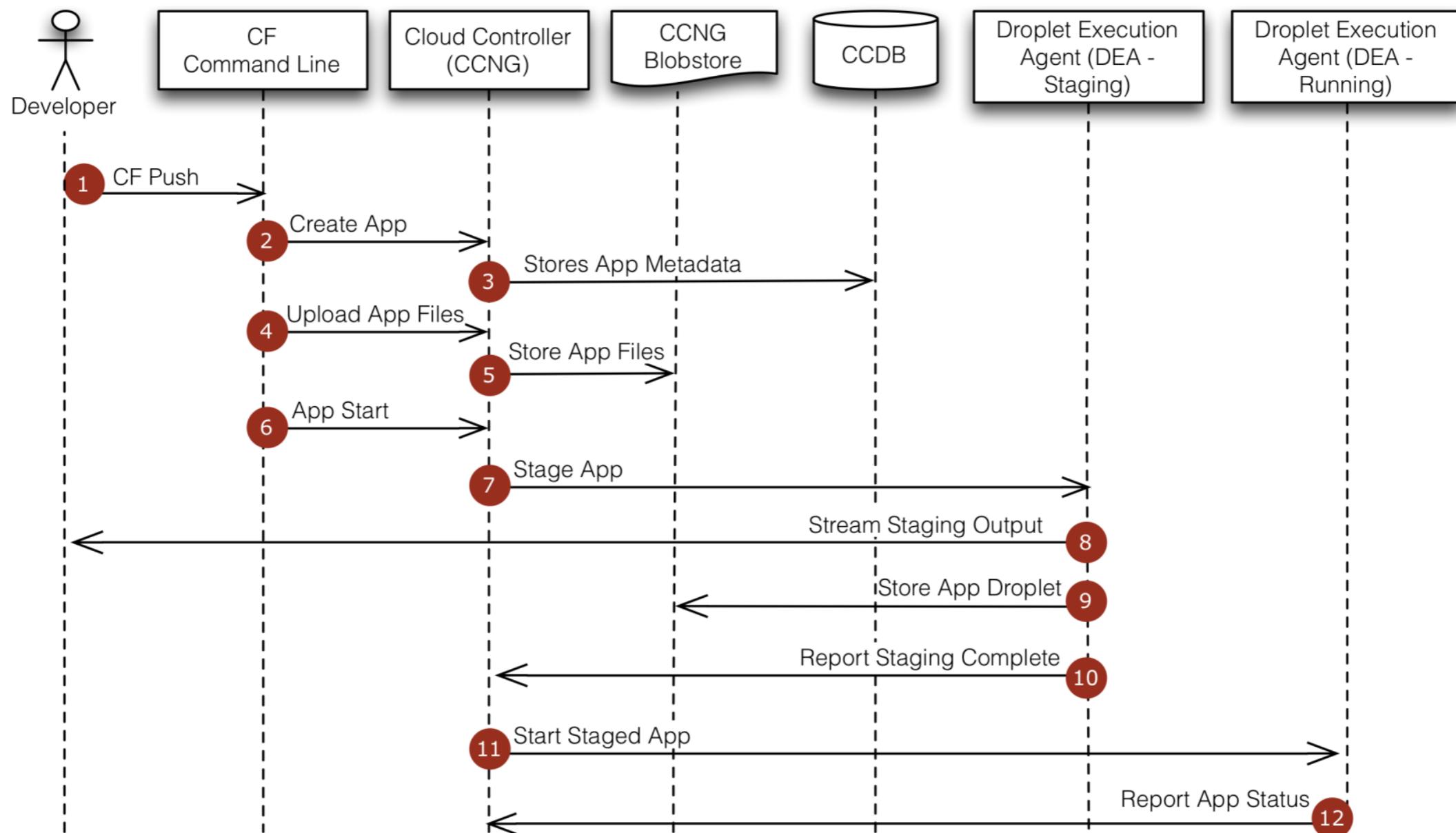
# Cloud Foundry Kernel Internal



# Design of a VM



# Cloud Foundry - Application Staging



# Choosing the runtime for an application

- If you don't specify a runtime when deploying an application Bluemix will try to find a suitable runtime
  - Each pre-defined runtime has a position set and runtimes are tried in ascending position order
  - Bluemix asks each runtime, starting with runtime at position 1, if they can run the application
  - First runtime that responds it is able to run the application is used
  - If no runtime can run the application an error is returned
- At deploy time you can specify the runtime to use



# Buildpacks

- Buildpacks provide the framework and runtime support for your application
- They will determine if they can run your application as part of the ‘autodetect’ mechanism Bluemix uses when you push an application
- You can control which platform provided buildpack your application should use or specify an external buildpack to be used to run your application = “Bring your own Buildpack”



# Specifying a buildpack

- CLI allows you to specify which buildpack to use when deploying your application using the ‘**-b**’ options with ‘**cf push**’.
- Can specify the name of an internal buildpack or provide the URL for an external buildpack
- ‘**cf buildpacks**’ shows the installed internal buildpacks available:

buildpack	position	enabled	locked	filename
liberty-for-java	1	true	false	buildpack_liberty-for-java_v1.12-20150130-1016-yp.zip
sdk-for-nodejs	2	true	false	buildpack_sdk-for-nodejs_v1.12-20150130-1059-yp.zip
noop-buildpack	3	true	false	noop-buildpack-20140311-1519.zip
java_buildpack	4	true	false	java-buildpack-v2.6.zip
ruby_buildpack	5	true	false	ruby_buildpack-offline-v1.2.0.zip
nodejs_buildpack	6	true	false	nodejs_buildpack-offline-v1.1.1.zip
go_buildpack	7	true	false	go_buildpack-offline-v1.1.1.zip
python_buildpack	8	true	false	python_buildpack-offline-v1.1.1.zip
php_buildpack	9	true	false	php_buildpack-offline-v1.0.2.zip
liberty-for-java_v1-11-20150119-1511	10	true	false	buildpack_liberty-for-java_v1.11-20150119-1511-yp.zip
sdk-for-nodejs_v1-11-20150115-2258	11	true	false	buildpack_sdk-for-nodejs_v1.11-20150115-2258-yp.zip

- **cf push myApp -b nodejs\_buildpack**
- **cf push myApp -b https://github.com/dmikusa-pivotal/cf-php-build-pack.git**

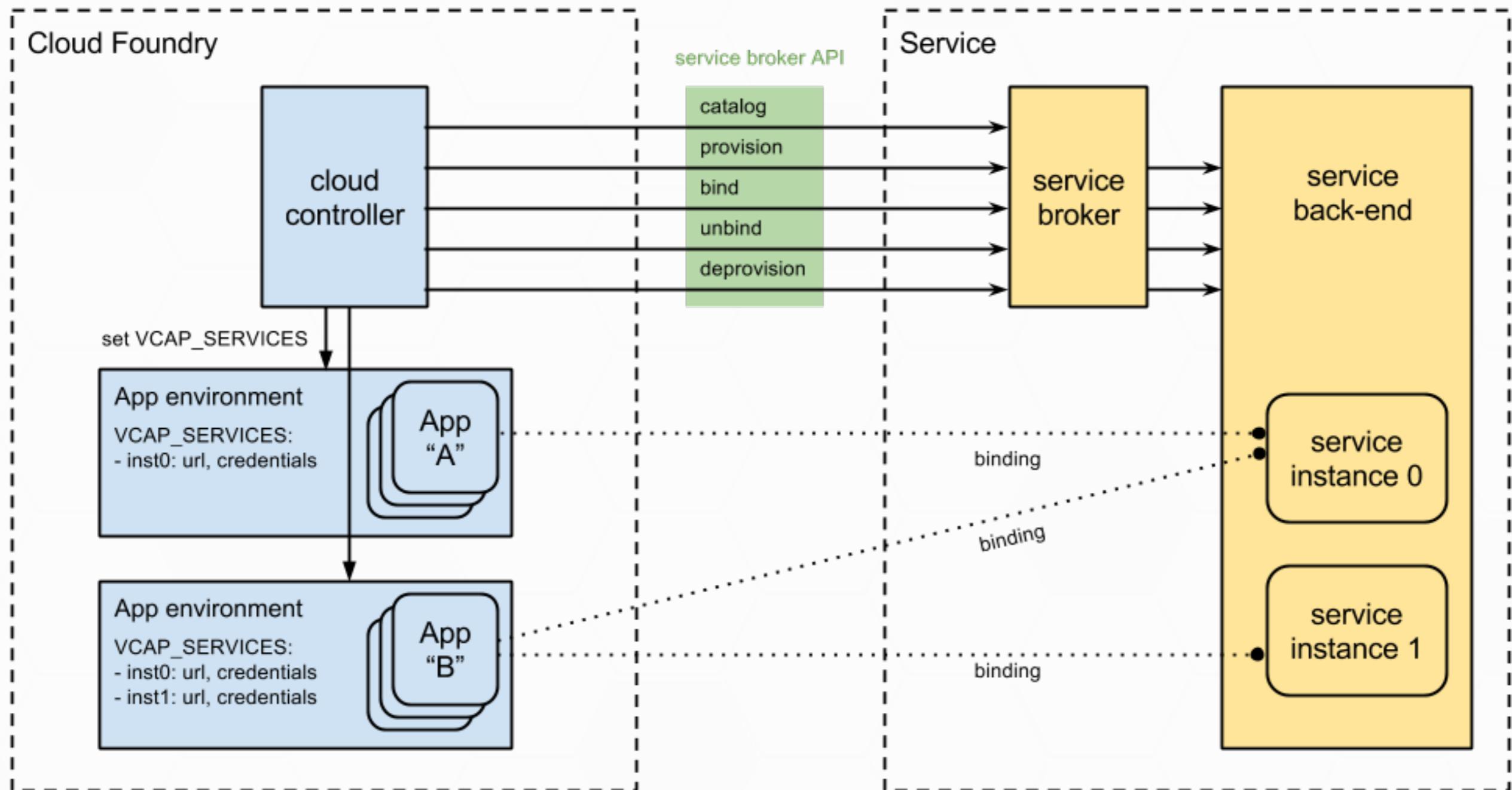




## Exercise 7.a

Specifying a buildpack when  
deploying an application

# Cloud Foundry - Services



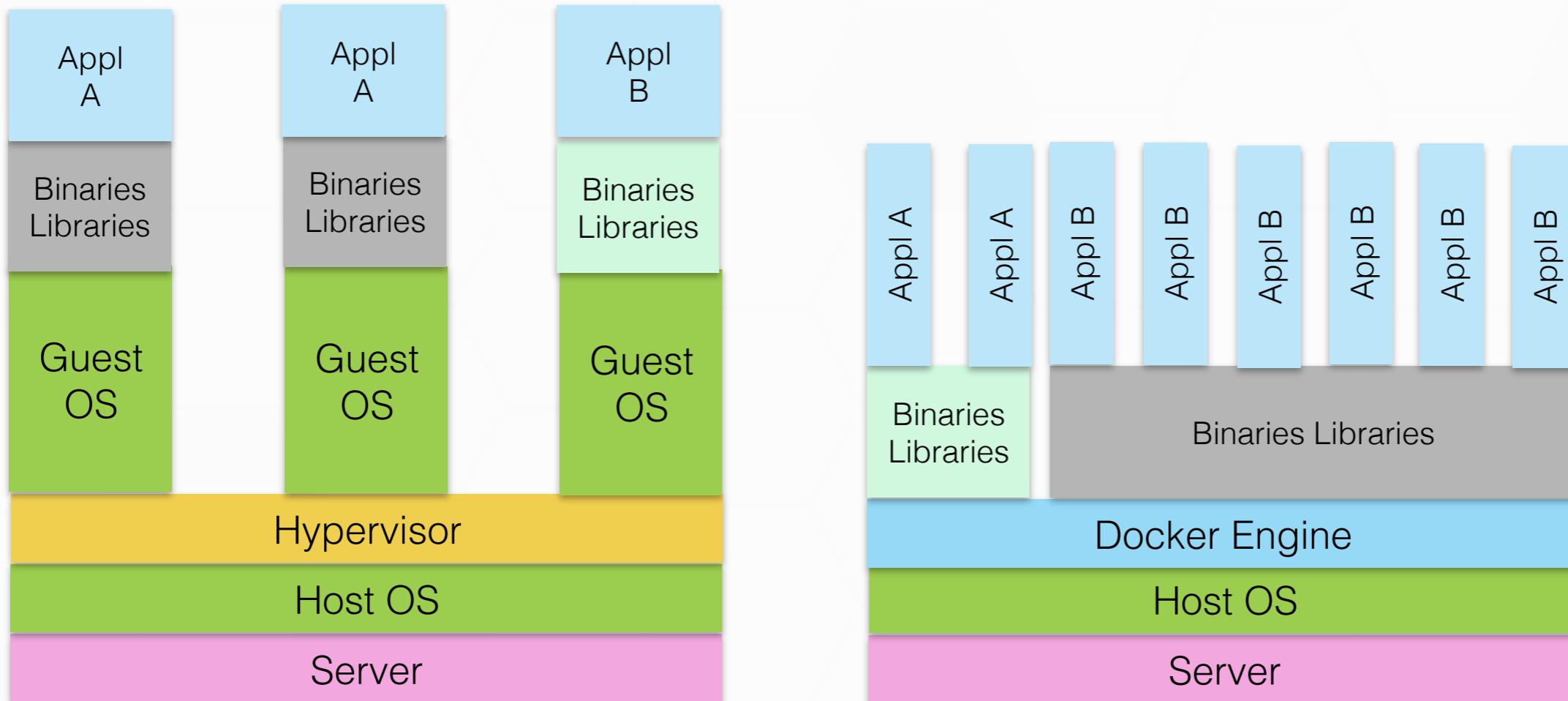


# IBM Bluemix

## Hands on Workshop Section 8 – Containers

# What is Docker?

## VM vs Docker



Docker = Linux namespaces + cgroups + overlay file system + image format

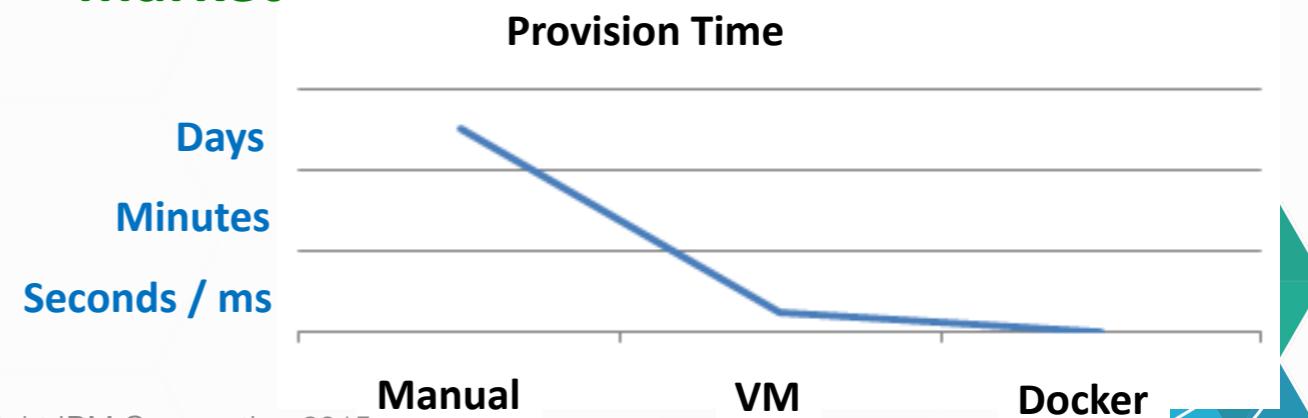


# Why Docker?

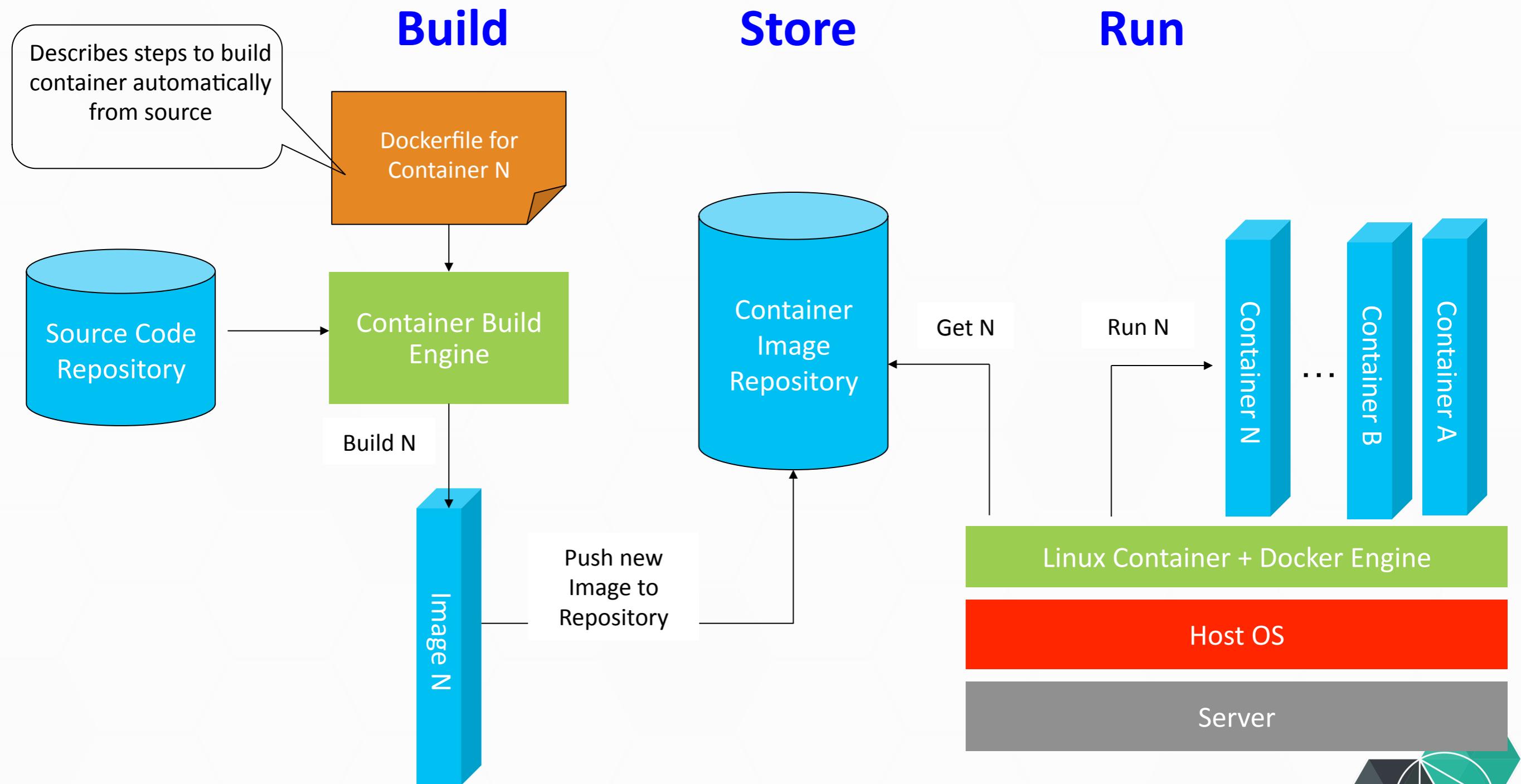
- **High Density:** Because containers share the same OS image and middleware (when applicable) we can run more applications on a server.
  - For example: If the full stack image required to run our application requires 1GB, if we wanted to use a Full VM we would need to have 1GB times the number of VMs we want. With LXC containers and AuFS we can share the bulk of the 1GB and we could have 100s of containers and still use a little over 1GB (assuming they are all running the same OS image).
- **Fast Start up:** Because containers may have several layers in common only the new layers need to be copied, reducing build/transfer/boot/load times dramatically.
  - A full virtualized system usually takes ~ 10 minutes to start, LXC containers take seconds, and sometimes even less than a second.
- **Portability** across environments
  - Deploying a consistent production environment is hard. Even if you use tools like chef and puppet, there are always OS and library updates that change between hosts and environments.
  - Docker gives us the ability to snapshot the OS into a common image, and, when combined with IBM's patterns technology will make it easy to deploy a collection of images comprising a given workload in another collection of Docker hosts.

- Cloudy operations with Docker LXC outperform VM
  - 48x server reboot, 1.5x server boot, 1.62x server snapshot, etc.
- Docker LXC density potential compared to VMs
  - 3x memory savings
  - 26x CPU savings
  - 3.22x smaller images in this test (note – image sizes can vary based on required packages)

**Significant threat to virtualization market**



# What are Docker's basic functions?



# Example of a Dockerfile

- **Dockerfile (for WebSphere Liberty Profile)**

```
FROM ubuntu:12.04
# add the files we require, jar + WLP files
ADD wlp-developers-extended-8.5.5.2.jar /root/
ADD wlp-developers-runtime-8.5.5.2.jar /root/
ADD JAXWSEJBSample.jar /root/
# install WLP
RUN apt-get update
RUN apt-get install -y default-jre
RUN java -jar /root/wlp-developers-runtime-8.5.5.2.jar --acceptLicense /root/
RUN java -jar /root/wlp-developers-extended-8.5.5.2.jar --acceptLicense /root/
RUN cd /root/wlp && java -jar ../JAXWSEJBSample.jar /root/wlp
EXPOSE 9080
CMD /root/wlp/bin/server run JAXWSEJBSample
```





## Exercise 8.a

Creating a container from  
DevOps Services (if service live)

# Summary

- The advantages of containers:
  - Containers and VMs allow you to easily bring applications to Bluemix.
  - They provide a way to virtualize and partition physical infrastructure.
  - They allow IaaS and PaaS to converge and be co-located.
  - They support a large scale deployment with low latency.
  - Containers provide portability across environments.





# IBM Bluemix

Hands on Workshop  
Section 9 – Beyond the 30-day  
trial

# End of 30 day trial

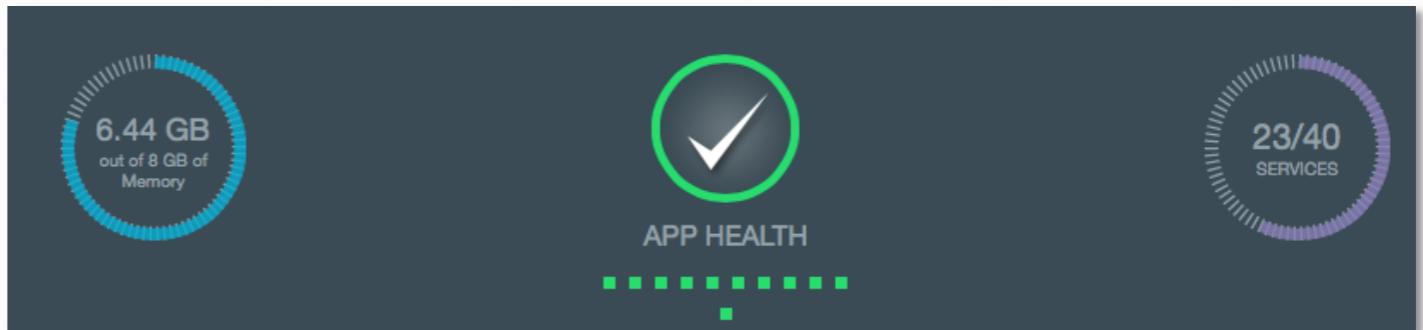
- Once your 30 day trial ends you will be asked to enter a credit card to continue using Bluemix.
- You are still able to use Bluemix for free after you have entered your credit card details as there is a 'freemium' level of service available:
  - 375 GB-hours are free every month
  - Services have a free plan
  - Use the [Pricing Sheet](#) and price calculator to verify what you can run for free each month
  - Use your account summary to check on your monthly usage
  - Set notifications to advise when cost thresholds are being reached

$$\text{GB-hours} = \text{Total GB/App} \times \text{Number of App Instances} \times \text{Total Hours running}$$



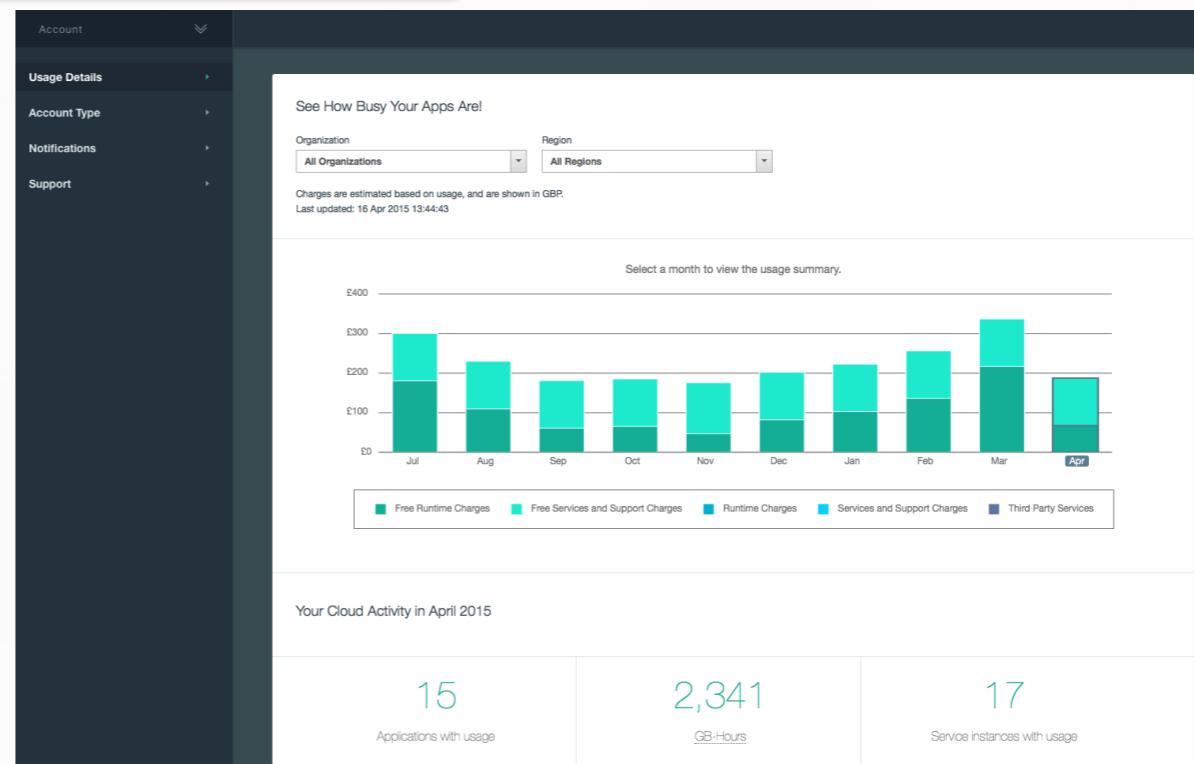
# Use the Bluemix UI to manage your Freemium usage

## Dashboard



SDK for Node.js™  
Default  
Run one or more apps free for 30 days (375 GB-hours free).  
£0.0424 GBP / GB-Hour  
Instances: 2  
Memory: 256 MB

## Pricing Calculator





# IBM Bluemix

Hands on Workshop  
Section 10 – Wrap up

# Summary

- Now you have completed the course you should:
  - Understand how Bluemix regions and spaces work
  - Understand how to develop applications using Eclipse, the Cloud Foundry command line tool or online, using IBM Bluemix DevOps Services
  - Understand how to maximize the value of IBM Bluemix:
    - User centred design
    - Agile methodology
    - DevOps pipeline
    - Microservices Architecture
  - Understand how to create and consume services on Bluemix
  - Understand the runtime options using Cloud Foundry, Containers or Virtual Machines



# What next?

- You have all the workshop material to take away, complete any labs not completed during the workshop.
- Use the online documentation and the dW & **stackoverflow** forums to get answers to your questions.
- All services have documentation and sample applications to help you get started.
- Look out for future meetups, hackathons and deep-dive Bluemix workshops on specific topics, such as Big Data and Analytics.





**Thank you for attending**