IBM Cloud Developer Certification Overview

Presented by:

David Carew

Sandhya Kapoor

IBM Cloud Ecosystem Development

IBM **Cloud**

Cloud Developer Certification V1
Target Audiences and Competencies Assessed

The *IBM Certified Application Developer - Cloud Platform V1* certification is testing developer's foundational competencies in open technologies and best practices specific to IBM's Cloud offerings

1. Demonstrate knowledge in typical cloud architectures and designs
2. Implementing cloud applications using DevOps and Monitoring
3. Distinguishing between different data services on IBM Bluemix and managing data in SQL and NoSQL databases
4. Optimizing applications using Data Cache, Messaging Frameworks and Object Storage (for large media files)
5. Designing cloud applications on portable open technologies like CloudFoundry and Git repositories

Audience	What value will Level 1 Cloud Developer Cert provide?
Cloud Deployment Leads and Technical Architects	Evaluate organization cloud strategy
Cloud Developers and Technologists	Undertake decisions related to Cloud Computing rollout
Channel partners	Cash in on the lack of certified cloud computing competency skills
Students and professionals	Make structured entry into a professional career in Cloud
Businesses hiring consultants	Hit the ground running in portable Open Technology Cloud

IBM Cloud Developer Certification

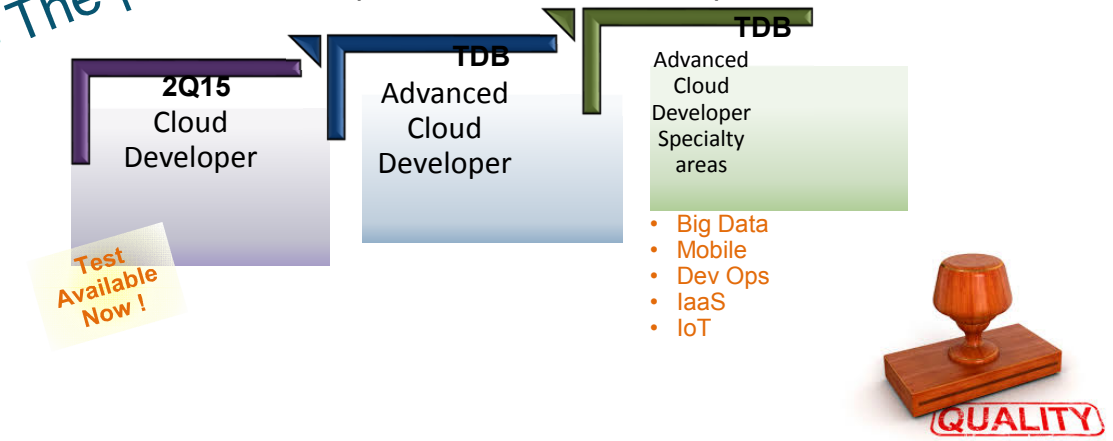
Goal: Deliver Certification on Cloud for Developers

Focusing on delivering quality and relevance in the market

- **Loyalty**
- **Job Opportunity**
- **Recruitment via influence**
- **Demonstrate knowledge of open technologies, best practices and IBM spec[illegible] cloud technologies**

The plan

Multiple levels of cloud developer certification



Targeting Enterprise Developers, ISVs, SIs, Individual Developers, Startups, Students

Why IBM Cloud Platform Developer Certification

- Focus is on essential cloud developer skills in a language neutral way
- Covers **all** essential skills and services
  - e.g. DevOps, Data Services (SQL and NoSQL), Messaging, Caching, SSO, Scaling, Monitoring and Analytics
- Requires knowledge of IBM specific cloud services as well as portable open technologies :
  - Bluemix, IBM DevOps Services, Git, CloudFoundry, Docker, OpenStack
- Requires knowledge of universally accepted best practices for developing, deploying and maintaining cloud applications
  - e.g. 12 Factor methodology
- Value for GTPs
  - Universal appeal
  - Blueprint for training enterprise developers or new developers
- Value for GSIs
  - Can demonstrate that developers have well rounded cloud developer knowledge as well as Bluemix specific knowledge

Agenda for today

- Drill down into all 6 Sections covered in the study guide to get you started
  - **Lecture 1 – Sections 1 – 3**
  - **Lecture 2 – Section 4**
  - **Lecture 3 – Sections 5 and 6**

- We'll end the day with the Assessment test for this certification exam
  - You'll leave with a flavor of what the actual test is like
  - A blueprint for further study because the score report will provide details of the score by section

Section/Category	Your Score
Overall	77% (37 out of 48)
Section 1: Hosting Cloud Applications	100% (4 out of 4)
Section 2: Planning Cloud Applications	75% (6 out of 8)
Section 3: Implementing Cloud Ready Applications	90% (9 out of 10)
Section 4: Enhancing Cloud Applications Using Managed Services	67% (4 out of 6)
Section 5: Using DevOps Services & Tools to Manage Cloud Applications	89% (8 out of 9)
Section 6: Using Data Services	55% (6 out of 11)

About the test

- 48 questions, 90 minutes allotted
- Passing score 66%
- Outline
  Section 1 – Hosting Cloud Apps (4 questions)
  Section 2 – Planning Cloud Apps (8 questions)
  Section 3 – Implementing Cloud Ready Apps (10 questions)
  Section 4 – Enhancing Cloud Applications Using Managed Services (6 questions)
  Section 5 – Using DevOps Services & Tools to Manage Cloud Apps (9 questions)
  Section 6 – Using Data Services (11 questions)

Cloud Application Developer Certification Review Part 1 Sections 1 - 3

Presented by:

IBM Cloud Ecosystem Development

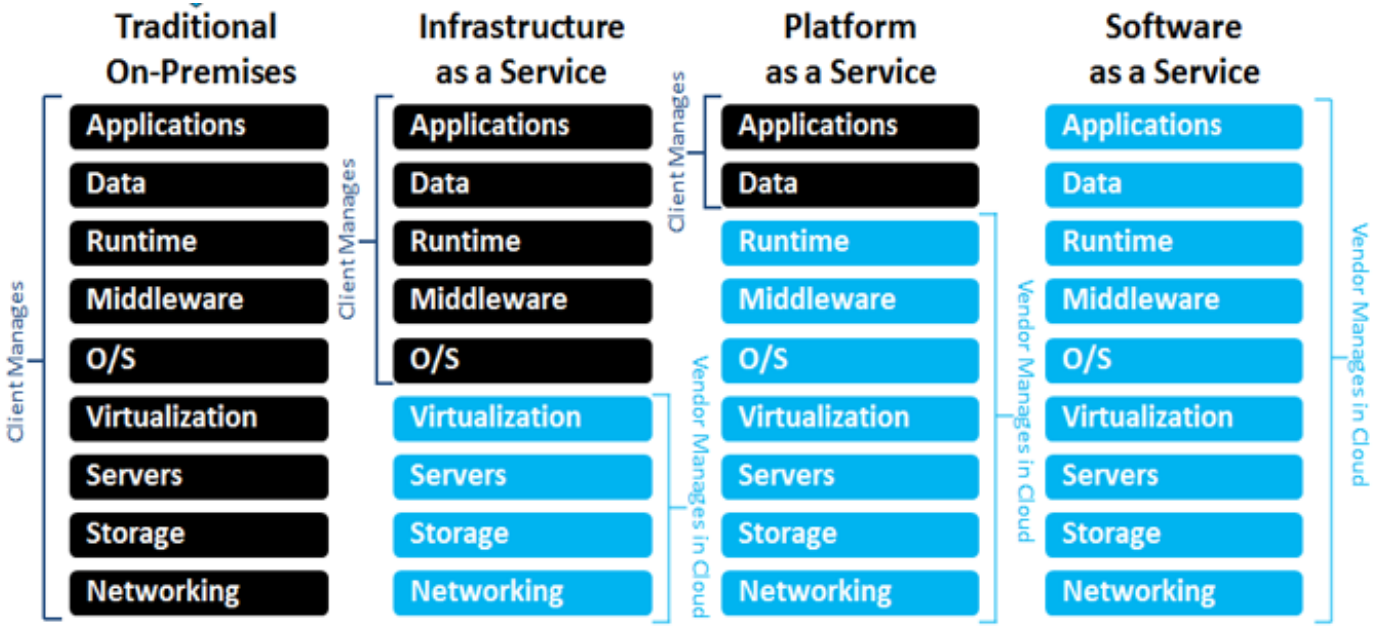
IBM **Cloud**

Section 1: Hosting Cloud Applications

a. Describe Cloud service models and IBM Cloud offerings
   - IBM SoftLayer Infrastructure as a Service (IaaS)
   - IBM Bluemix Platform as a Service (PaaS)
   - Software as a Service (SaaS) and IBM Cloud Marketplace

b. Describe the different capabilities of IBM Bluemix
   - IBM Bluemix PaaS provided by Cloud Foundry
   - IBM Bluemix Containers using docker
   - IBM Bluemix virtual machines powered by OpenStack

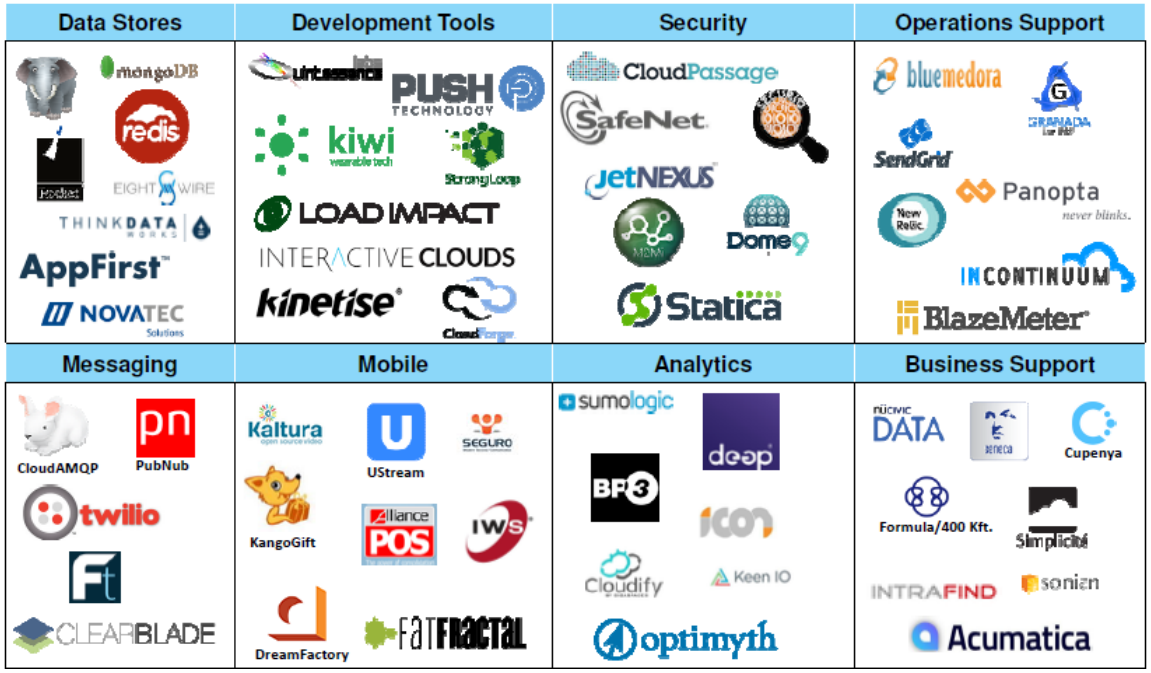
Cloud Services Models and IBM Cloud Offerings – 1a



Customization; higher cost; slower time to value

Standardization; lower cost; faster time to value

IBM Cloud Marketplace – Cloud Services Store – 1a



- Is a dedicated area on ibm.com/cloud for IBM customers to discover enterprise cloud services that run on or are deployable on SoftLayer or integrated with Bluemix and are available from IBM & Business Partners. ibm.biz/newway4partners

Bluemix offers different deployment models to run your code – 1b

Cloud Foundry application

Docker based Container

Virtual Machine

expanding

Instant Runtimes

App-centric runtime environments based on Cloud Foundry.

CLOUD FOUNDRY

IBM Containers

Portable and consistent delivery of your app without having to manage an OS.

NEW

docker

Virtual Machines

Get the most flexibility and control over your environment with VMs.

openstack

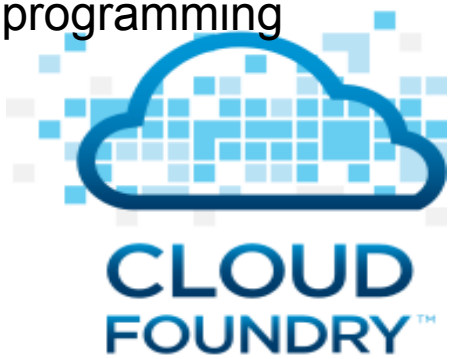
POWERED

Cloud Foundry – an Open PaaS – 1b

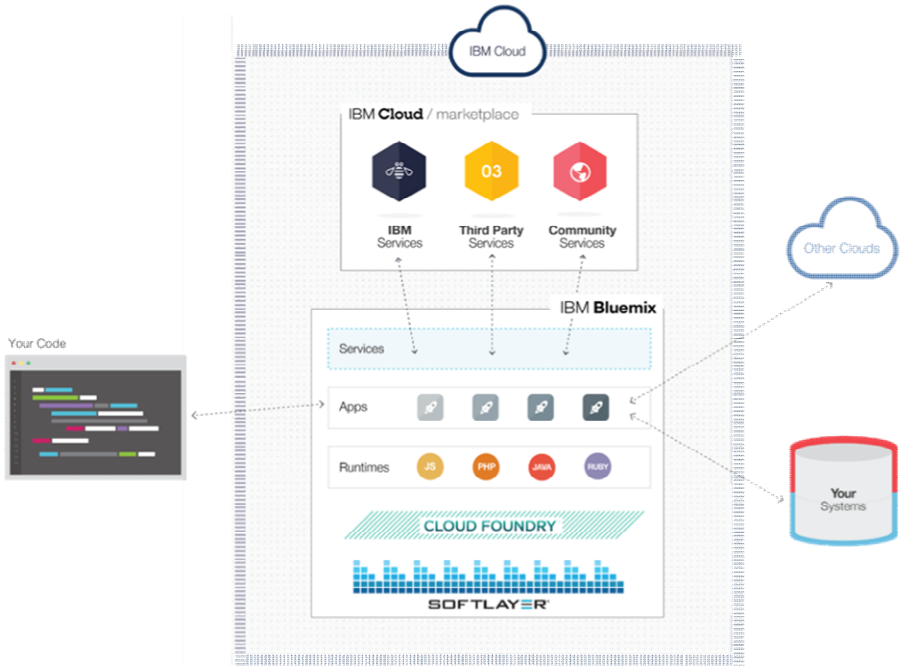
- Open Standards, Open Governance model
  - Open and extensible nature

- Interoperable Platform as a Service framework
  - Enable rapid application development, deployment and scaling of new cloud-centric applications
  - Runs on broad range of cloud infrastructure platforms, without fear of vendor lock-in
  - Supports a wide range of application programming runtimes and frameworks

- IBM uses Cloud Foundry in Bluemix



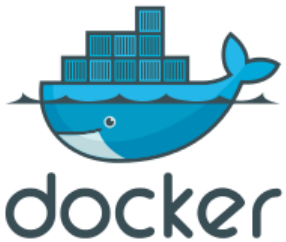
IBM Bluemix PaaS Advantages – 1b



- **A range of services that enable you to build and extend web and mobile apps fast.**
- **Processing power for you to deliver app changes continuously.**
- **Fit-for-purpose programming models and services.**
- **Manageability of services and applications.**
- **Optimized and elastic workloads.**
- **Continuous availability.**

IBM Containers -1b

- Docker based virtual software objects that include all the elements that an application needs to run.
- Each container includes just the code it needs to run and its dependencies
- Runs as an isolated process on the hosting operating System.
- Benefits of resource isolation and allocation
- Portable and efficient to help you build application fast



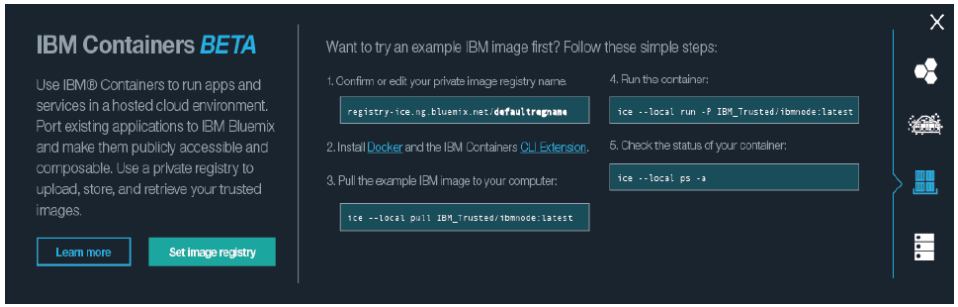
IBM Bluemix Container Advantages & Differentiators – 1b

5 Key Differentiators:

1. Push and pull containers from on-premises to off-premises service
2. Hosted private registry with access controls
3. Integrated container monitoring and logging
4. One-step public IP configuration
5. Services integration, including seamless infrastructure as a service and platform as a service with other Bluemix services such as analytics

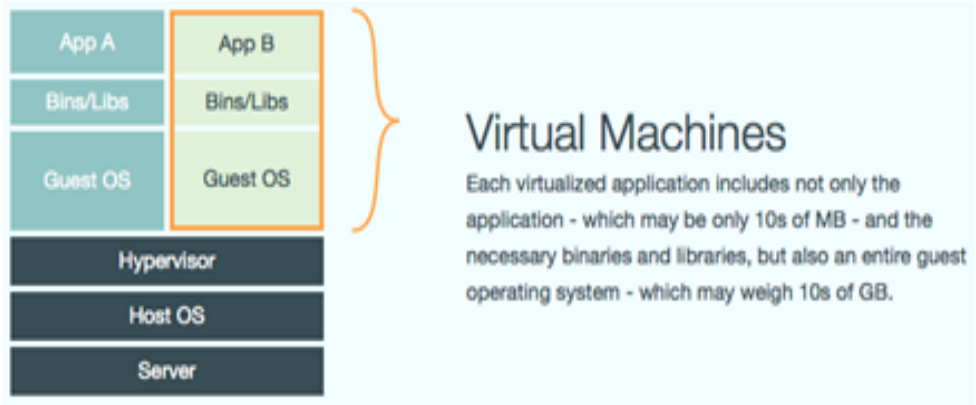
Key Advantages:

1. Automate the build of Docker images
2. Manage and distribute Docker images in private Docker registries
3. Easily host container in the cloud
4. Scale and auto-recovery built-in
5. Logging and monitoring built-in



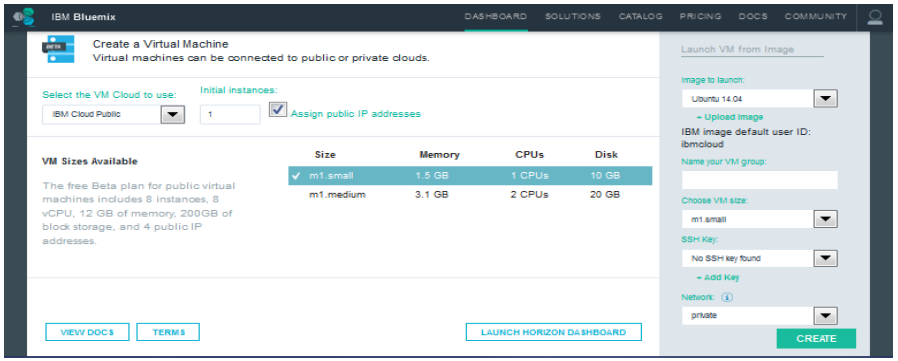
What is an Openstack Virtual Machine? – 1b

- A software implementation of a machine hosted by a real machine based on Openstack
- Includes the application, its dependencies/middlewares and a guest OS
- Managed by a hypervisor server for resources allocation and provisioning
- Multiple VMs can run simultaneously on a single host
- Guest OSs might not be the same as the host OS



IBM Openstack VM's Advantages – 1b

- Bluemix VMs infrastructure as beta feature in select regions
- Enable creation of VMs running in public and private Openstack clouds
- Allow reuse of existing VM images
- Provision VM groups with auto scaling and load balancing against a target cloud
- Manage operations via dashboard or CLI/API
- View VM info, such as instance ID, IP, CPU, Memory, Disk resources and status of VM
- Supports OpenStack project deployments, and scales both horizontally and vertically



Section 2: Planning Cloud Applications

a. Describe key components of IBM Bluemix PaaS environment

b. Describe components of IBM Bluemix PaaS architecture based on Cloud Foundry

c. Explain the process of staging an application in IBM Bluemix PaaS

d. Describe the organization management elements in IBM Bluemix PaaS: Spaces, Users, Domains and Quota

e. Understand IBM Bluemix Regions and how to manage applications in multiple regions

f. Use the Cloud Foundry CLI (cf) tool to manage applications in IBM Bluemix PaaS

Key Components in Bluemix PaaS Environment – 2a

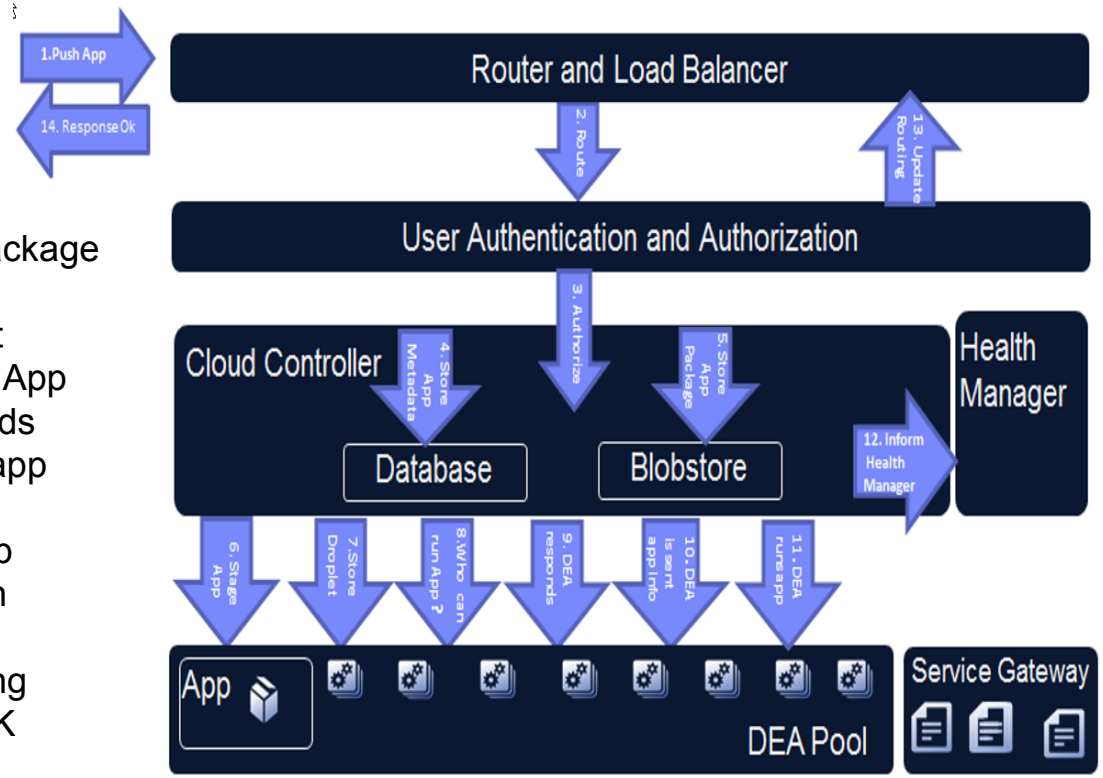
- **Starter** - A *starter* is a template that includes predefined services and application code that is configured with a particular buildpack. There are two types of starters: boilerplates and runtimes.
- **Boilerplate** - A *boilerplate* is a container for an application and its associated runtime environment and predefined services for a particular domain.
- **Runtime** - A *runtime* is the set of resources that is used to run an application. Bluemix provides runtime environments as containers for different types of applications. The *runtime environments* are integrated as buildpacks into Bluemix, are automatically configured for use, and require little to no maintenance.
- **Buildpack** - A *buildpack* is a collection of scripts that prepare your code for execution on the target PaaS. A buildpack gathers the runtime and framework dependencies of an application. Then, it packages them with the application into a droplet that can be deployed to the cloud.
- **Service** - A *service* is a cloud extension that is hosted by Bluemix. The service provides functionality that is ready-for-use by the app's running code. The predefined services that are provided by Bluemix include database, messaging, push notifications for mobile apps, and elastic caching for web apps.

Key Cloud Foundry Components in Bluemix PaaS – 2b

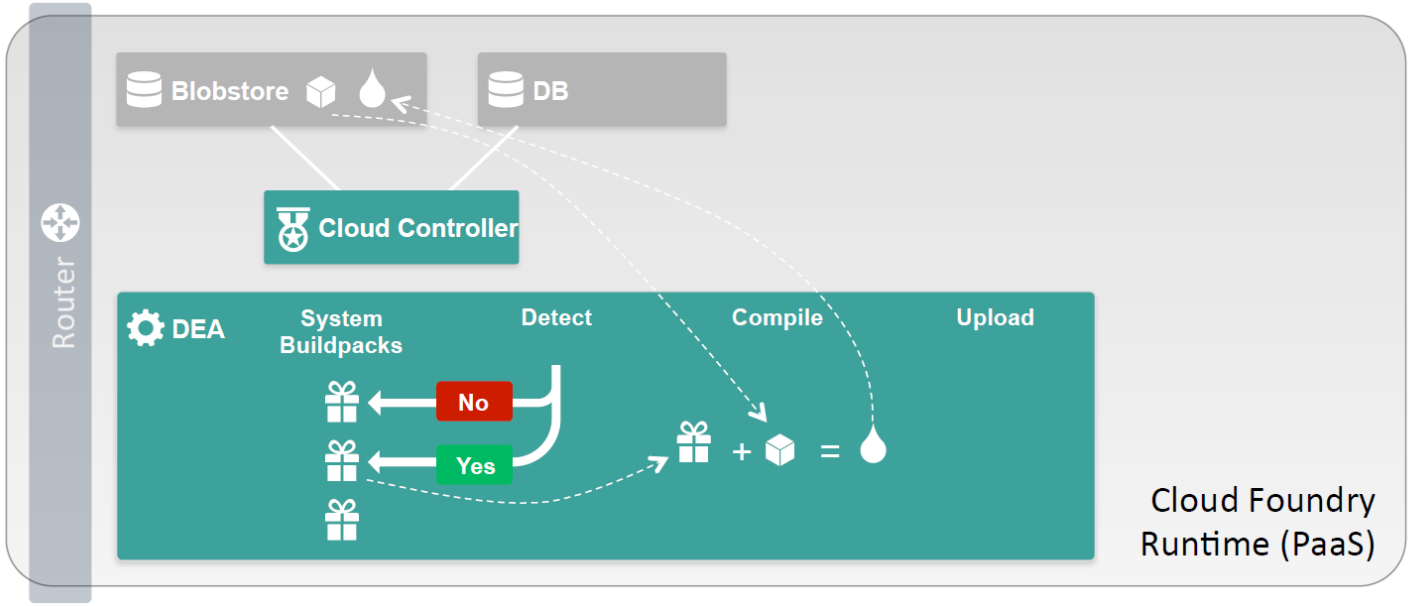
- **Droplet Execution Agent (DEA)** - Manages application instances, tracks started instances, and broadcasts state messages. Application instances live inside Warden containers.
- **Cloud Controller** - Manages the lifecycle of applications. When a developer pushes an application to Cloud Foundry the Cloud Controller stores the raw application bits, creates a record to track the application metadata, and directs a DEA node to stage and run the application.
- **Router** - The router handles all incoming traffic and forwards to the appropriate component: Cloud Controller or a running application on a DEA node
- **Service Broker** - When a developer provisions and binds a service to an application, the service broker for that service is responsible for providing the service instance.
- **Health Manager** –
  - Monitor applications to determine their state (e.g. running, stopped, crashed, etc.), and number of instances. Updates the actual state of an application based on heartbeats and droplet.exited messages issued by the DEA running the application.
  - Determines applications' expected state, version, and number of instances from a dump of the Cloud Controller database.
  - Reconcile the actual state of applications with their expected state. For instance, if fewer than expected instances are running, it will instruct the Cloud Controller to start the appropriate number of instances.

Application Deployment – 2c

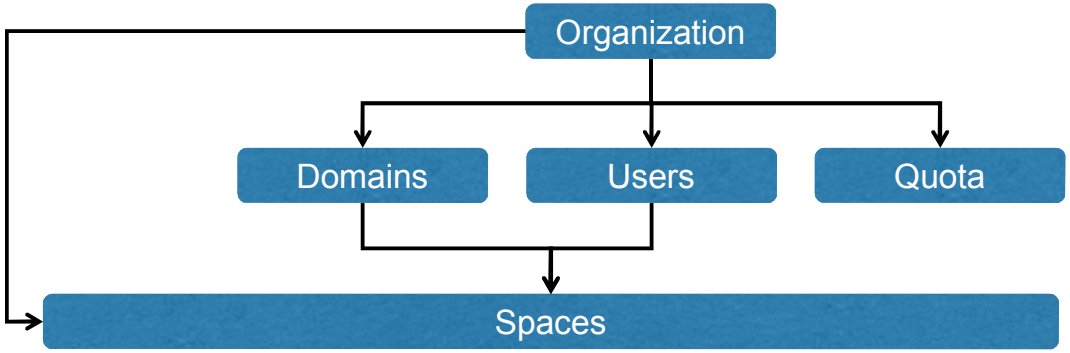
1. Push App
2. Route
3. Authorize
4. Store App Metadata
5. Store App Package
6. Stage App
7. Store Droplet
8. Who can run App
9. DEA Responds
10. DEA is sent app info
11. DEA runs app
12. Inform Health Manager
13. Update routing
14. Response OK



Staging an application – 2c



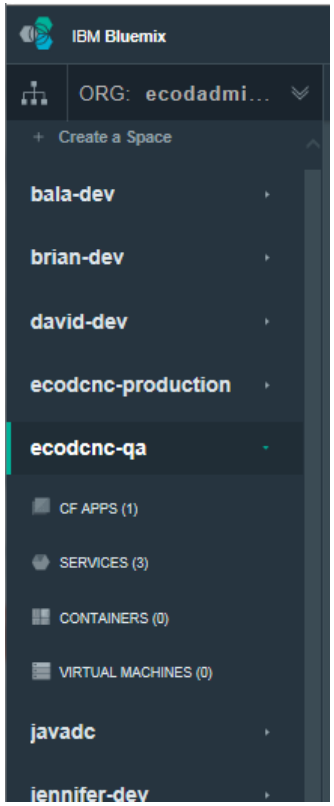
Organization - The grouping methodology for users in Bluemix – 2d



- Is defined by domains, spaces, users and quota. Organization name must be unique in Bluemix
- Users in an org share memory and service instance quotas
- Administrator can use *Manage Organizations* page on Bluemix dashboard to view ands manage the settings of the organization
- Can be used to collaborate between users and to facilitate the logical grouping of project resources
- When delete an organization, all the spaces, applications, and services within the organization are deleted.

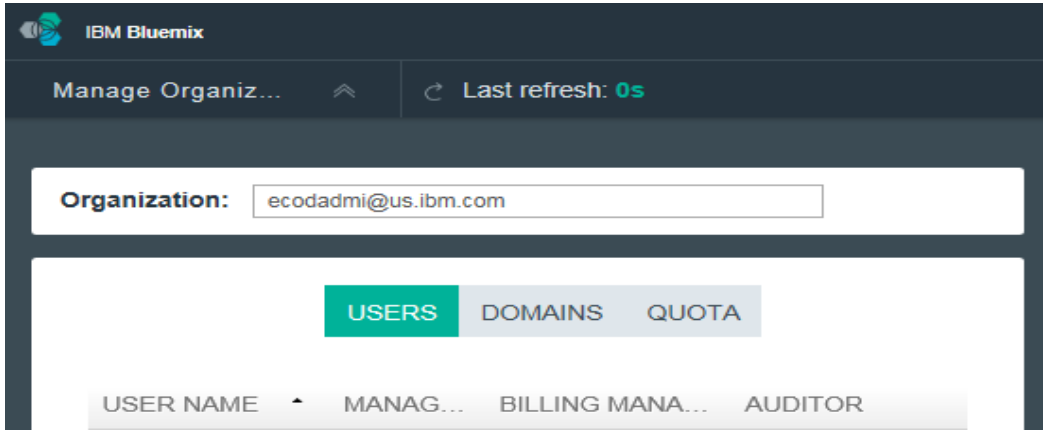
Space – An Organization unit and A sub-group within Bluemix Org – 2c

- Contains applications and services in the Cloud Foundry infrastructure and can be used to store and track application resources
- An organization can contain multiple spaces. In Bluemix, you must associate all the applications and services that you create with a space.
- Users who are members of an org are given access to one or more of its spaces, with permissions associated with a particular role (such as developer, manager, or auditor).
- Any member of the space can view apps, but only members with the developer role can create apps and add service instances to the space.
- Applications and service instances are associated with spaces



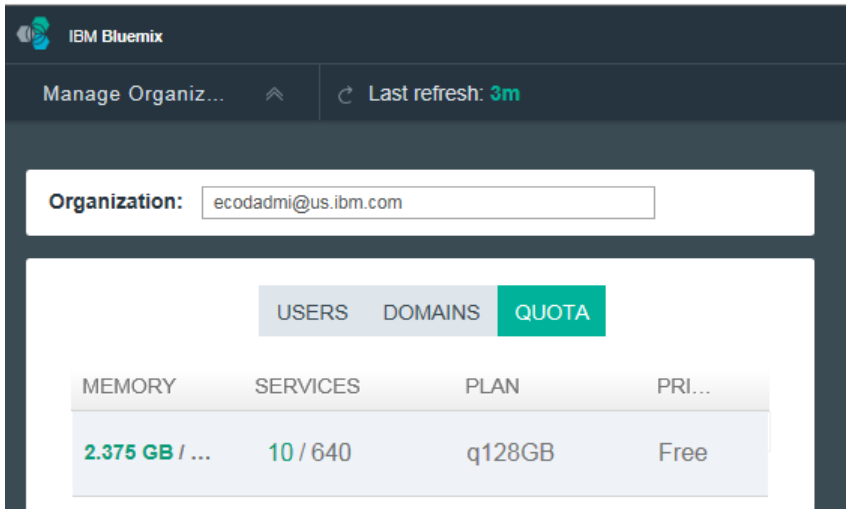
Bluemix User – The basic role in organizations and spaces – 2c

- Bluemix user must be assigned to an organization before the user can be granted permissions to the spaces within the organization.
- User can switch organizations and view resources of only one organization at a time
- An user can be either a member or a collaborator of an organization:
  - You are a collaborator of an organization, if you already have a Bluemix account, and someone else invites you to the organization.
  - You are a member of an organization if you don't have a Bluemix account, but then someone invites you to the organization and you sign up for Bluemix from the invitation.



Bluemix Quota – The organization resource limits - 2c

- Represents the resource limits for the organization and defines the number of services and the amount of memory that can be allocated for use by the organization.
- Any applications or services in a space of the organization contributes to the usage of the quota.
- The quota is assigned when organizations are created.

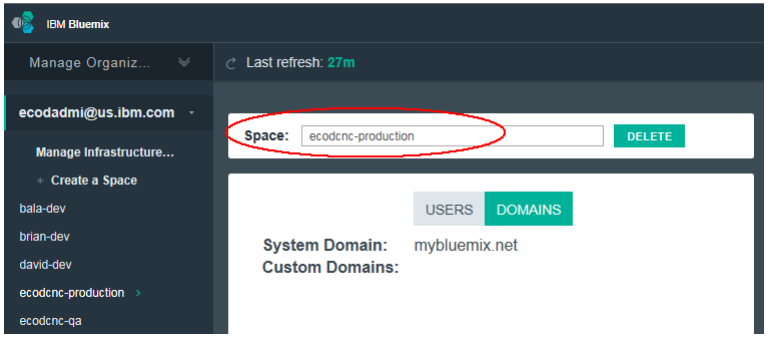


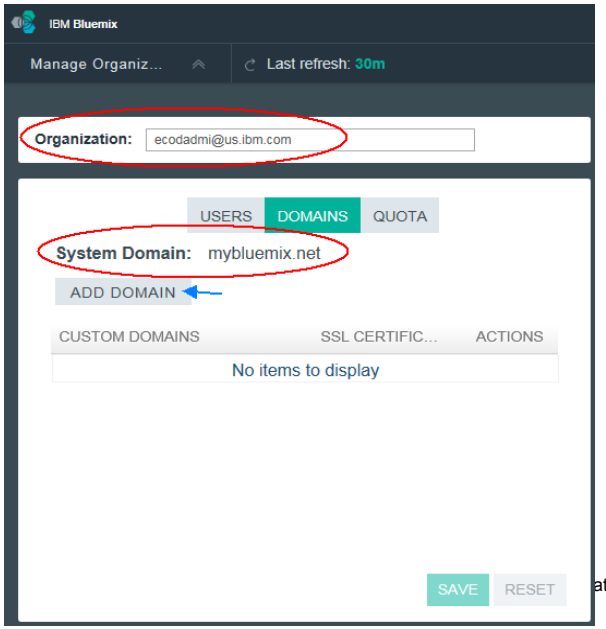
Bluemix Roles and Permissions – 2c

- Organization Roles
  - **Organization users** can view the settings of the organization
  - **Organization managers** can create spaces and add users to the organization or spaces.
  - **Billing managers** can view organization charges.
  - **Auditors** can view all organization and space content.
- Space Roles
  - **Space managers** can add users to the space.
  - **Developers** can add and configure applications and services.
  - **Auditors** can only view content in the space.

Bluemix Domains – 2c

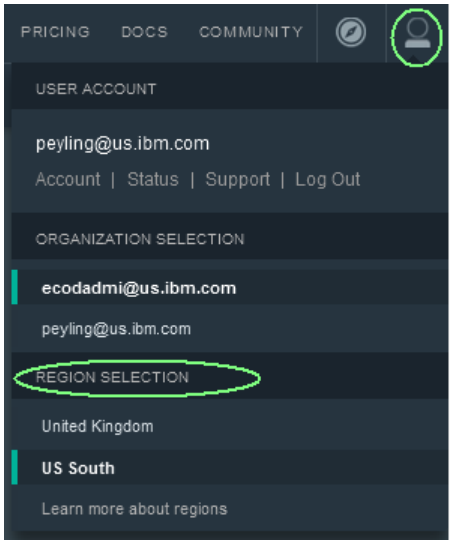
- Provide the route on the internet that is allocated to the organization. A route has a sub-domain and a domain. A sub-domain is typically the application name. A domain might be a system domain, or a custom domain that you registered for your application.
- Part of a naming hierarchy that specifies the route. For example, example.com.
- In Bluemix, domains are associated with orgs. Domain objects are not directly bound to apps





Bluemix Region– A Defined Geographical Territory for App Deployment – 2e

- A unique prefix, CF API endpoint & UI console are assigned to each region
- Applications and service instances can be created in different regions with same Bluemix infrastructure
- You can switch to a different region to work with the spaces in that region from Bluemix Dashboard UI Console after log in to Bluemix
- If using the cf command line interface, you must connect to the Bluemix region that you want to work with by using the cf API command and specifying the CF API endpoint of the region. For example,
  - $ cf api https://api.eu-gb.bluemix.net



Bluemix region list			
Region name	Region prefix	CF api endpoint	UI console
US South region	us-south	api.ng.bluemix.net	console.ng.bluemix.net
Europe United Kingdom region	eu-gb	api.eu-gb.bluemix.net	console.eu-gb.bluemix.net

Login to Bluemix Cloud Foundry Using Command Line Interface – 2f

- **Usage**
  - cf login [-a API_URL] [-u USERNAME] [-p PASSWORD] [-o ORG] [-s SPACE]
- **Arguments:**
  - API endpoint: This is the URL of the Cloud Controller in your Cloud Foundry instance.
  - Username: Your username, such as your Bluemix ID
  - Password: Your password.
  - Org: The organization where you want to deploy your application.
  - Space: The space in the organization where you want to deploy your application.
- **Example** : cf login –a https://api.ng.bluemix.net –u developer@us.ibm.com -p XXXX –o developer@us.ibm.com -s dev

Push an application to Bluemix Cloud Foundry – 2f

- **Usage** :
  - cf push APP [-b URL] [-c COMMAND] [-d DOMAIN] [-i NUM_INSTANCES] [-m MEMORY] [-n HOST] [-p PATH] [-s STACK] [--no-hostname] [--no-route] [--no-start]
- **Arguments:**

  -b — Custom buildpack URL, for example, https://github.com/heroku/heroku-buildpack-play.git or https://github.com/heroku/heroku-buildpack-play.git#stable to select stable branch
  -c — Start command for the application.
  -d — Domain, for example, example.com.
  -f — replaces --manifest
  -i — Number of instances of the application to run.
  -m — Memory limit, for example, 256, 1G, 1024M, and so on.
  -n — Hostname, for example, my-subdomain.
  -p — Path to application directory or archive.
  -s — Stack to use.
  -t — Timeout to start in seconds, give your application more time to start, up to 180 secs.
  --no-hostname — Map the root domain to this application (NEW).
  --no-manifest — Ignore manifests if they exist.
  --no-route — Do not map a route to this application (NEW).
  --no-start — Do not start the application after pushing.
- **Example: cf push my-app -c “node my-app.js”**

Scale your Application by using Cloud Foundry CLI – 2f

- Scaling an application - Adjustments of resources required by running the application to improve overall performance
- Cloud Foundry provides **cf scale** command to scale your application up or down horizontally and vertically to meet demand of workload
- Scaling horizontally
  - **cf scale APP -i INSTANCES** to horizontally scale your application
  - Cloud Foundry will increase or decrease the number of instances of your application to match INSTANCES
  - Example: **cf scale myApp –i 5**
- Scale vertically
  - Vertically scaling an application changes the disk space limit or memory limit that Cloud Foundry applies to all instances of the application
  - Change disk space limit: cf scale App –k DISK
  - DISK must be an integer followed by either an **M**, for megabytes, or **G**, for gigabytes.
  - Example: **cf scale myApp -k 512M** or **cf scale myApp -k 1G**

Application Loggregator – 2f

- user application logging subsystem of Cloud Foundry
- Loggregator allows users to:
  - Tail their application logs
  - Dump a recent set of application logs (where recent is a configurable number of log packets).
  - Continually drain their application logs to 3rd party log archive and analysis services.
  - (Operators and administrators only) Access the firehose, which includes the combined stream of logs from all apps, plus metrics data from CF components.
- **Usage** : **cf logs APP_NAME [--recent]**

Manage Organizations Using Command Line Interface – 2f

- To view your organization information such as domains, quota, spaces:
  - **cf org *ORGNAME***

```
c:\>cf t

API endpoint:   http://api.ng.bluemix.net (API version: 2.27.0)
User:           peyling@us.ibm.com
Org:            ecodadmi@us.ibm.com
Space:          ecodcnc-qa

c:\>cf org ecodadmi@us.ibm.com
Getting info for org ecodadmi@us.ibm.com as peyling@us.ibm.com...
OK

ecodadmi@us.ibm.com:
                     domains:        ng.bluemix.net, mybluemix.net
                     quota:          q128GB (131072M memory limit, Unlimited i
, 2000 routes, 640 services, paid services allowed)
                     spaces:         peyling-dev, ecodcnc-qa
                     space quotas:

c:\>
```

Manage Spaces Using Command Line Interface – 2f

- To check what organization, and space you are logged onto or to change the space:
  - **cf target**
  - **cf t -s test**
    - The first option will print the current target organization and space
    - The second option will switch to the test space

```
c:\>cf t

API endpoint:   http://api.ng.bluemix.net (API version: 2.27.0)
User:           peyling@us.ibm.com
Org:            ecodadmi@us.ibm.com
Space:          ecodcnc-qa

c:\>cf t -s peyling-dev

API endpoint:   http://api.ng.bluemix.net (API version: 2.27.0)
User:           peyling@us.ibm.com
Org:            ecodadmi@us.ibm.com
Space:          peyling-dev

c:\>cf t

API endpoint:   http://api.ng.bluemix.net (API version: 2.27.0)
User:           peyling@us.ibm.com
Org:            ecodadmi@us.ibm.com
Space:          peyling-dev

c:\>
```

Manage Users and Roles of an Organization using CLI – 2f

- Commands for listing users:
  - **cf org-users *ORG*** — List users in the organization by role.

```
c:\>cf org-users ecodadmi@us.ibm.com
Getting users in org ecodadmi@us.ibm.com as peyling@us.ibm.com...

ORG MANAGER



BILLING MANAGER



ORG AUDITOR
```

- Commands for managing roles (admin-only) in your organization:
  - **cf set-org-role USERNAME ORG ROLE** — Assign an organization role to a user. The available roles are “OrgManager”, “BillingManager”, and “OrgAuditor”.
  - **cf unset-org-role USERNAME ORG ROLE** — Remove an organization role from a user.

Manage Users and Roles of a Space using Command Line Interface -2f

- Commands for listing users:
  - **cf space-users ORG SPACE**— List users in the space by role.

```
c:\>cf space-users ecodadmi@us.ibm.com ecodcnc-qa
Getting users in org ecodadmi@us.ibm.com / space ecodcnc-qa as peyling@us.ibm.com

SPACE MANAGER



SPACE DEVELOPER









SPACE AUDITOR


c:\>
```

- Commands for managing roles (admin-only) in your organization:
  - **cf set-space-role** — Assign a space role to a user. The available roles are “SpaceManager”, “SpaceDeveloper”, and “SpaceAuditor”.
  - **cf unset-space-role** — Remove a space role from a user.

Manage Domains Using Command Line Interface – 2f

- All domains are mapped to an Organization
- Domains can be **shared** or **private**. Shared domains are registered to multiple orgs while private domains, or **owned** domains, are registered to one org. A Cloud Foundry instance defines a default shared domain that your application uses unless you specify a different domain.
- To list domains in the target organization - **cf domains**

```
$ cf domains
Getting domains in org my-org as user@example.org... OK

name            status
example.com     shared
example.org     owned
```

- Commands for managing domains:
  - **cf create-domain** — Create a domain.
  - **cf delete-domain** — Delete a domain.
  - **cf create-shared-domain** — Share a domain with all organizations. Admin only.
  - **cf delete-shared-domain** — Delete a domain that was shared with all organizations. Admin only.

Section 3: Implementing Cloud Ready Applications

a. Understand how to design, develop, deploy and manage a IBM Bluemix PaaS application following the Twelve-Factor App methodology (http://12factor.net/)

b. Understand scaling concepts for a Cloud application and steps to scale an application in IBM Bluemix PaaS

c. Debug a Cloud application using development mode of IBM Bluemix PaaS

d. Perform load testing on Cloud applications using simulated loads and describe the benefits of load testing

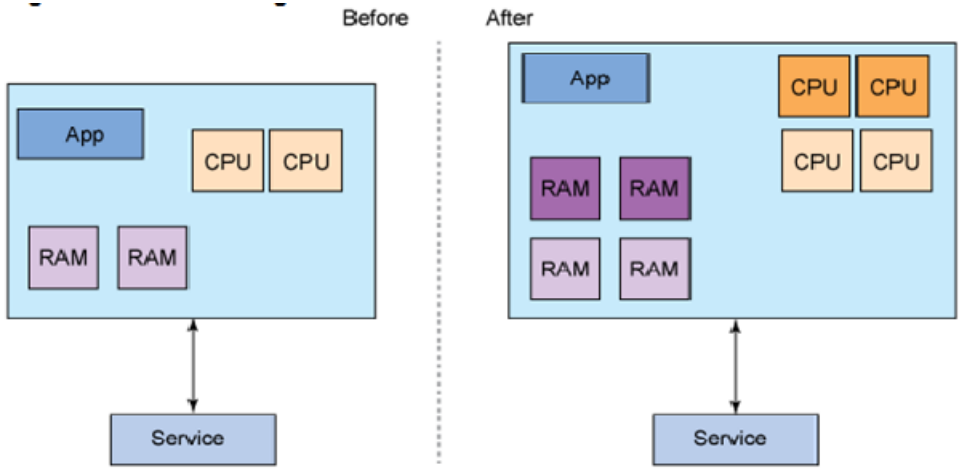
e. Explain various methods to monitor an application in IBM Bluemix PaaS

What are the Twelve Factors? – 3a

1. Codebase - One codebase tracked in revision control, many deploys
   - 1-1 relationship between app & code repo – use packages for shared code
2. Dependencies – Declared and isolated (no system wide dependencies)
3. Config - Store config in the environment (not in constants in the app)
4. Backing Services - Treat backing services as attached resources
   - Can be attached and reattached w/o affecting code
5. Build, release, run - Strictly separate build and run stages
   - Release has unique id
6. Processes - Execute the app as one or more stateless processes
   - State shared via external services – no sticky sessions !
7. Port binding - Export services via port binding
   - In deployment, a routing layer handles routing requests from a public-facing hostname
8. Concurrency - Scale out via the process model
9. Disposability - Maximize robustness with fast startup and graceful shutdown
10. Dev/prod parity - Keep development, staging, and production as similar as possible
11. Logs - Treat logs as event streams
12. Admin processes - Run admin/management tasks as one-off processes

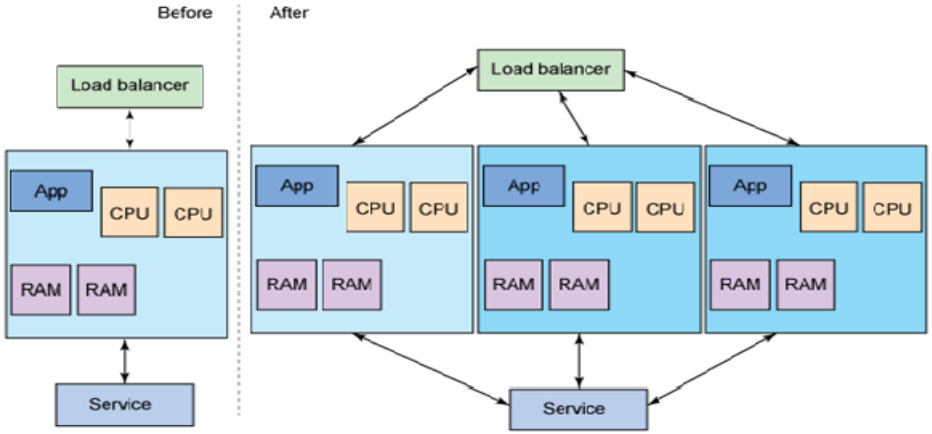
Vertical Scaling – 3b

Vertical scaling is often referred to as *scaling up*. Vertical scaling increases the resources available to an application by adding capacity directly to the individual nodes — for example, adding additional memory or increasing the number of CPU cores. Picture below shows the concept of vertical scaling with the addition of both memory and CPU to an application.



Horizontal scaling – 3b

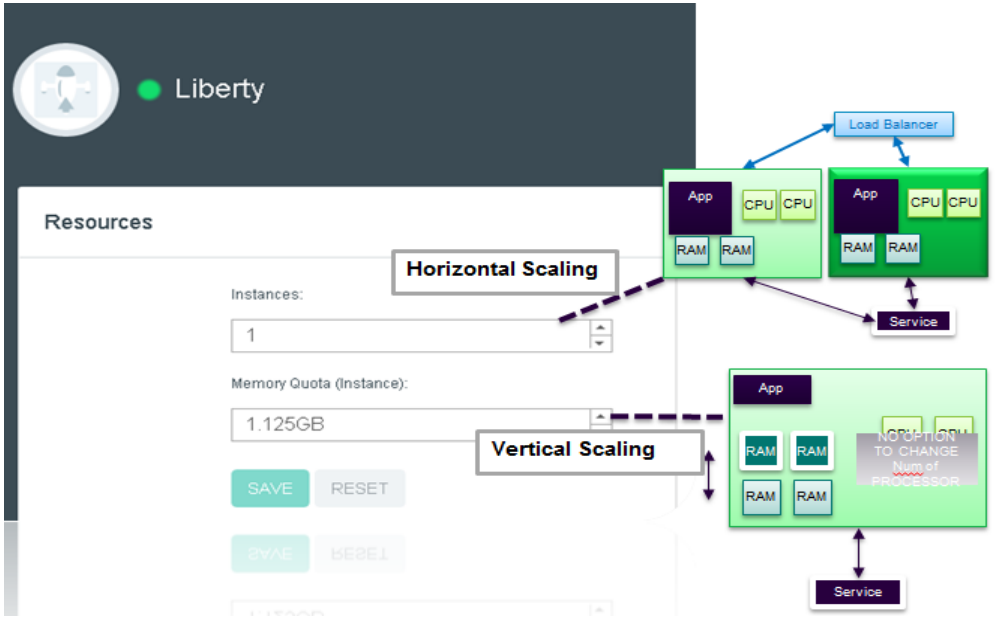
Horizontal scaling is often referred to as *scaling out*. The overall application resource capacity grows through the addition of entire nodes. Each additional node adds equivalent capacity, such as the same amount of memory and the same CPU. Horizontal scaling typically is achievable without downtime. Picture below illustrates the concept of horizontal scaling; you see additional identical nodes added with a load balancer in front of the application nodes.



3

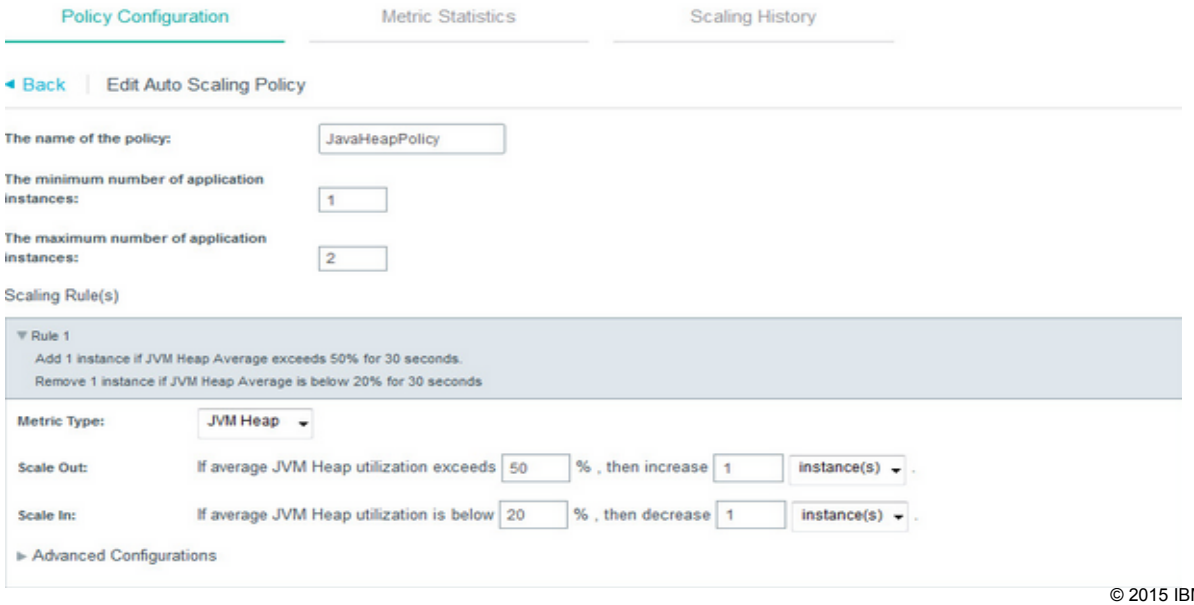
Steps to Manually Scale in IBM Bluemix PaaS – 3b

The IBM Bluemix UI Dashboard and command line support both vertical and horizontal scaling through increasing the amount of memory and increasing the number of instances of an application runtime. Both techniques can be applied to the same application.



Auto-Scale in IBM Bluemix – 3b

- **Auto-Scale in Bluemix - Policy Configuration**
  - Where you can create or edit the scaling policy that contains instructions on how to trigger the scaling activities. For Liberty for Java™ applications, you can define scaling rules for Throughput, JVM Heap, Memory, and Throughput.
  - For Node.js applications, you can define scaling rules for Throughput and Memory.
  - For Ruby applications, you can define scaling rules for Memory.
  - **Breach duration** = time a threshold may be exceeded w/o triggering an auto scale action
  - **Cooldown period** = time to wait after an autoscale action before triggering more

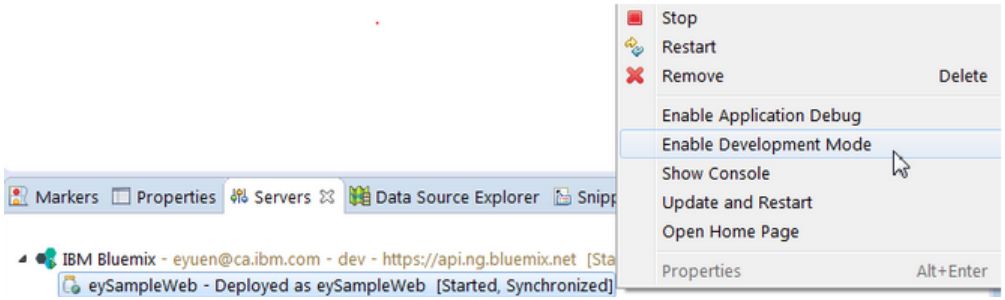


IBM Eclipse Tools for Bluemix – 3c

- Install IBM Eclipse Tools for Bluemix from Eclipse Marketplace

Eclipse - Incremental Publish of Applications in Bluemix – 3c

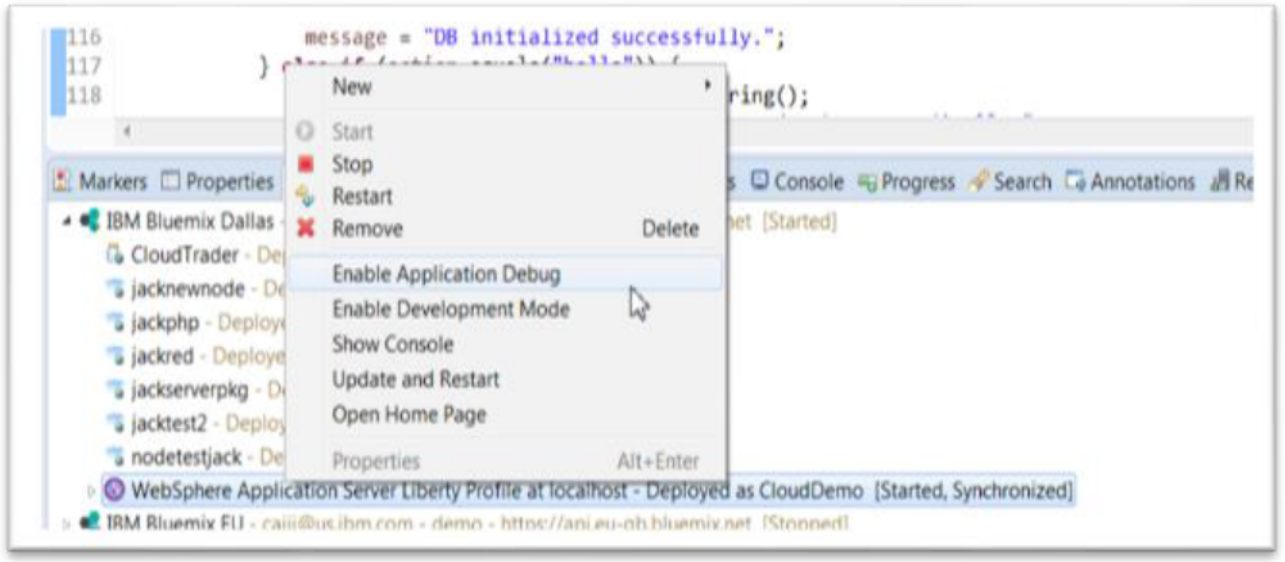
- Enable Development Mode



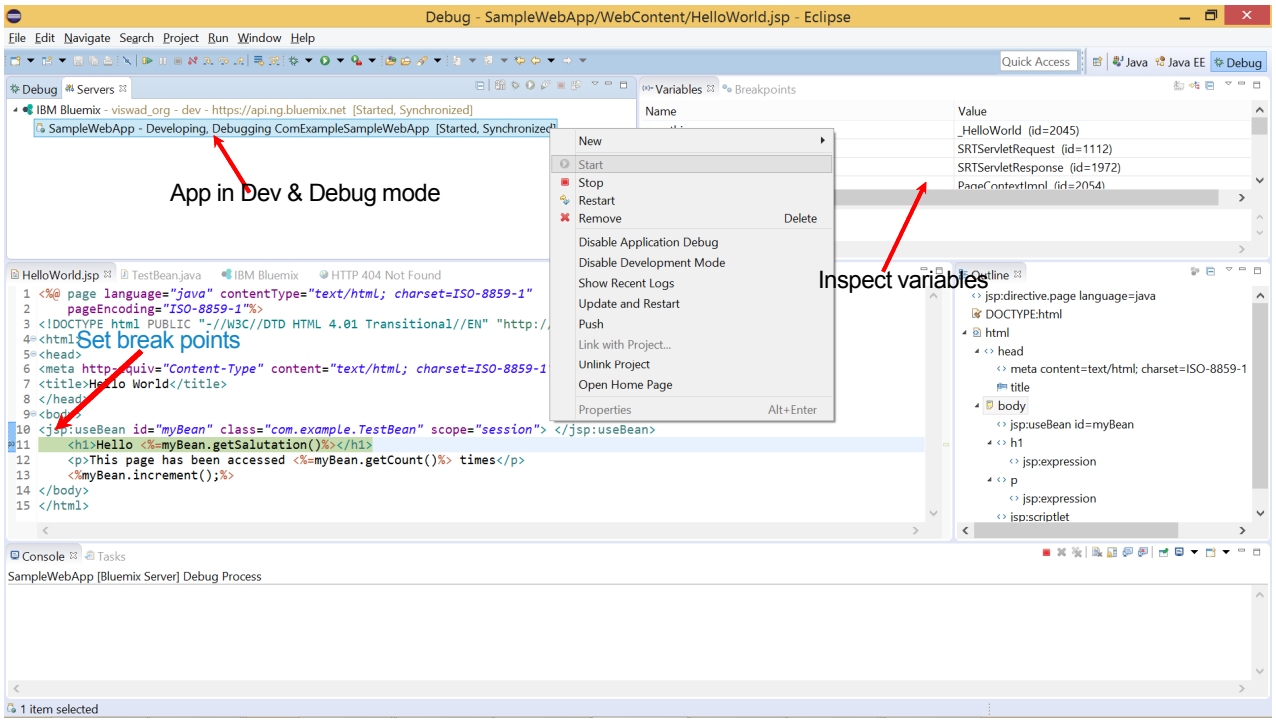
- Enabling Development Mode provides access to
  - Push incremental file updates
  - Run additional tools inside app container like SSH web console

Eclipse - Enable Debug Mode – 3c

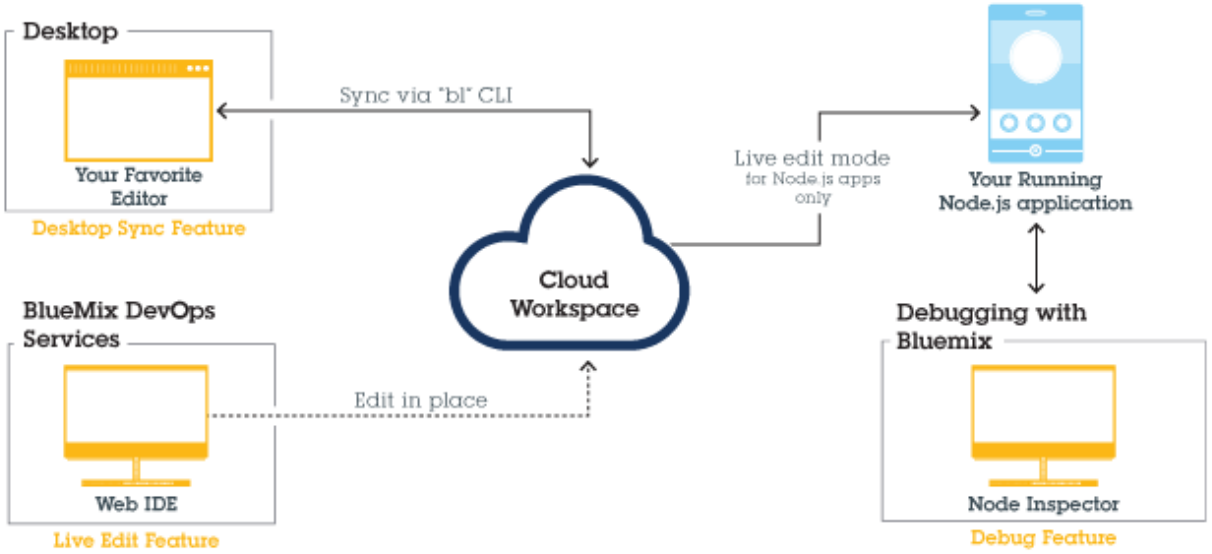
- Automatically enables development Mode if not yet enabled



Eclipse - Remotely Debug Applications in Bluemix – 3c

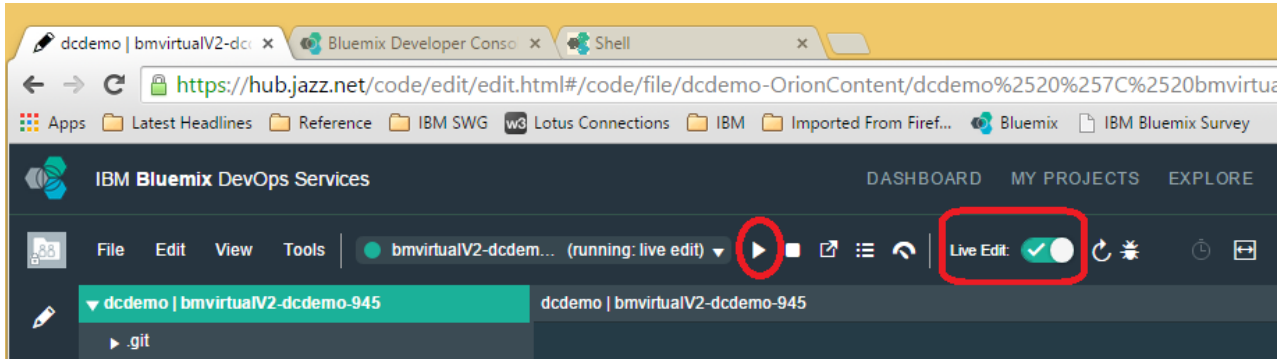


Bluemix Live Sync - Node.js Applications – 3c

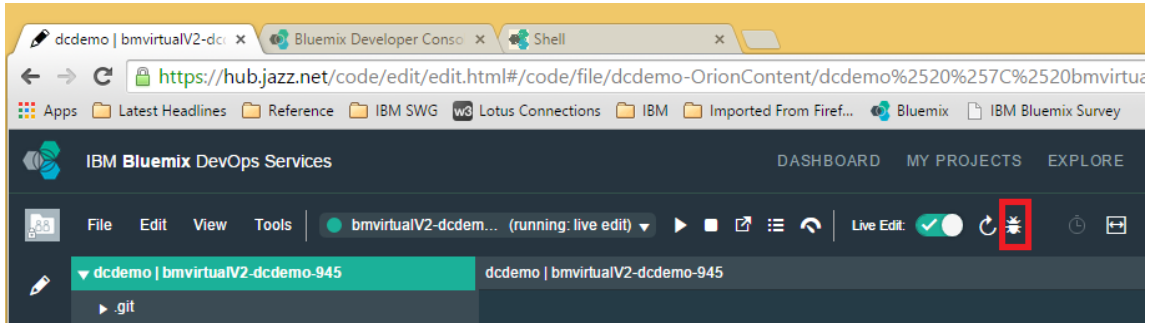


Dev Ops Services - Node.js Applications – Live Edit, Debug - 3c

- Live Edit - Click on Live Edit and redeploy



- Debug- Click on debug icon



Dev Ops Services - Node.js Debugger - 3c

IBM **Bluemix** Debugger

You are now managing the runtime (Node.js process) of your app. To manage your project, visit your app dashboard.

Sentiment Analysis App Speed Lab

Route: SimpleSentimentAnalysis.mybluemix.net

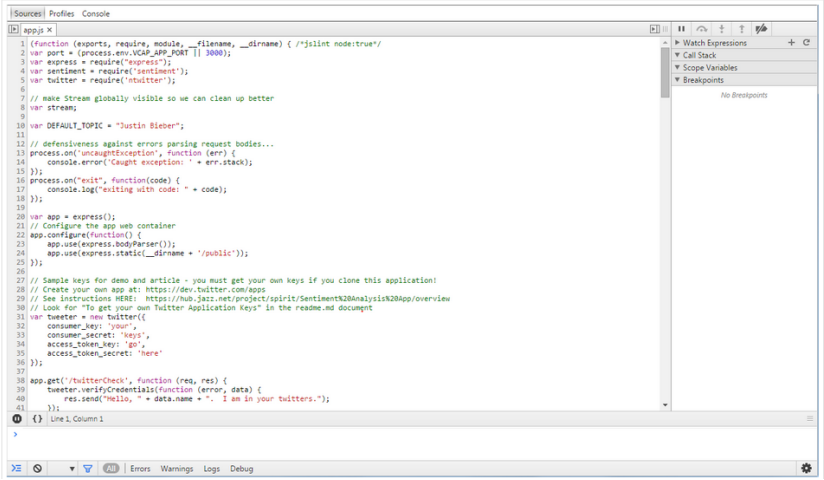
Your app is running.

SUSPEND

RESTART

Open Shell

Open Debugger



What is Load Testing? – 3d

Load testing is the simplest form of performance testing it helps determine the capability of handling a certain amount of users (load) on your cloud application. The goal is to see how your system will perform when subjected to both an expected and stressful amount of load - the latter of which is called stress testing.

Benefits of Load Testing

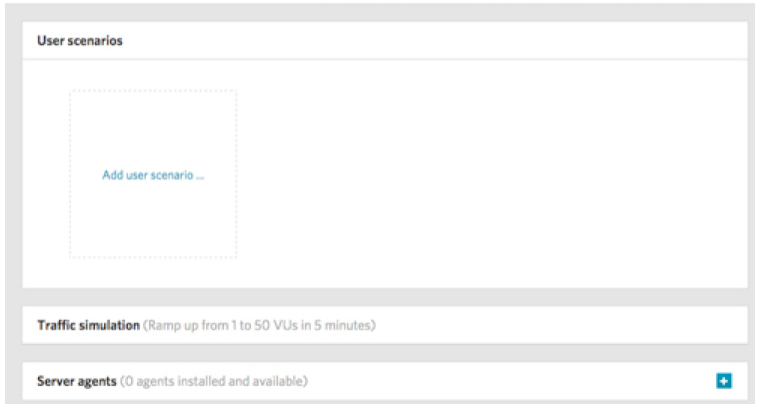
- No setup or maintenance required
- Easy and fast deployment
- Scalability
- Easy management
- Tests can be done from different geographical regions
- Low maintenance cost

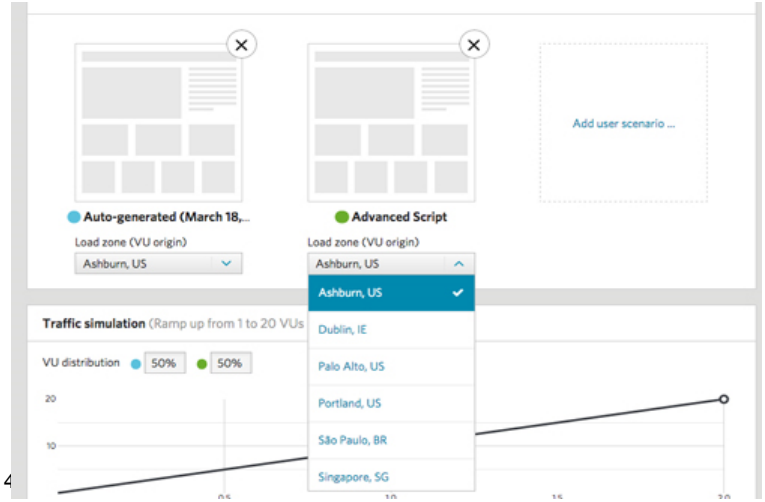
Load Testing Tools in IBM Bluemix PaaS

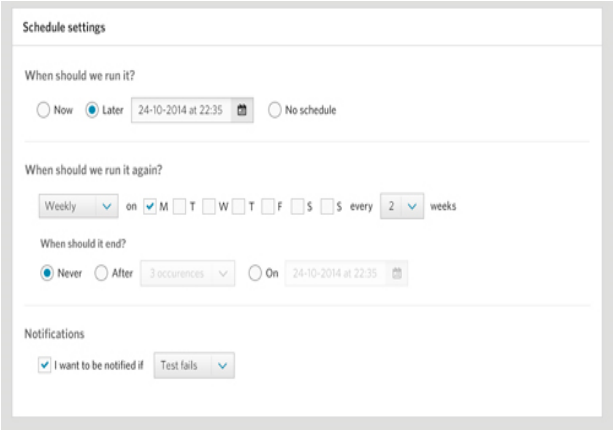
- BlazeMeter
- Load Impact

Load Test Execution – 3d

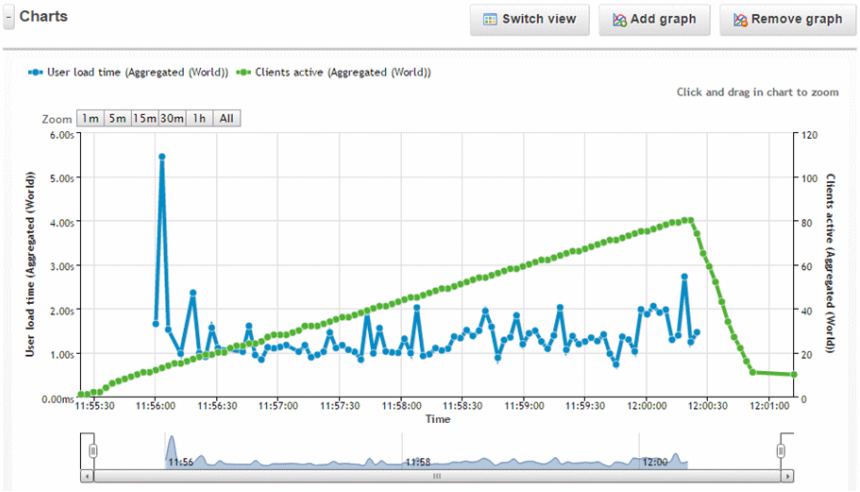
- Load Test
  - Create User Scenario(s)
  - Create Test(s)
  - Define the Load
  - Execute the Multiple Iterations







Analyze and Interpret the Results in Load Impact – 3d



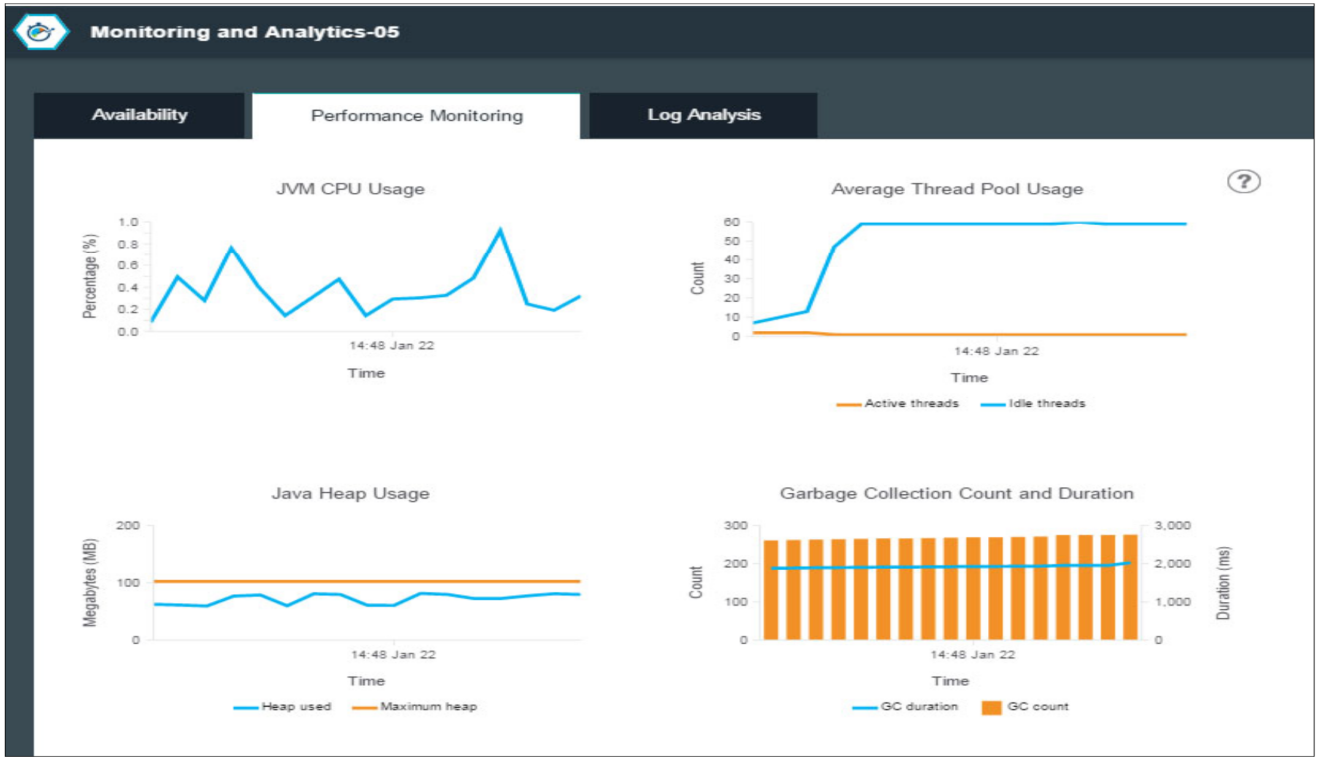
- Green dots show the virtual user (VU) load, and
- Blue dots the duration of time for a user scenario to complete.
- In the example shown, there is no strong correlation between the response time and the number of active users.
  - No reached peak capacity
- When reviewing results, it is critical to verify that all application responses are successful and not showing error codes.

Options to Monitor Applications in Bluemix – 3e

Different Options for Monitoring the IBM Bluemix Application

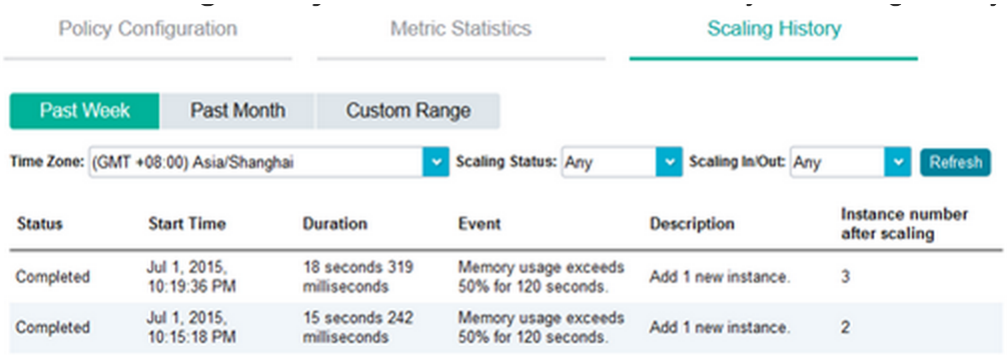
- Option 1: Monitoring IBM Bluemix Application Logs before and after it has started using cf logs command in the Cloud Foundry CLI
  - Covered in Section 2
- Option 2: Viewing Performance Metrics using IBM Bluemix Monitoring & Analytics Service
- Option 3: Viewing Performance Metrics using IBM Bluemix Autoscaling Service – if the Application uses that Service
- Option 4: Viewing Performance Metrics using IBM Bluemix Autoscaling Service – if the Application uses that Service

Option 3 - Viewing Performance Metrics using IBM Bluemix Monitoring Service – 3e



Option 4: Viewing Performance Metrics using IBM Bluemix Autoscaling Service – 3e

- Scaling history indicates when scaling events occurred



Cloud Application Developer Certification Review Part 2 Section 4

Presented by:

IBM Cloud Ecosystem Development

IBM **Cloud**

Section 4: Enhancing Cloud Applications using Managed Services

a. Improve performance and scalabilty of IBM Bluemix PaaS applications with caching

b. Understand how to configure external authentication using IBM Bluemix PaaS web applications with the Single Sign On service (SSO)

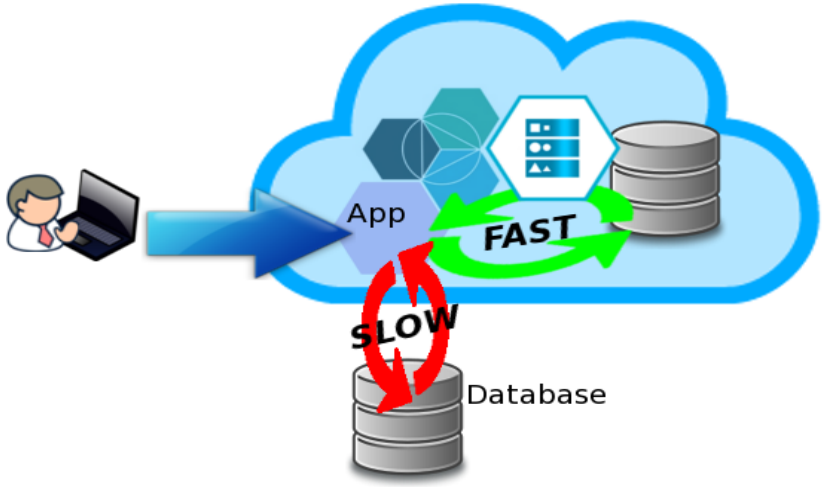
c. Enable loosely coupled integration for IBM Bluemix PaaS applications and components by using MQ Light service

d. Describe cognitive capabilities to process unstructured data and images in IBM Bluemix PaaS

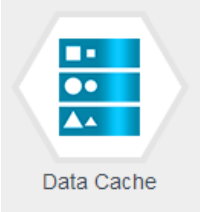
e. Understand how to store and retrieve files using the IBM Object Storage service in Bluemix

Bluemix Data Caching Service – 4a

- Distributed cache
- NoSQL style in-memory grid (IMDG)
- CRUD operations
- Linear scalability



IBM Bluemix DataCache Service – 4a



- A caching service for cloud resident applications.
- Supports distributed caching scenarios for web and mobile applications.
- Store key and value objects in memory for fast access.
- Provides linear scalability, predictable performance, and fault tolerance of the web application's data requirements.
- Replicates data so that cache components may be restarted without data loss or performance penalty.
- The developer requests the service and binds it to the application. Provisioning automatically and quickly. The caching service is an example of a composable element that can be used to build applications quickly.
- The service is continually maintained by the cloud environment.

Monitoring DataCache Service – 4a

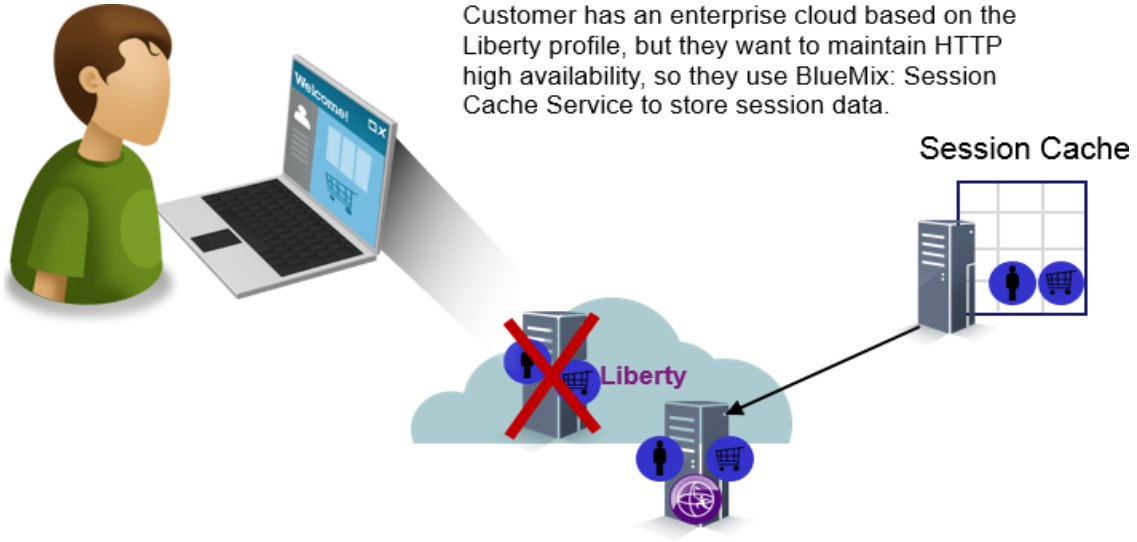
- Space Used
  - Cache Contents
- Throughput
- Transaction Time
- Hit Rate

Bluemix Session Caching Service – 4a

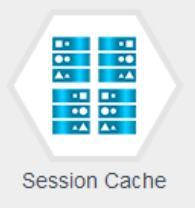
HTTP Session Failover in the Cloud

- **Distrubuted session cache**
- **Session failover**
- **Ease of Use**
- **Session replication**

Customer has an enterprise cloud based on the Liberty profile, but they want to maintain HTTP high availability, so they use BlueMix: Session Cache Service to store session data.



IBM Bluemix SessionCache Service – 4a

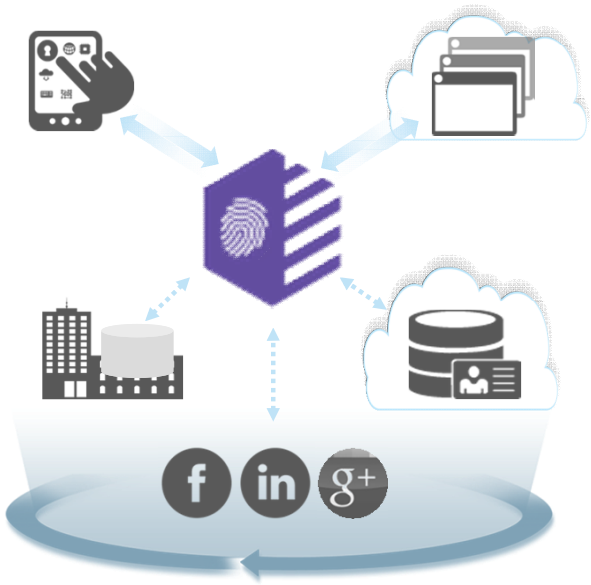


- Special purpose elastic in memory cache for storing HTTP session data.
- Stores and persists HTTP session objects to the data grid so that they do not have to be stored in the memory.
- No code change to applications using the J2EE standard HTTP session cache.
- Data for each HTTP session Survives on a server outage for an application
- Requires no developer effort to manage.
- Replicates the session data to avoid a single point of failure.
- Provides low latency data access, transactional semantics.
- Just as for the DataCache service, the developer requests the service and binds it to the application. Provisioning of the service happens automatically, within seconds.

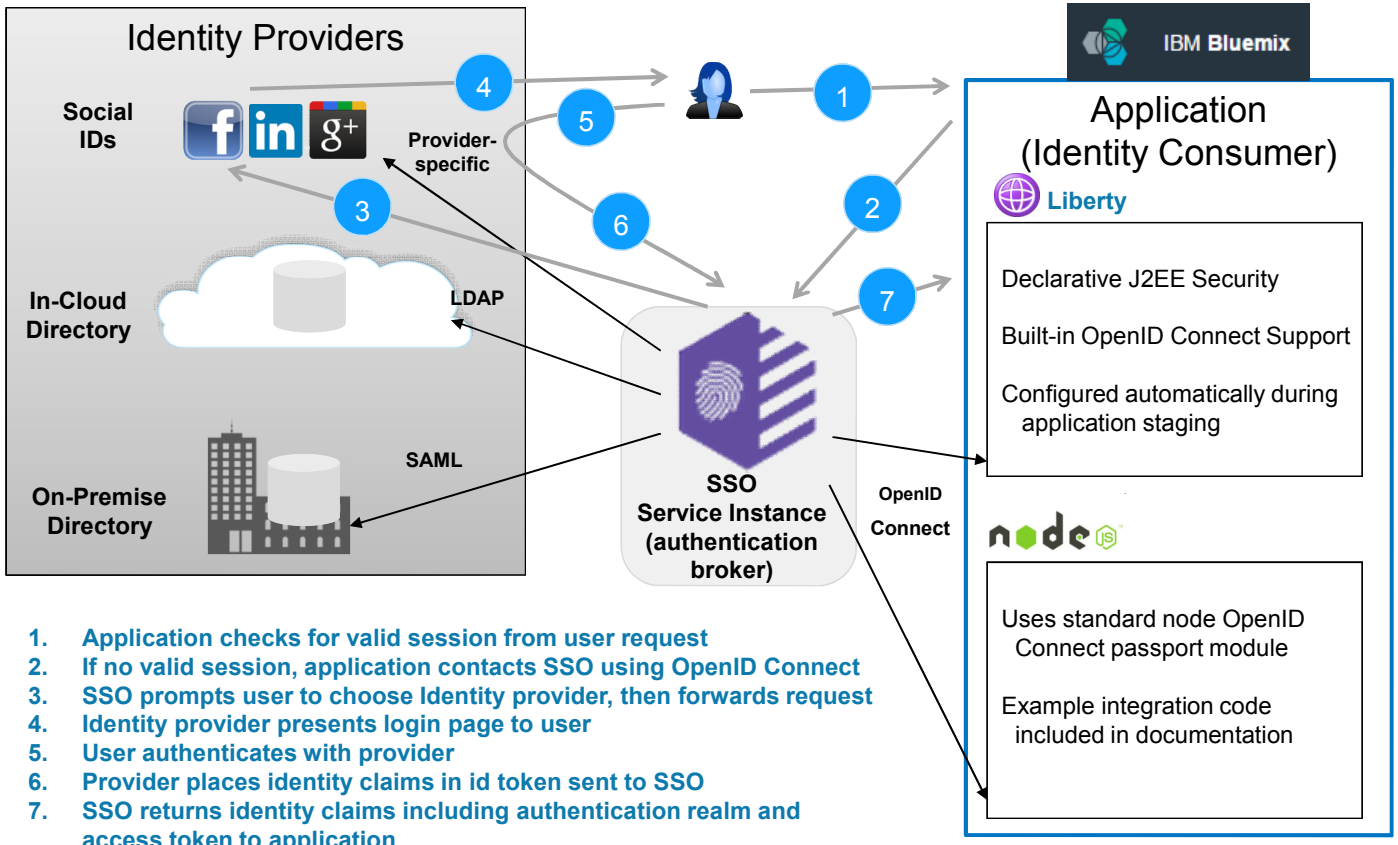
7

IBM Single Sign On service in Bluemix – 4b

- Configure and control external authentication for web applications
  - Apps use OpenID Connect client to service
    - Configuration examples for Java, Node.js
  - Apps don't require knowledge of interface to external identity source
  - SSO service control panel supports multiple identity sources
    - Social Media sites (Facebook, Google, LinkedIn)
    - Cloud Directory identity source
    - SAML Enterprise identity source
      - Identity Bridge appliance for LDAP
  - SSO service user login panel uses templates for customization



Authentication flow with IBM Single Sign On -4b

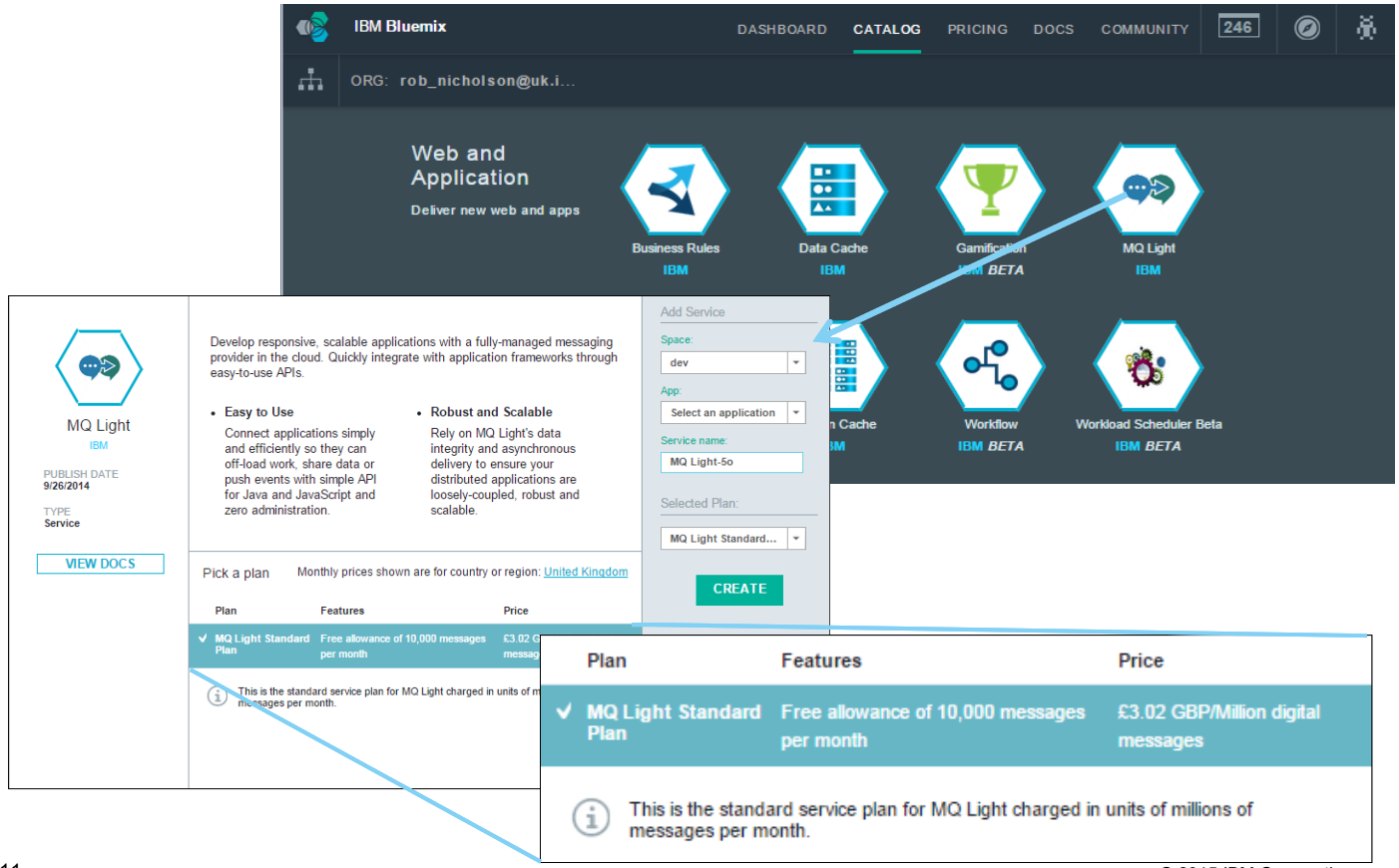


1. Application checks for valid session from user request
2. If no valid session, application contacts SSO using OpenID Connect
3. SSO prompts user to choose Identity provider, then forwards request
4. Identity provider presents login page to user
5. User authenticates with provider
6. Provider places identity claims in id token sent to SSO
7. SSO returns identity claims including authentication realm and access token to application

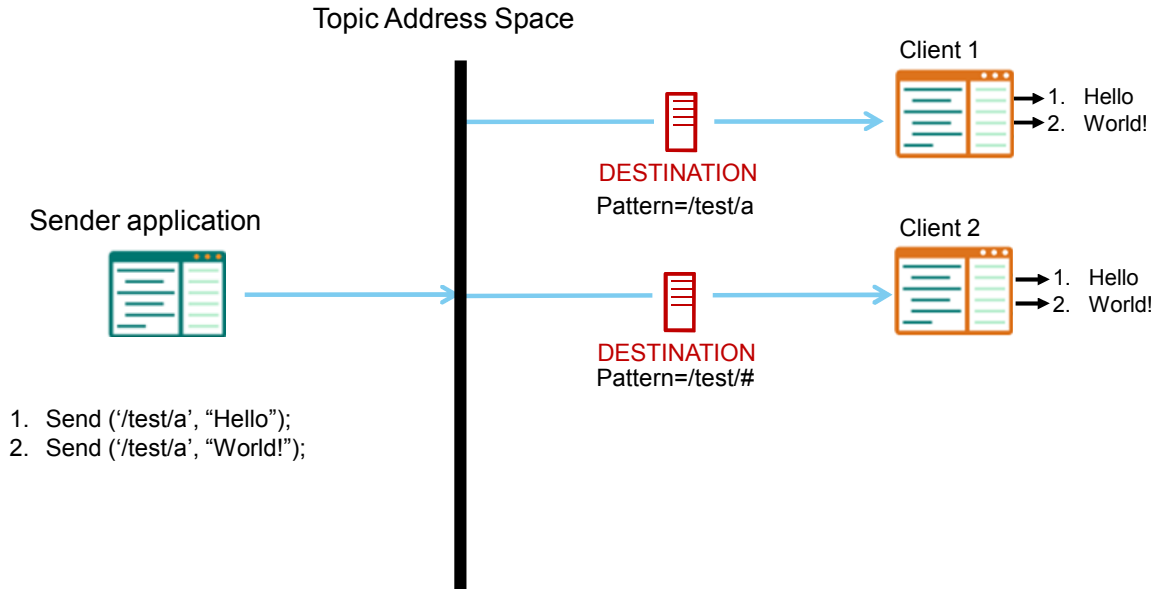
Configuring SSO High Level Steps -4b

1. Add the Single Sign On service to the dashboard
2. Select the identity source(s) to configure
3. Configure settings for identity source
4. Bind SSO service to application and access integrate tab to download Node.js module
5. Add integration code into application (implement a authentication callback)
   - Node.js and Java samples provided, others use an OpenID Connect client compatible ibrary
6. Provide callback URL and specify one or more configured identity sources using the service integrate tab
7. Redeploy application and access a protected URL

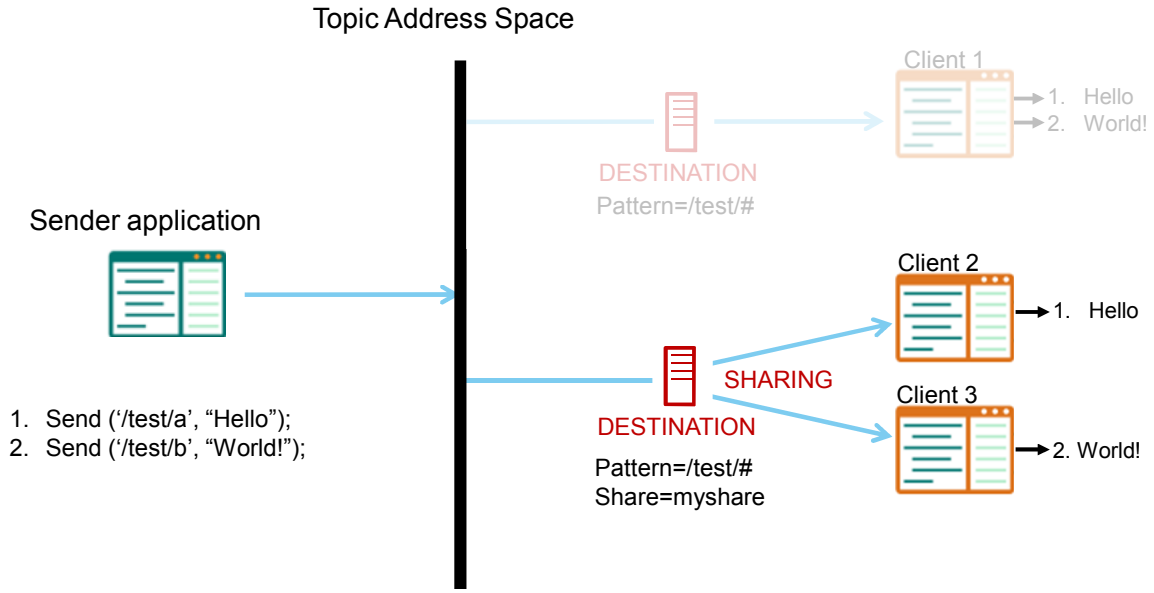
Introduction to MQ Light Service in Bluemix – 4c



MQ Light Messaging Model – Pub/Sub – 4c



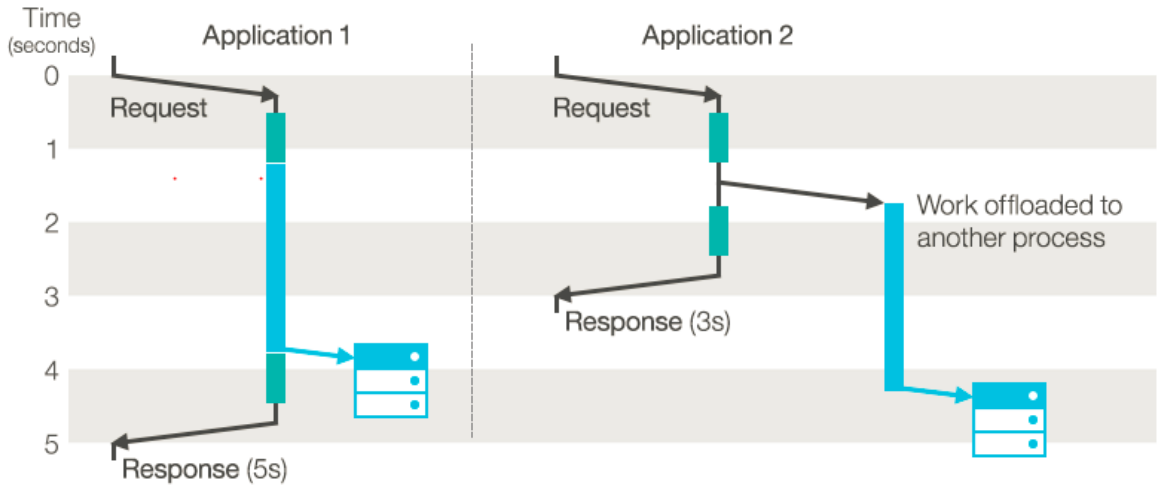
MQ Light Messaging Model – Sharing – 4c



13

Worker Offload Pattern using MQ Light – 4c

Improve the level of responsiveness by offloading work from the application's main event loop.



What is Object Storage ?

- Scalable, cost effective, file based storage available in the Cloud
  - Examples include;
    - IBM Soft Layer Object Storage
      - An implementation of OpenStack Swift
    - Amazon S3
    - Google Drive

- Different implementations share common characteristics
  - "Folder" like grouping of files
    - *Containers* in OpenStack Swift
    - *Buckets* in Amazon S3
    - *Folders* in Google Drive
  - REST API
    - Supports create, update, retrieve, delete and list operations on files and containers
  - Redundancy and failover
    - Multiple copies of files are maintained to minimize the possibility of data loss and to increase overall performance

OpenStack Swift Primer

- OpenStack Object Store
  - Distributed, Scale-out Object Storage
  - **CAP**
    - Eventually consistent
    - Highly Available – no single point of failure
    - Partition Tolerant
  - Well suited for unstructured data
- Uses container model for grouping objects with like characteristics
  - Objects are identified by their paths and have user-defined metadata associated with them
- Accessed via RESTful interface
  - GET, PUT, DELETE
- Built on standard hardware
  - Cost effective, efficient

SoftLayer Object Storage Services in Bluemix

- Object Storage Version 1
  - Powered by SoftLayer Object Storage
    - OpenStack Swift API
  - For Bluemix applications needing file storage via API access
    - Credentials generated upon binding to an app
  - Each instance has a unique username and password

- Object Storage Version 2
  - Powered by SoftLayer Object Storage
    - OpenStack Swift API
  - Uses OpenStack Keystone authentication
  - For access from within Bluemix apps and outside of Bluemix
    - Credentials generated upon instance creation
      - No need to bind instance to an app to access them
      - One account per Bluemix organization
  - Provides access to Object Storage that resides in your private cloud.

Swift API

- REST API to support management of objects and containers and their metadata
  - Upload, download, delete, update, list objects
  - Create, delete, list (contents of) containers
  - Create, retrieve, update metadata for containers and objects
    - Examples of metadata:
      - Access control info (read, write etc)
      - Object mime type
      - Custom fields
- See http://developer.openstack.org/api-ref-objectstorage-v1.html for more info

MQ Light – Message Types and Delivery Assurance -4c

- Messages
  - Payload is either Text or Binary.
  - Content-type is used by clients to transfer JSON
  - Per message time to live.

- Message delivery model
  - At most once delivery (QoS 0)
  - At least once delivery (QoS 1)
  - Acknowledge & Reject messages
  - Control over the number of unacknowledged messages delivered. (link credit)

Alchemy Language APIs – 4d

- Entity extraction - identify the proper nouns, i.e. people, companies, locations, etc.
- Sentiment analysis - determine the overall sentiment or sentiment specific to a keyword or entity.
- Keyword extraction - extract the important terms.
- Author extraction - identify the author of a blog post or news article.
- Language detection - detect 97+ languages.
- Text extraction - pull out only the import content from a web page.
- Feed detection - extract the ATOM or RSS feeds from a web page.
- **Concept tagging** - identify the overall concepts of the text.
- **Relation extraction** - extract subject-action-object relations.
- **Taxonomy Classification** - automatically categorize your text, HTML or web based content into a hierarchical taxonomy.
- **Microformats Parsing** - Automatically detect and parse the microformats embedded within a webpage.
- **Linked Data Support** - AlchemyAPI supports Linked Data in the web there by exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF

Alchemy Vision APIs – 4d

- **Image Link Extraction** - perform image link extraction on Internet-accessible URLs and posted HTML files.
- **Image Tagging -** perform image tagging on your Internet-accessible URLs and posted image files.
- **Face Recognition** - perform face detection and recognition on your Internet accessible URLs and posted image files.

Cloud Application Developer Certification Review Part 3 Sections 5 - 6

Presented by:

IBM Cloud Ecosystem Development

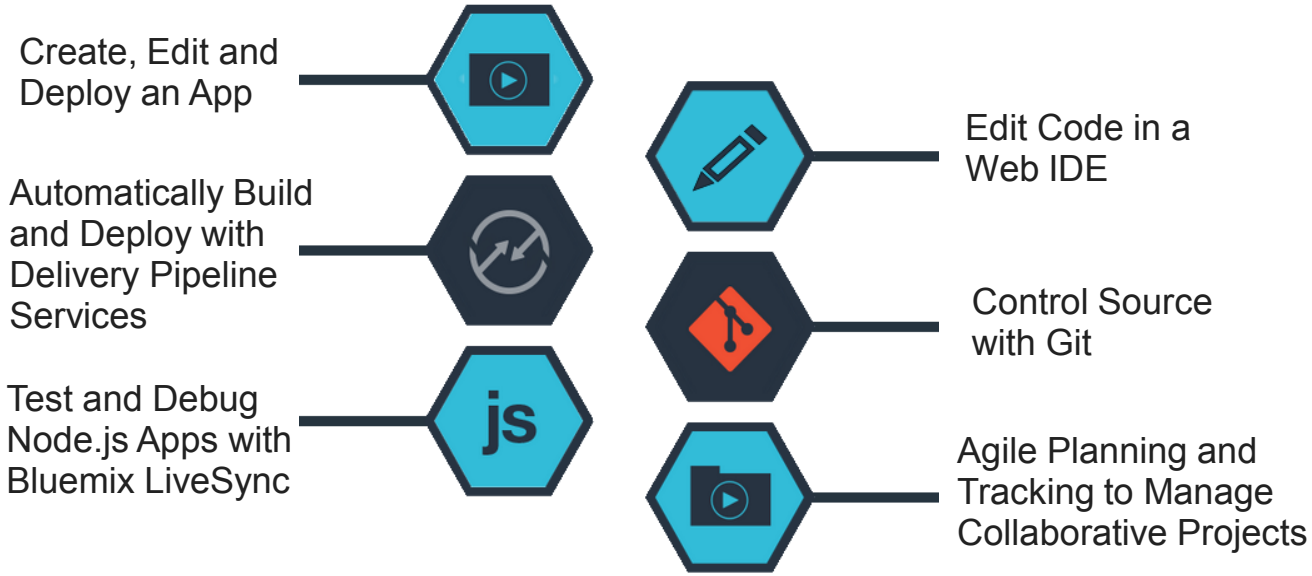
IBM **Cloud**

Section 5: Using DevOps Services & tools to Manage Cloud Applications

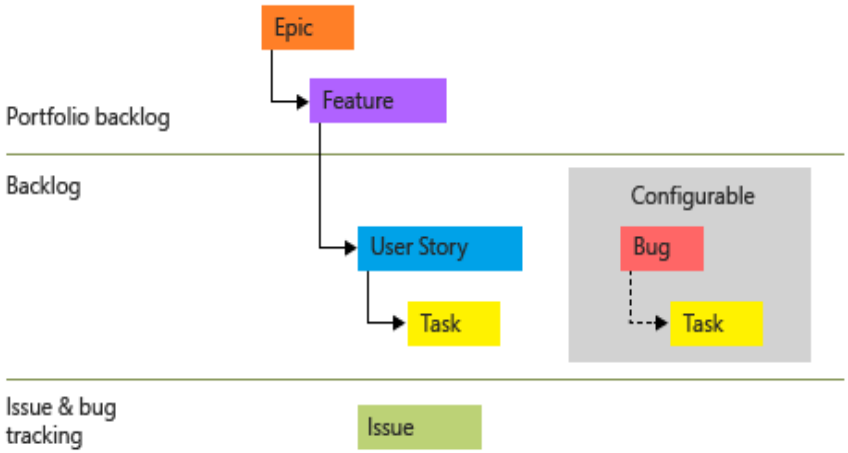
a. Describe capabilities of IBM Bluemix DevOps Services
b. Plan and track work for agile team collaboration
*c. Edit and debug Cloud applications using IBM Bluemix DevOps Services Web code editor**
*d. Understand capabilities of IBM Bluemix DevOps services source code management for projects**
*e. Describe how use the to Build & Deploy option to manage continuous integration and continuous delivery**

* Covered via lab exercise

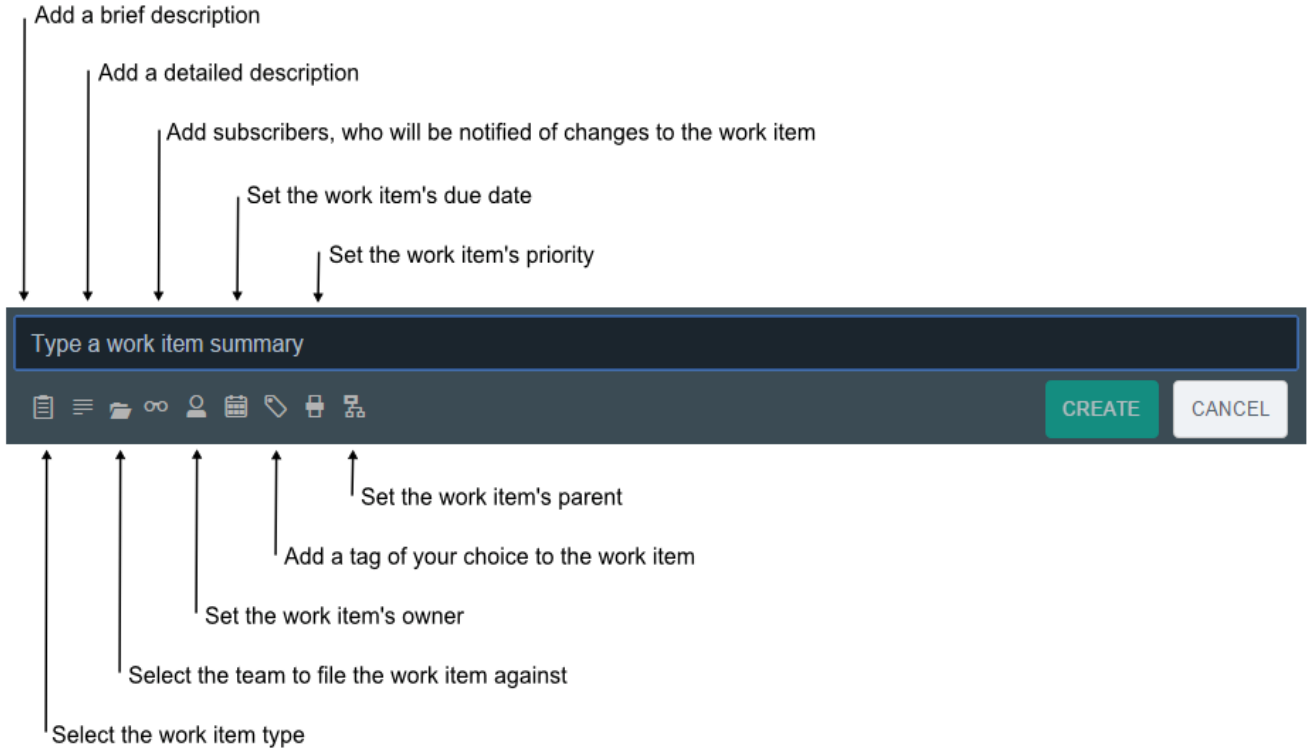
What is IBM Bluemix DevOps Services? – 5a



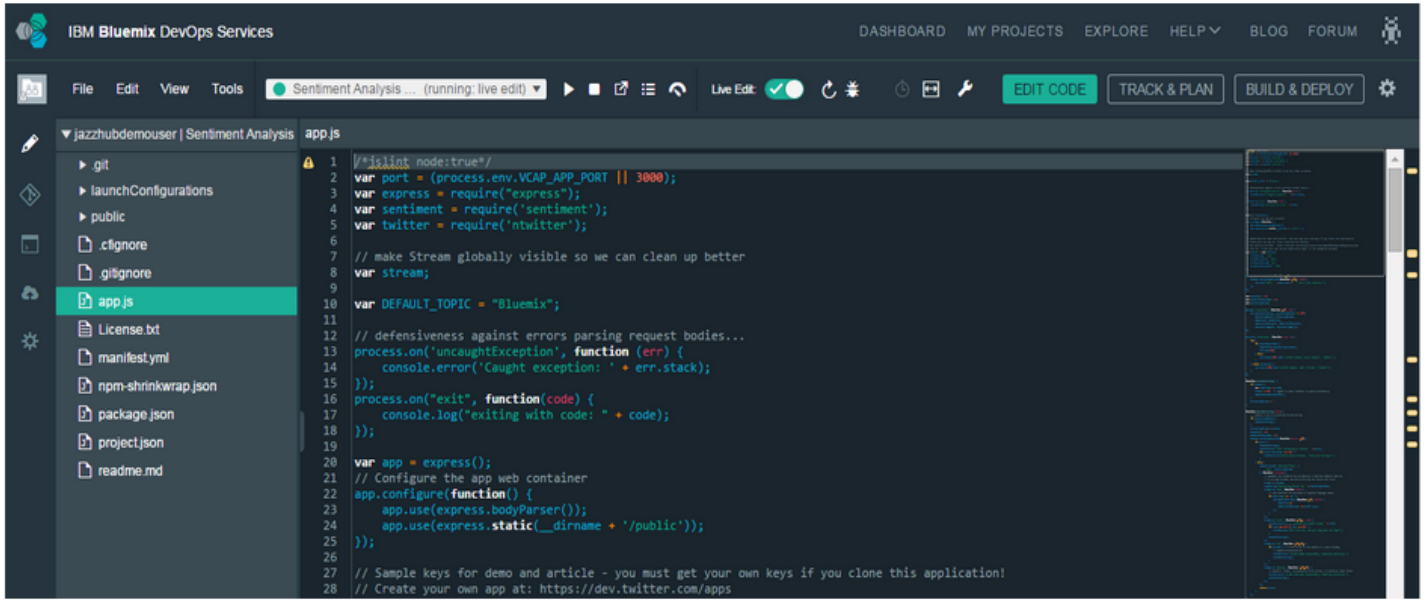
Agile Capabilites - Track and Plan – 5a



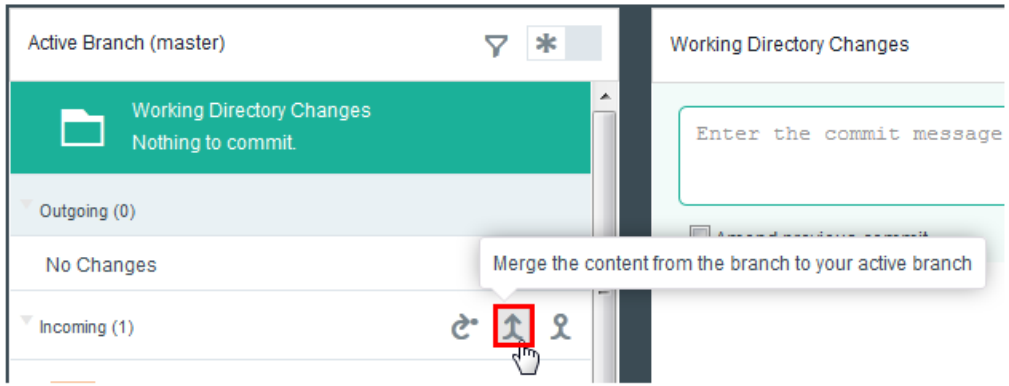
Agile Capabilites - Creating Work Items -5a



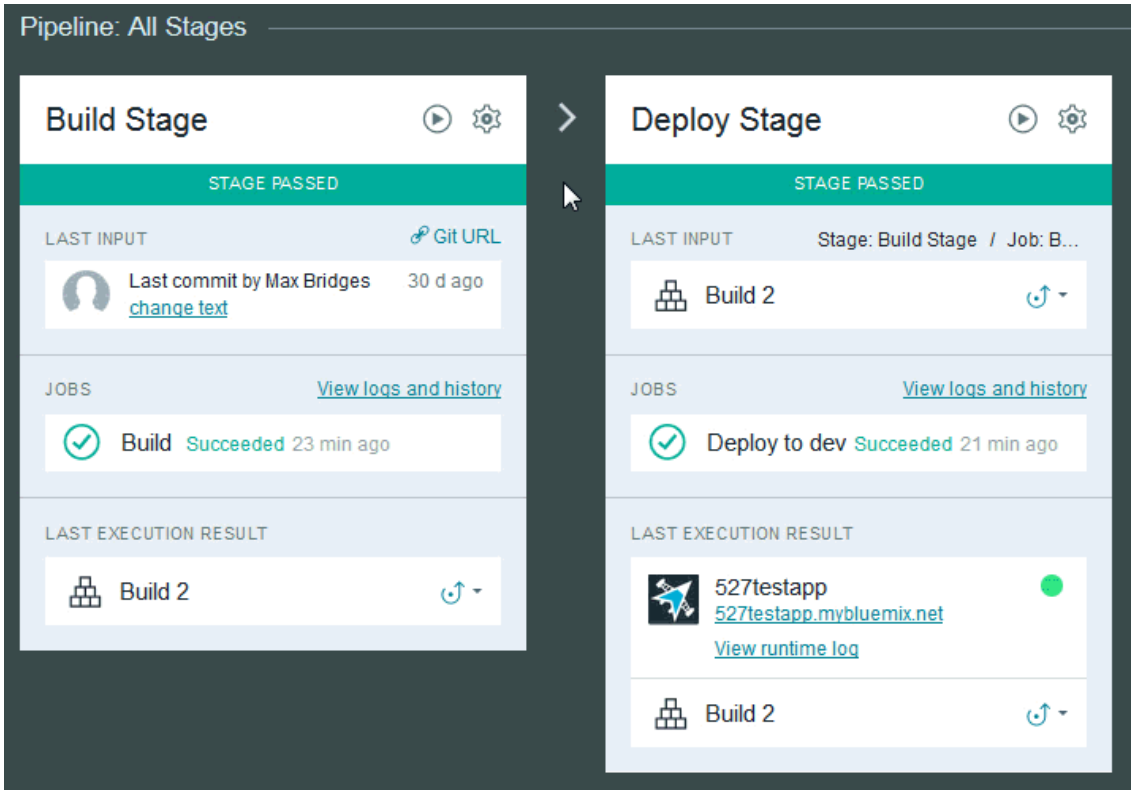
Web Code Editor – 5a



Source Control Management -5a



Build & Deploy - Delivery Pipeline – 5a



Agile Terms and Definitions – 5b

•**Backlog** - A list of features or technical tasks which the team maintains and which, at a given moment, are known to be necessary and sufficient to complete a project or a release.

•**Sprint** - A regular, repeatable, confined work cycle from 1 week to a month.

•**Scrum** - A flexible, holistic product development strategy where a development team works as a unit to reach a common goal.

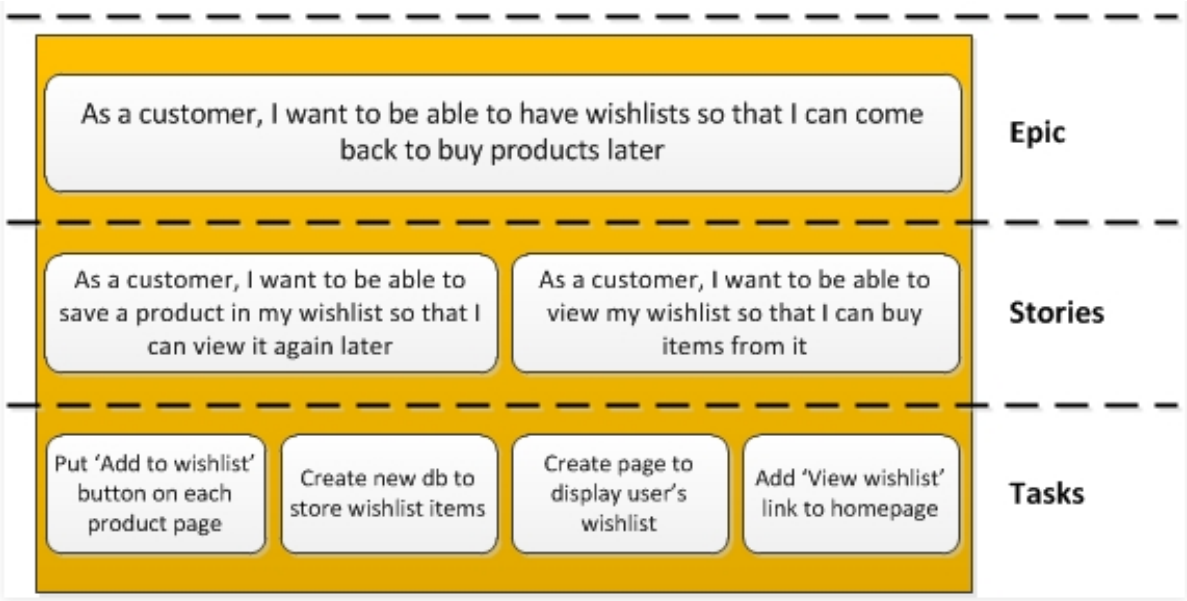
•**Epic** - A large user story that can be broken down into a number of smaller stories.

•**Story** - A tool used to capture a description of a software feature from an end-user perspective. "As [user type], I want [function], because [reason]."

•**Task** - Specific work typically estimated in hours (approx. 2-8), assigned to one person and must be delivered in a single sprint.

•**Defect** - Bug or problem found after a new functionality has been accepted by the product owner.

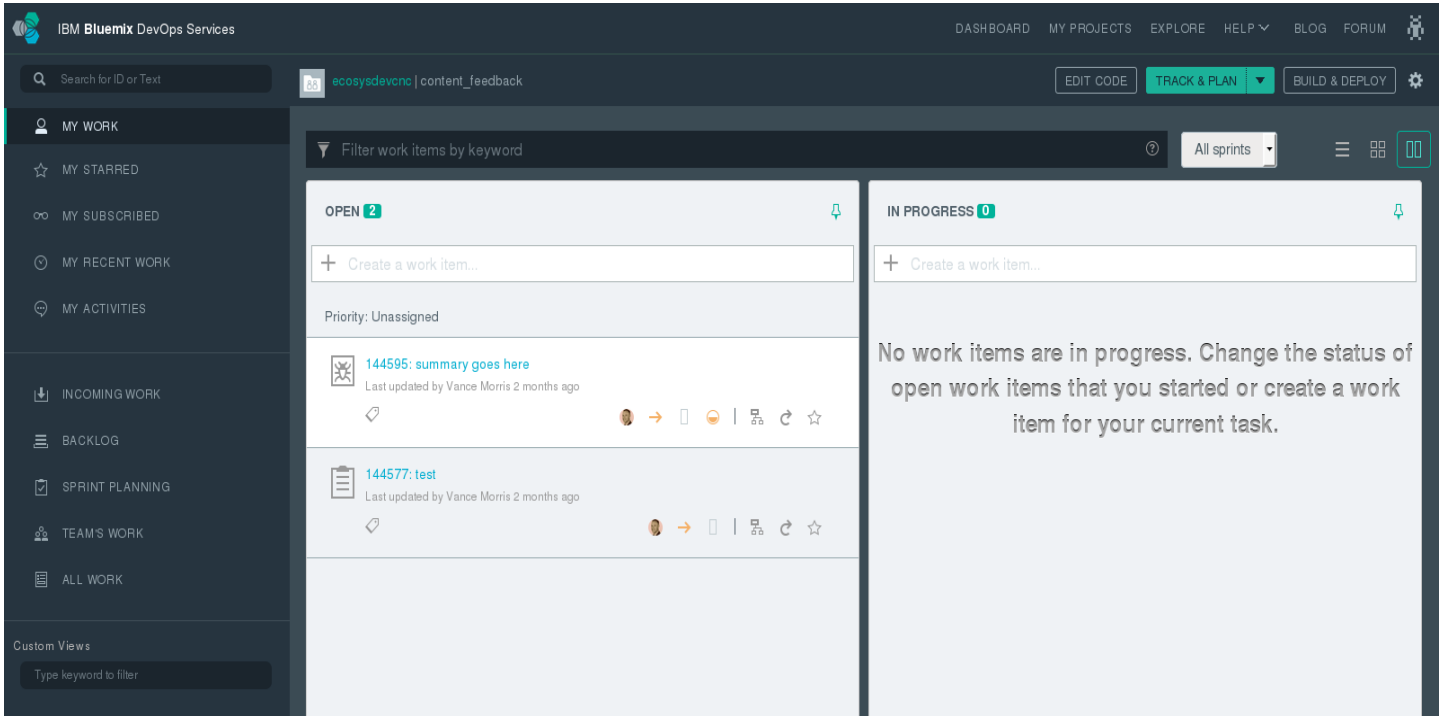
Agile hierarchy – 5b



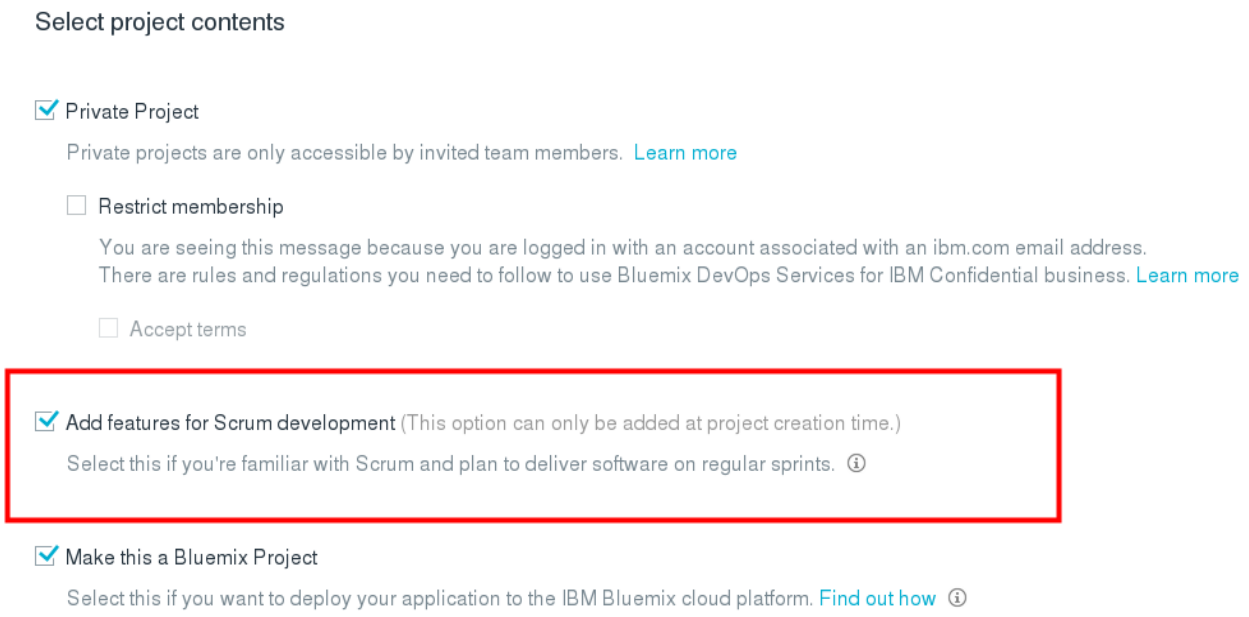
Notes:

- In DevOps Services all 3 are Work Items
- When Work Items are accepted they are added to the Backlog
- Sprint are defined from Backlog

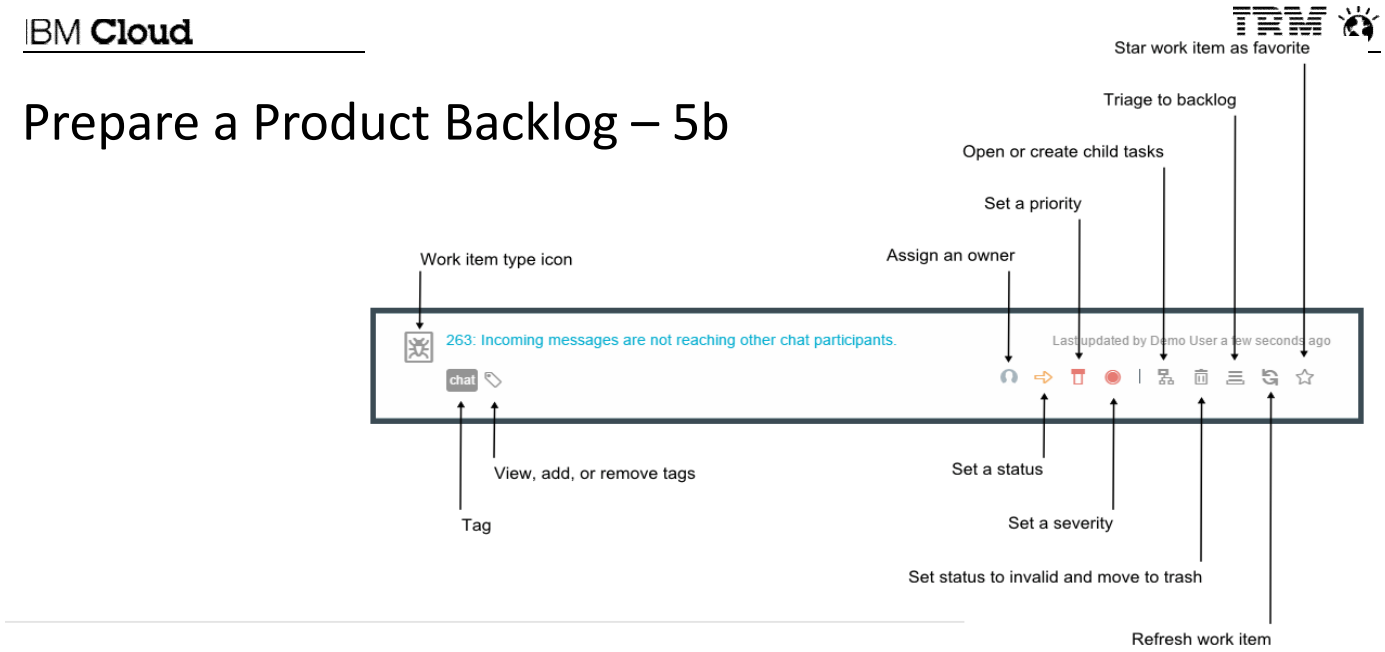
IBM Bluemix DevOps Services Track & Plan – 5b

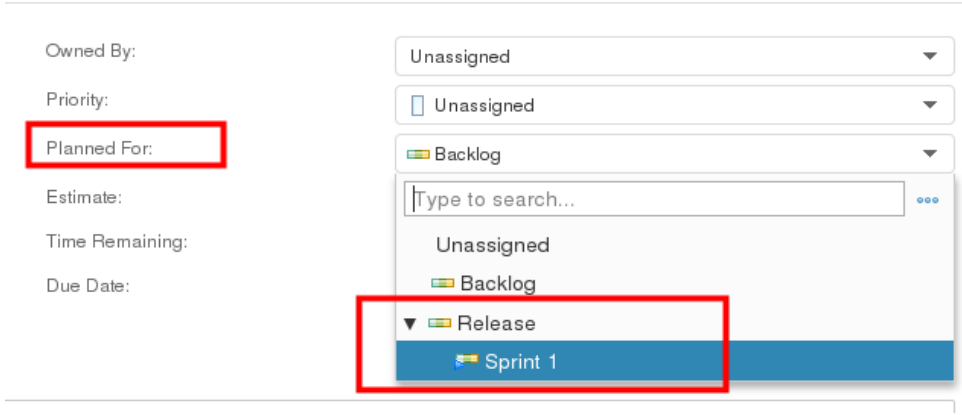


Enable Agile Track and Plan on a Project – 5b

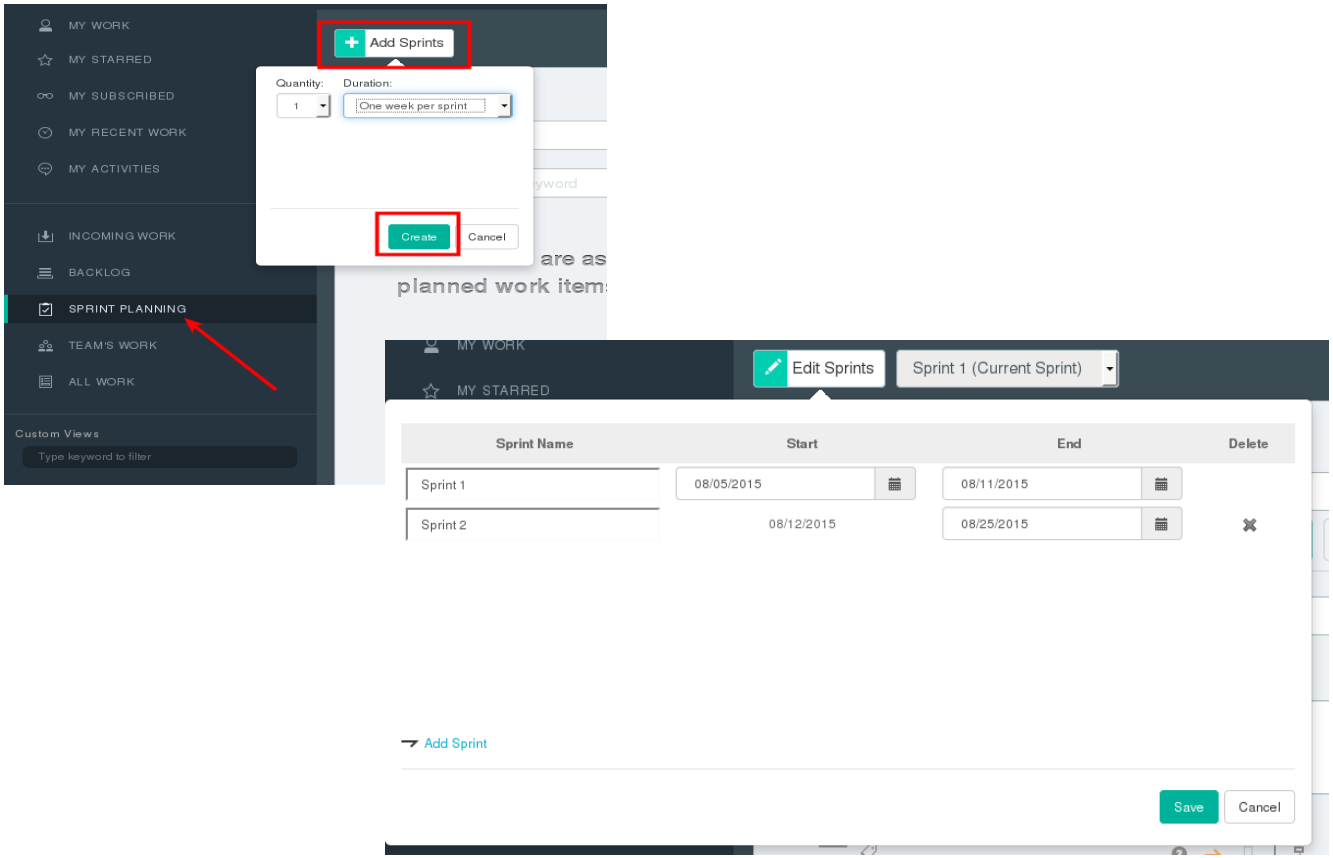


Prepare a Product Backlog – 5b

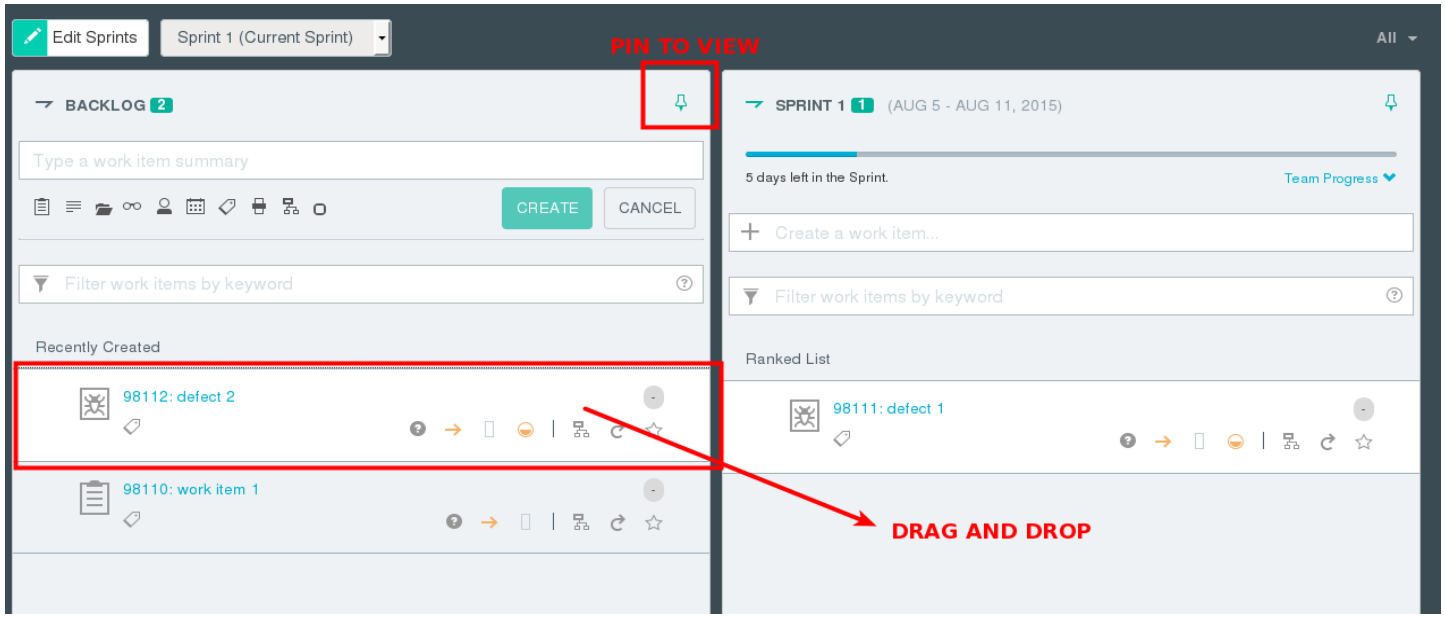




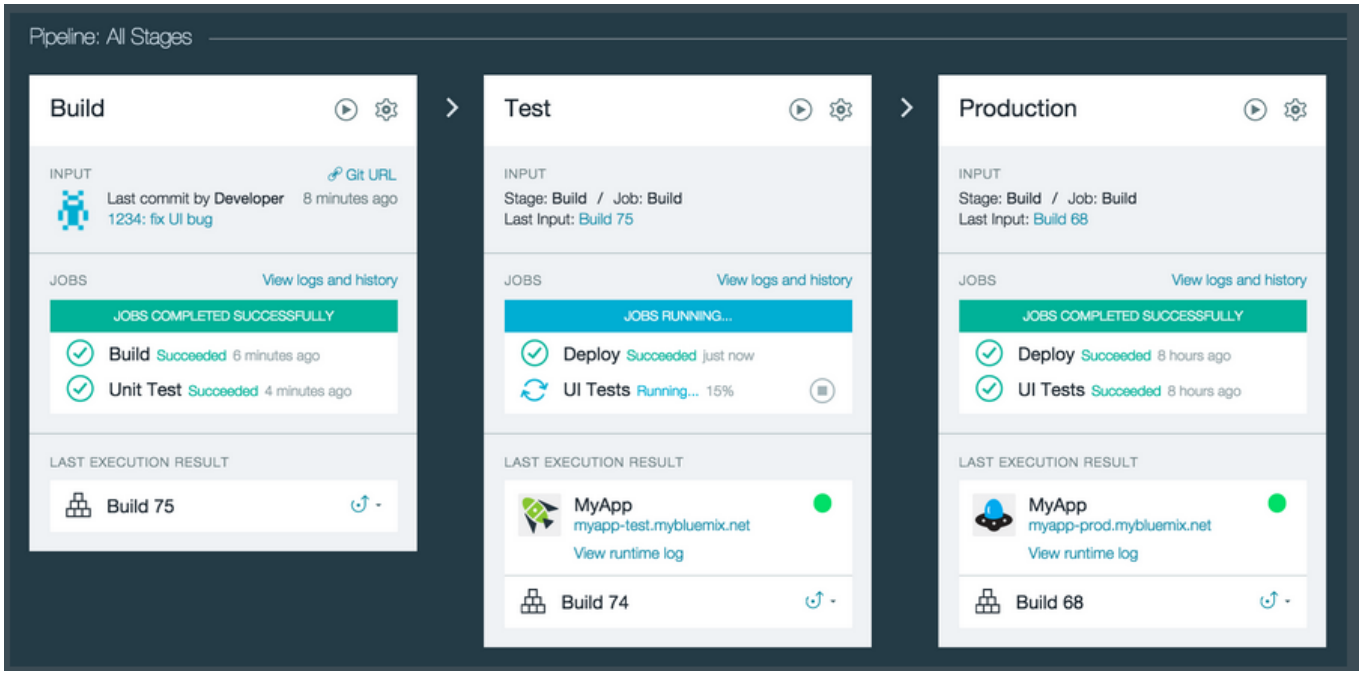
Define New Sprints for a Project – 5b



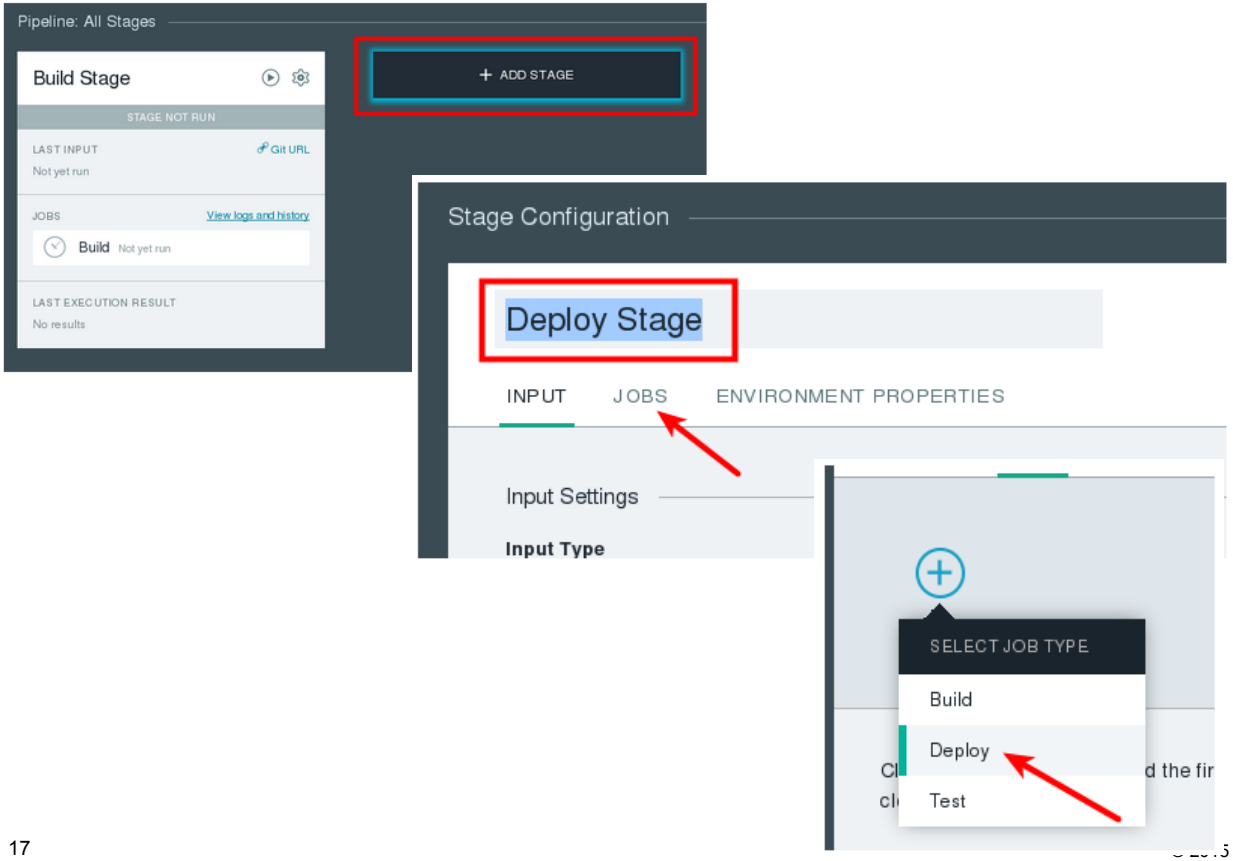
Assign Backlog Items to a Sprint – 5b



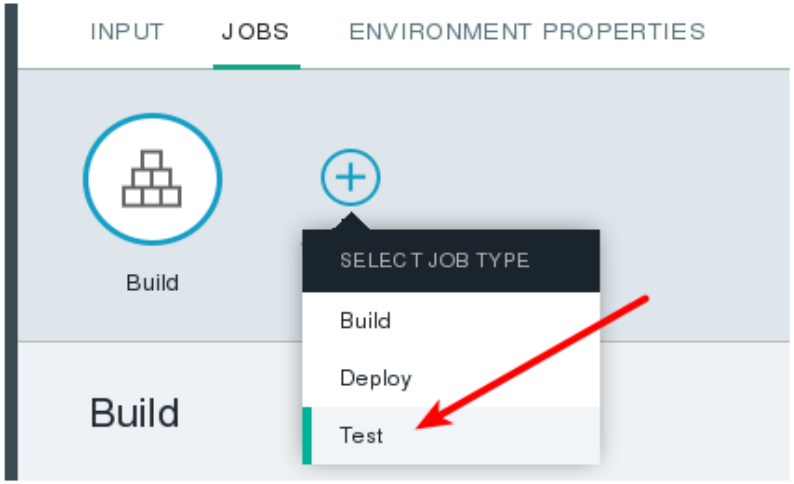
Build & Deploy Pipeline Overview – 5e



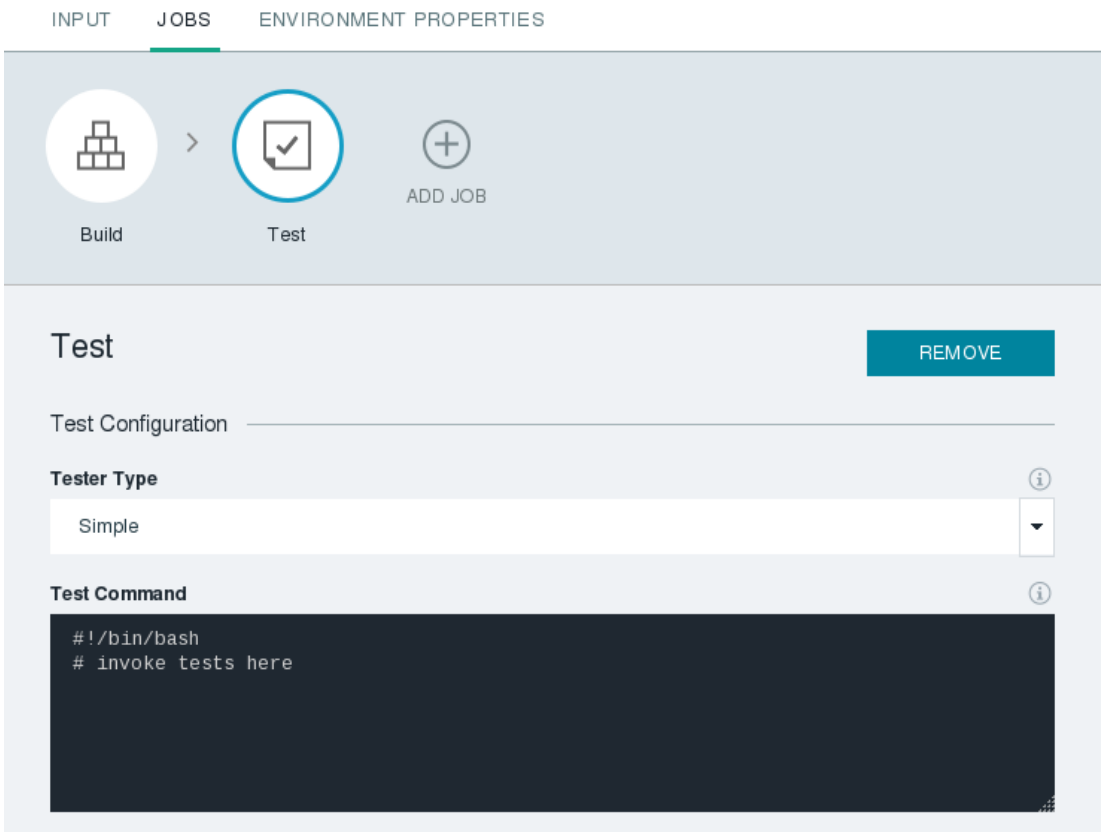
Create a Deployment Stage – 5e



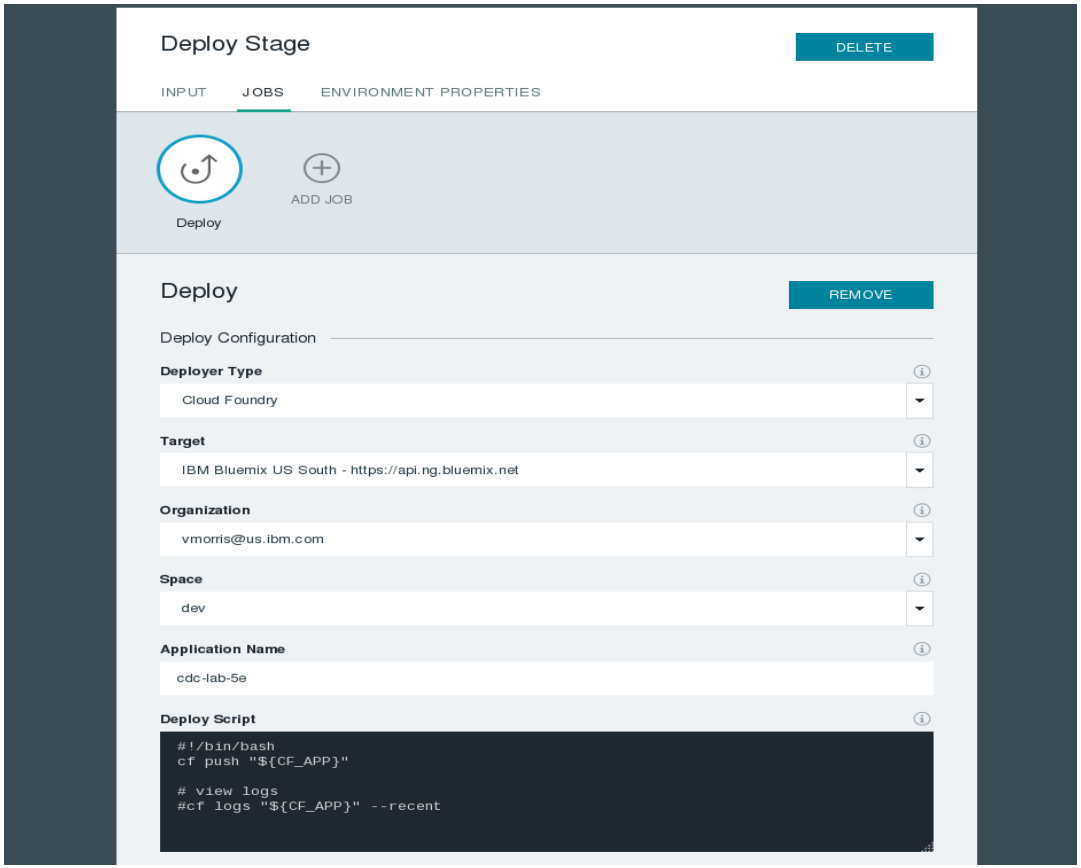
Add & Configure a Test Job – 5e



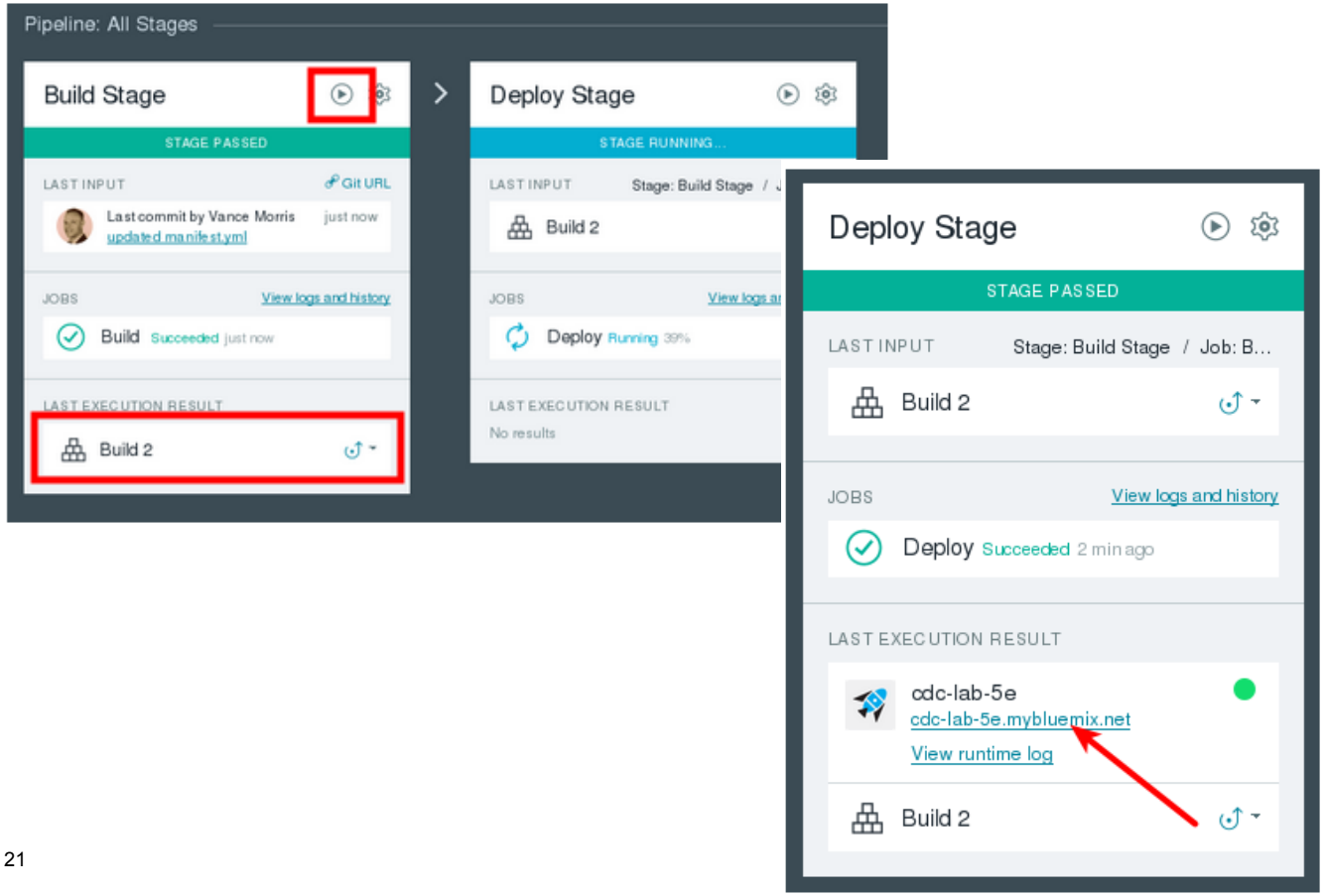
Add & Configure a Test Job – 5e



Configure a Deploy Job – 5e



Deploy Applications to IBM Bluemix – 5e



Section 6: Using Data Services

a. Describe the different types of data services available in IBM Bluemix PaaS

b. Describe the unique features of IBM Bluemix PaaS data services

c. Manage instances of IBM Bluemix PaaS data services: Cloudant NoSQL Database, dashDB, and SQL Database.

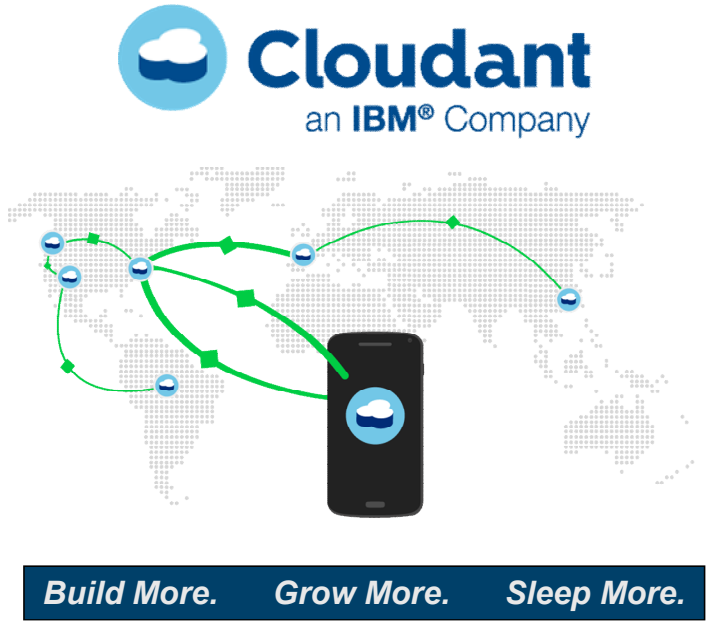
d. Describe the IBM DataWorks service for Bluemix

Types of data services available in IBM Bluemix - 6a

- NoSQL database service
  - Document oriented schema less datastores optimized for horizontal scaling
  - Eaxample - Cloudant
- SQL database service
  - SQL data services based on specific underlying relational databases
    - Example – SQL Database
- In memory columnar database service
  - Column organization optimized for analytic workloads
  - Example dashDB
- Key value pair service
  - Fast in memory storage and retrieval of key value pair data
  - Example IBM Data Cache
- Time Series service
  - Optimized for time series data
  - IBM Time Series Database

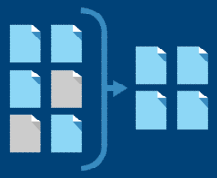
What is the Cloudant NoSQL Service ? – 6b

- A fully-managed NoSQL Database as a Service
- Transactional JSON “document” database with RESTful API
- Can spread data across data centers & devices for scale & HA
- Ideal for apps that require:
  - Massive, elastic scalability
  - High availability
  - Geo-location services
  - Full-text search
  - Occasionally connected users



24

Cloudant API Feature Highlights – 6b



JSON Documents

- Create
- Read
- Update
- Delete

Primary Index

- Exists for every database “OOTB”
- Find docs by their primary key → _id

Secondary Indexes

- Built using MapReduce
- Use when you need to analyze data
- Ex: count data fields, aggregate/sum results, etc.

Search

- Built using Lucene
- Ad-hoc queries
- Find documents based on their contents

Cloudant Documents: _id and _rev – 6b

```
{ "_id":"7f123e23a328bd50ee123cd35452ae47",
  "_rev":"2-3123414209",
  "title":"IBM Cloudant Redbook",
  "author":"Christopher Bienko" }
```

- Each document has an _id (ID) field which is unique per database
  - Any string can be supplied as an _id, but it is recommended you allow Cloudant to generate a UUID (universally unique identifier) for you

- There is also a unique _rev (revision number) field per document
  - Generated by an md5 hash of the transport representation of the document
  - *N*-prefix reflects the number of times this document has been updated
  - Updates to existing documents must provide the latest _rev value, otherwise the update request is rejected

Cloudant API: Returning a single document -6b

- HTTP GET with database name and _id of document . For example to get document with _id **100** from the **authors** database:

GET https://[username].cloudant.com/authors/100

```
{
  "_id": "100",
  "name": "John Smith",
  "agent": "Mary Reid",
 "telephone":"512-555-1212"
}
```

Cloudant API: Inserting a document -6b

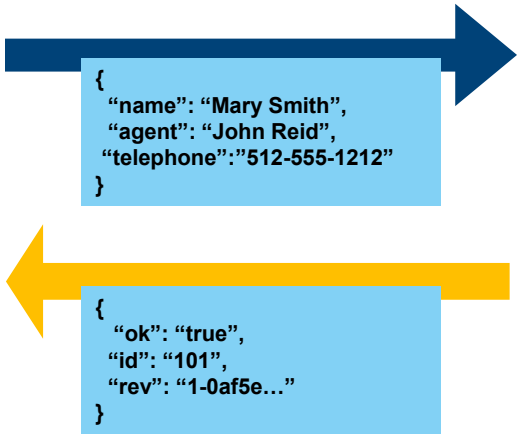
- HTTP PUT or POST

*Via **POST** – document _id in document body*

POST https://[username].cloudant.com/authors

*Via **PUT** – document _id in URL*

PUT https://[username].cloudant.com/authors/101

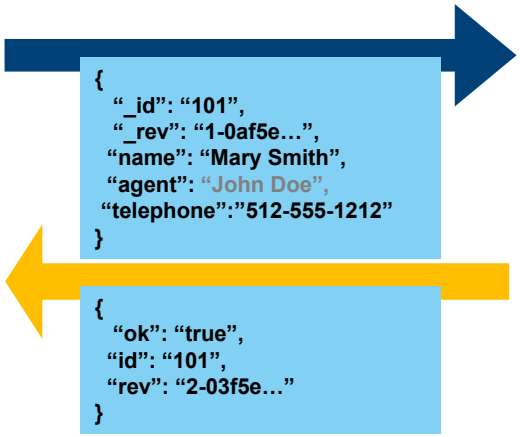


Cloudant API: Updating a document -6b

- HTTP PUT or POST
  - Same HTTP operations as insert - if _id exists then it's an update
  - Latest **_rev** is required or operation will fail

*Via **POST** – document _id in document body*

POST https://[username].cloudant.com/authors



*Via **PUT** – document _id in URL*

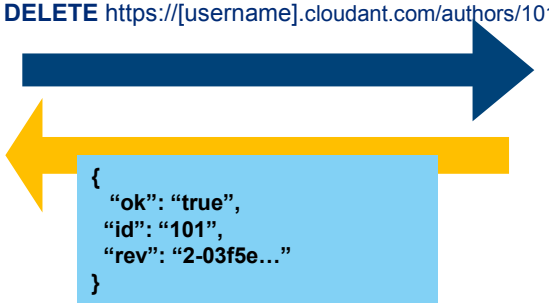
PUT https://[username].cloudant.com/authors/101

Cloudant API: Deleting a document - 6b

- HTTP DELETE or PUT
  - Latest **_rev** is required or operation will fail

*Via **DELETE** – document _id and _rev in URL*

DELETE https://[username].cloudant.com/authors/101?rev=...



*Via **PUT** – document _id in URL , _rev and `"deleted": true` in document body*

PUT https://[username].cloudant.com/authors/101

Cloudant: Secondary Indexes -6b

- Cloudant allows the creation of secondary indexes (or views) that use MapReduce to return specific subsets of data.
  - Map function returns a list of key/value pairs
    - Must be defined for a view
  - Reduce function reduces the list to a single value per key
    - Is optional
- Map and Reduce functions for a view are:
  - written in JavaScript
  - stored in special documents called *design documents*

MapReduce Function Example - 6b

- A list of cars

```
_id: 1
make: Audi
model: A3
year: 2000
bookval: 5400
```

```
_id: 2
make: Audi
model: A4
year: 2009
bookval: 16000
```

```
_id: 3
make: VW
model: Golf
year: 2009
bookval: 15000
```

```
_id: 4
make: VW
model: Golf
year: 2008
bookval: 9000
```

```
_id: 5
make: VW
model: Polo
year: 2010
bookval: 12000
```

- A list of car makes and their book values (map function)

```
function(doc) {
  emit (doc.make, doc.bookval);
}
```

Key

Value

- Aggregated book value by make (reduce function)

```
function(key, values) {
    return sum (values);
}
```

Map results

Key	Value
Audi	5400
Audi	16000
VW	9000
VW	12000
VW	15000

Reduce results

Key	Total Book Value
Audi	21400
VW	36000

Cloudant Sync

- Native replication feature that allows you to push database access to mobile devices, remote facilities, sensors, and Internet-enabled devices
- Enables mobile and distributed apps to scale by replicating and syncing data between multiple readable, writable copies of a database even on mobile iOS and Android devices
- Simplifies large-scale mobile development by enabling you to create a single local database for every user.
  - Reduces round-trip database requests with the server because when there is no network connection, the app runs off the database on the device.
  - When the network is restored local data is synced with server

Unique features of dashDB, SQL Database and Time Series Database - 6b

- dashDB service
  - data warehousing service for relational data, including special types such as geospatial data.
  - Data can be analyzed with SQL or advanced built-in analytics like predictive analytics and data mining, analytics with R, and geospatial analytics
- SQL Database service
  - DB2 backed database service with high availability, automated backups and data privacy.
- Time Series Database service
  - Managed data store for time stamped Internet of Things device data
  - Optimized storage for large volumes and time based SQL extensions

Manage instances of IBM Bluemix PaaS data services: Cloudant NoSQL Database, dashDB, and SQL Database. – 6c

- Manage instances of the Cloudant NoSQLDB service
  - Create a database
  - Add data to an existing database
  - Edit documents in an existing database
  - Clone existing documents
  - Simple query of all documents in an existing database
- Manage instances of dashDB service
  - Create a new table in dashDB
  - Browse the contents of an existing table in dashDB
  - Run SQL scripts in dashDB
  - Import CSV data into dashDB
- Manage instances of SQL Database services
  - Create a new table in SQL Database
  - Browse the contents of an existing table in SQL Database
  - Run SQL SELECT Queries in SQL Database
  - Import CSV data into SQL Database

IBM DataWorks Forge - the IBM® data refinery – 6d

- Provides a lightweight, cloud-based set of data refinement and access services
  - Makes fit-for-purpose data quickly and easily available to everyone across the enterprise.
- Ingests raw data, and provides services to enable you to perform:
  - Data preparation
    - Delivers quality scores & value distributions, so that users can visualize & understand the data
    - Allows users to quickly & easily shape, enrich & improve quality by joining, standardizing, filtering, removing duplicates, etc.
  - Data movement and delivery
    - Allows users to apply their changes and deliver the data to her chosen target – or targets

Cloud Application Developer Certification Review Sample Questions

Presented by:

IBM Cloud Ecosystem Development

IBM **Cloud**

1a *Describe Cloud service models and IBM Cloud offerings*

(1) What two items are managed by the developer in a Platform as a Service (PaaS) service model?

A. data
~~B. storage~~
C. application
~~D. networking~~
~~E. operating system~~

Answer **AC**

1b . *Describe the different capabilities of IBM Bluemix*

(2) What are two capabilities of the IBM Bluemix Container Service ?

~~A. Direct import of virtual machines from various IaaS providers~~

B. Persistent storage capability for running images

~~C. Supports a variety of runtimes via an extensible buildpack architecture~~

~~D. Direct support for running images via both Docker and Warden~~

E. Elastic scaling and auto recovery

Answer **BE**

2c. Explain the process of staging an application in IBM Bluemix PaaS

(3) In IBM Bluemix PaaS, what is the role of a buildpack in staging an application?

A. to provide an app framework and runtime support to the Droplet Execution Agent

B. push application code from the development workstation to IBM Bluemix PaaS

C. notify Health Manager of successful application instance start

D. start application request handling by sending a status message to the Router

Answer **A**

2e. Understand IBM Bluemix Regions and how to manage applications in multiple regions

(4) How can an IBM Bluemix Paas developer distinguish between multiple regions when using the Cloud Foundry CLI (cf) tool ?

A. By using a region specific user id and password with the cf login command
B. By using the --region command line parameter with each cf command
C. By setting the environment variable REGION
D. By using a region specific api endpoint with the cf login command

Answer **D**

3a *Understand how to design, develop, deploy and manage a IBM Bluemix PaaS application following the Twelve-Factor App methodology*

(5) Based on the Twelve-Factor App methodology, how should a developer handle their application's log output on IBM Bluemix PaaS?

A. write as an unbuffered stream to stdout

~~B. write to the application runtime file system~~

~~C. write to a logging service external to the application~~

~~D. follow the logging best practices for each specific runtime~~

Answer **A**

3b *Explain various methods to monitor an application in IBM Bluemix PaaS*

(6) A developer is investigating performance issues with a web application in IBM Bluemix PaaS by using the Monitoring and Analytics service. Which two metrics are provided for the developer by this service ?

A. CPU usage

~~B. disk space usage~~

C. request response time

~~D. network latency~~

~~E. network throughput~~

Answer **AC**

4a *Understand how to configure external authentication using IBM Bluemix PaaS web applications with the Single Sign On service (SSO)*

(7) A developer has an IBM Bluemix PaaS application that requires authentication and they wish to allow users to pick between their Google, Facebook and LinkedIn identities to log in. If they use the Bluemix SSO Service what is the most efficient way to accomplish this ?

A. Create multiple instances of the SSO service each configured for a single identity provider and have the application logic select the appropriate instance for each user login

B. Configure a single SSO instance for multiple identity providers and have the application logic select the appropriate provider for each user login

C. Write custom code in the application's SSO callback to directly interface with each identity provider via the providers's SSO API

D. Extend the SSO service by adding a custom identity provider module to cycle through all three identity providers for each login attempt.

Answer **B**

4c Enable loosely coupled integration for IBM Bluemix PaaS applications and components by using Messaging Services

(8) Multiple IBM Bluemix PaaS application instances use the same IBM MQ Light service instance. If a destination is created in this instance using the SHARING option and all application instances are attempting to consume messages from this destination, which of the following is true ?

A. Exclusive access to all messages is given to the first application instance to start consuming messages from the destination.

B. Each message arriving at the destination is delivered to only one app instance.

C. Each message arriving at the destination is delivered to all application instances.

D. All application instances will be notified of a new message arrival but only one will be able to consume it.

Answer **B**

5b *Plan and track work for agile team collaboration*

(9) A development team is using the agile process and has a series of work items that need to be prioritized in order to plan their first sprint . Which IBM Bluemix DevOps Services Track & Plan feature would best meet the team's needs?

~~A. Scrum~~

B. Backlog

~~C. Project Planner~~

~~D. Iteration Planner~~

Answer **B**

5e *Describe how to use the Build & Deploy option to manage continuous integration and continuous delivery*

(10) Which 2 of the following are true about *jobs* in the IBM® Bluemix™ DevOps Services Delivery Pipeline?

A. The jobs in a stage run sequentially

~~B. Jobs can be either created as part of a stage or independent of any stage~~

~~C. The jobs in a stage run in the order designated in the **Run Order** attribute of the stage~~

~~D. Build jobs can only be defined for projects developed with compiled languages like Java~~

E. Jobs run in discrete working directories created by the pipeline

Answer **AE**

6a *Describe the different types of data services available in IBM Bluemix PaaS*

(11) Which of the following application types would benefit the most from a NoSQL database service like Cloudant that prioritizes availability over immediate consistency ?

~~A. A stock trading app with a large number of concurrent users~~

~~B. An airline reservation app with a large number of concurrent users~~

C. A news portal with a large number of concurrent users that permanently captures user comments for each news article

~~D. A live auction website app with a large number of concurrent users~~

Answer **C**

6b *Describe the unique features of IBM Bluemix PaaS data services*

(12) Which of the following is true about the _*rev* field in a Cloudant document when using the Cloudant API for Create, Retrieve, Update, and Delete operations (CRUD) on that document ?

A. It is required input for creates and updates only

B. It is required input for deletes and updates only

C. It is required input for retrieves and updates only

D. It is required input for deletes and retrieves only

Answer **B**