(11 of 12)

Updated: Jan 9th, 2017

IBM Bluemix Development & Certification

Summary decks for a course that covers the A to Z of IBM Bluemix.

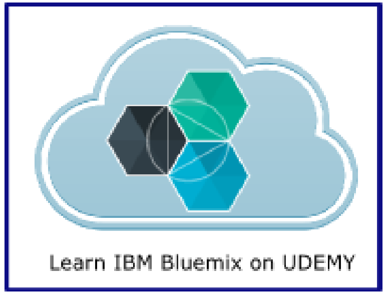
For more information visit:
http://www.acloudfan.com

raj@acloudfan.com

1. DevOps – Intro
2. Edit Code
3. Agile Tracking & Planning
4. Build & Deploy

PS: Certification practice test questions NOT included in the summary decks

Discounted access to the courses:



https://www.udemy.com/ibm-bluemix/?couponCode=BLUE100

Coupon Code = **BLUE100**

https://www.udemy.com/rest-api/?couponCode=REST100

Coupon Code = **REST100**

PS:

- For latest coupons & courses please visit: http://www.acloudfan.com
- Enter to **WIN Free access** – please visit: http://www.acloudfan.com/win-free-access

raj@acloudfan.com
http://www.acloudfan.com

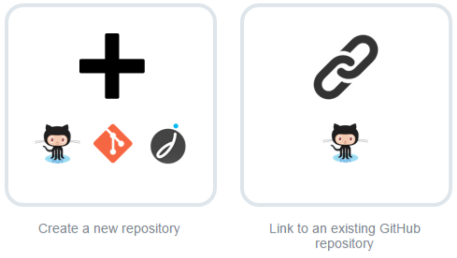
Data Service Types

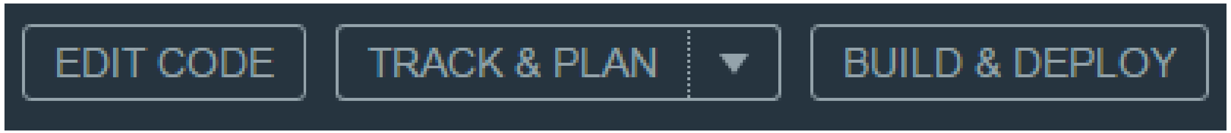
Introduction to IBM DevOps

- DevOps services is a Software as a Service (SaaS) on the cloud that supports continuous delivery

  - Build applications
  - Planning
  - Collaborate with the team members
  - Track tasks, defects etc.
  - Automate the builds, tests and deployment process

Bluemix DevOps





- **Web IDE**
- Debugging
- Code push

- **Agile planning**
- Sprint tracking

- **Automate builds & deploy**
- Delivery pipeline

raj@acloudfan.com
http://www.acloudfan.com

Bluemix SCM

- Create a GitHub repo
- Create a Bluemix Git repo
- Create a Jazz SCM repo

- Link to existing GitHub repo

Edit Code

Web IDE

1. Browser based development environment
2. Content assist, code completion and error checking for JavaScript, HTML/CSS
3. Integrated with SCM
4. Syntax highlighting for most file types
5. Editor may be personalized for example color, tools, settings
6. Debugging of Node/Javascript apps

Deploying from editor

- Create a launch configuration
- Multiple launch configurations
  - Multiple team members
  - Multiple test targets i.e., orgs/spaces

Live Editor

- Changes automatically saved to SCM
- Saves time as restarting app is faster than redeployment
- Debugging of NodeJS applications
  - Debug live application – supported in Chrome & Opera browsers only
  - Bash shell to accessthe container for the application

Deployment, Live edit & Debugging

1. Deploy from workspace by creating launch configurations
2. Live edit mode for quick changes with need for app push
3. Debugging
   - Logs
   - Shell & Debugger
4. Live sync tools allow synching of local file system with DevOps projects

raj@acloudfan.com
http://www.acloudfan.com

Agile Planning & Tracking

Agile planning through Track & Plan

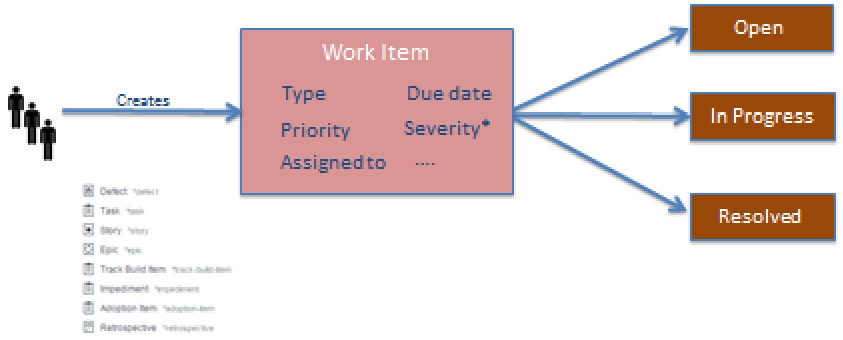
1. Create & Triage work items

2. Sprint planning

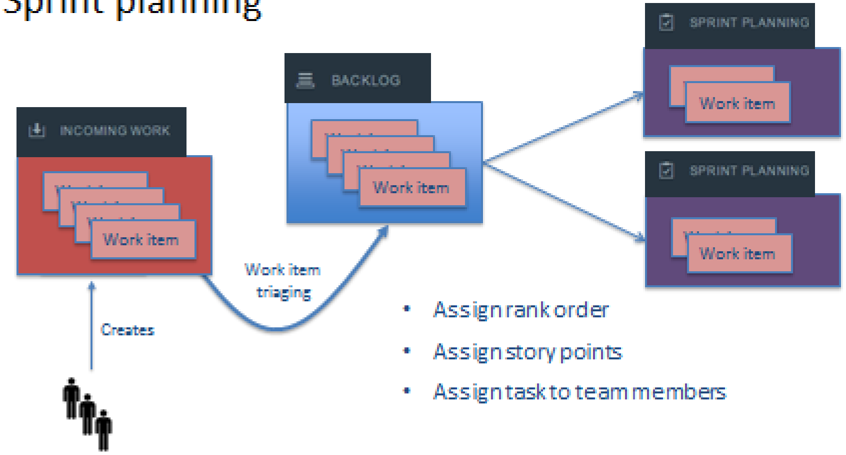
3. Track Progress (Member & Team)

Work items

- Unit of work



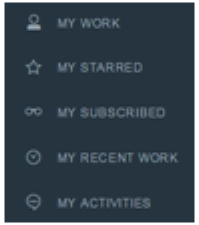
Sprint planning



- Assign rank order
- Assign story points
- Assign task to team members

Tracking

- Member work item tracking



- Team progress tracking
  - Hours worked vs. total hours estimated
  - Work items completed vs. total work items
  - Story points achieved vs. total estimated story points

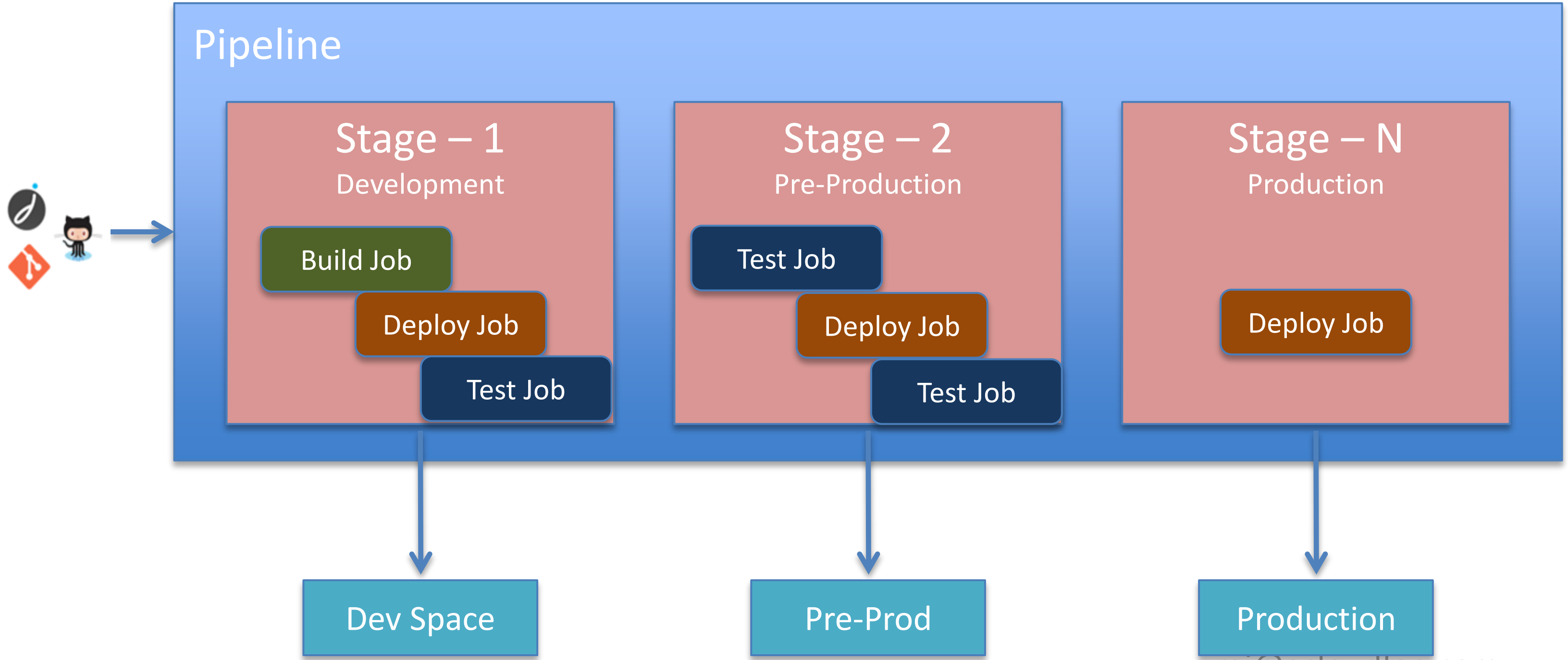
Summary

- Team members create different types of work items which become part of the *incoming work*
- The work items are triaged to the *backlog*
- Sprint planning is carried out
- Member tracks and works on the items assigned
- Overall project tracking can be done at sprint, backlog and all work level

Build, Deploy & Test

Build & Deploy | Pipeline

- Automated continuous deployment of projects



Stages

- Stages execute sequentially within the pipeline
- Stages receive input from the previous stage or from source control repos
- Triggering of stage leads to execution of jobs within the stage
  - Automatic execution anytime changes pushed to the source control repo
  - May be set to execute manually
  - May be set to execute automatically on completion of previous stage

Jobs

- Execution unit within a stage
  - Build job
  - Deploy job
  - Test Job
- By default stage execution stops if a job fails
- Jobs in the stage cannot pass artifacts to each other
  - Stage environment properties are shared by jobs
  - E.g., CF_APP, CF_ORG
  - E.g., MY_APP_PROPS

Deployment Targets

- Manifest file controls how the project is deployed
  - There will be a *Route* conflict
- To support multiple targets use the *cf push* command line options
  - Use the *cf push* command with –n to set the route

```
cf push "${CF_APP}" –n stage_host
```

```
cf push "${CF_APP}" –n "${my_env_var}"
```

```
cf push "${CF_APP}" --random-route
```

Key points to remember

- Automation is the driver for the delivery pipeline
- Deploy may be triggered by SCM commit/push
- Testing jobs may be triggered before/after build/deploy