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

Cloud Foundry

1. Architecture
2. CF CLI commands
3. CF Push & Manifest file
4. Environment variables
5. No-Route Applications

PS: Certification practice test questions NOT available 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**

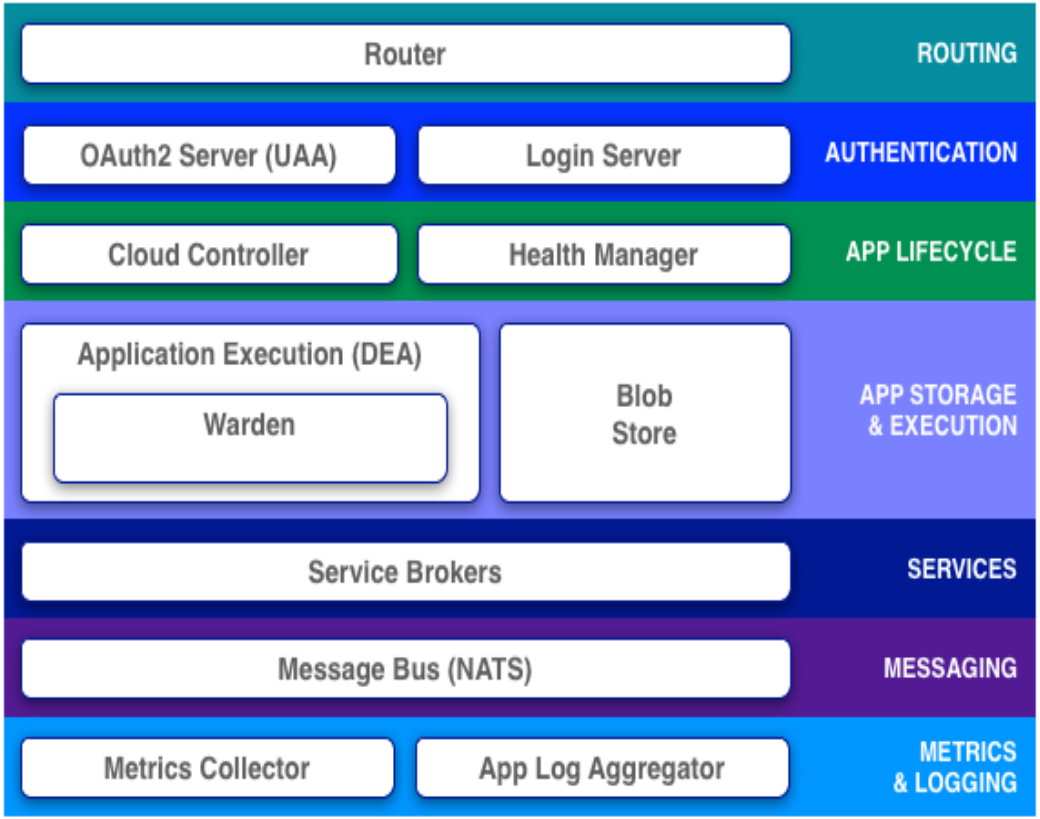
PS:

- 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

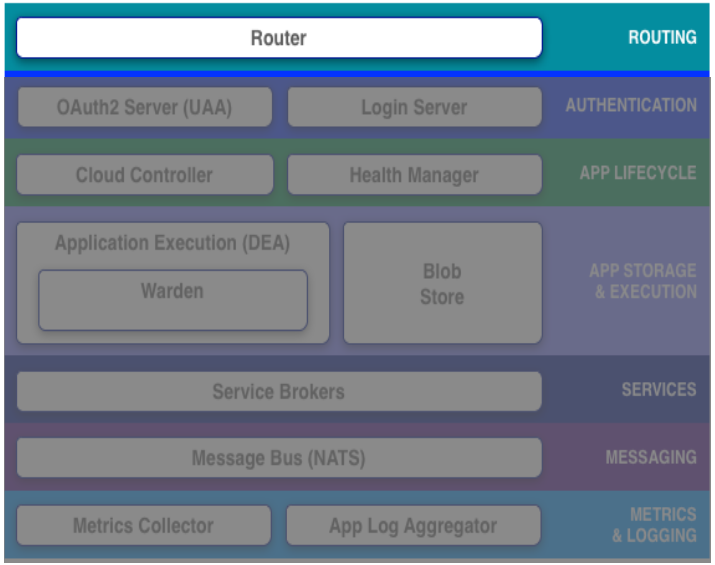
Cloud Foundry Architecture

Astronaut's view

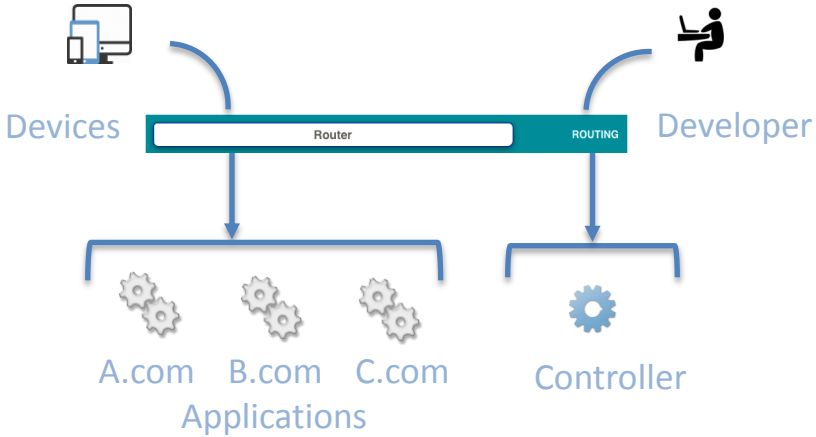


raj@acloudfan.com
http://www.acloudfan.com

Router

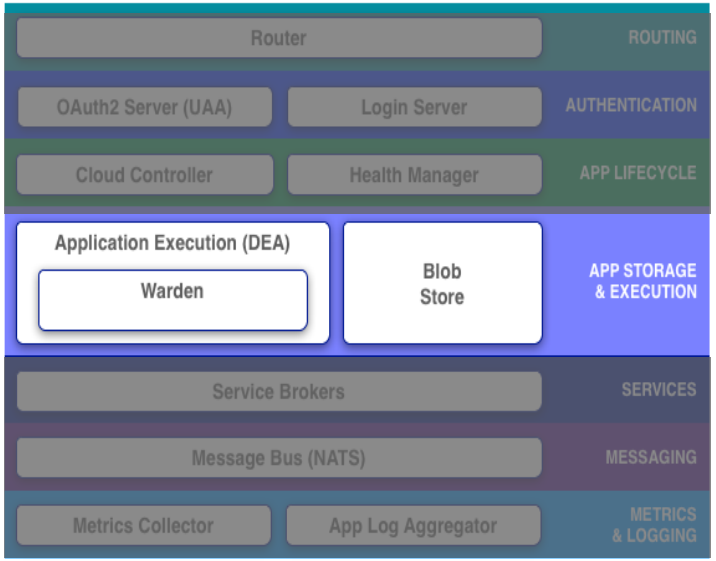


- All calls to applications on CF end up on Router
- Routes to the appropriate application and controller



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

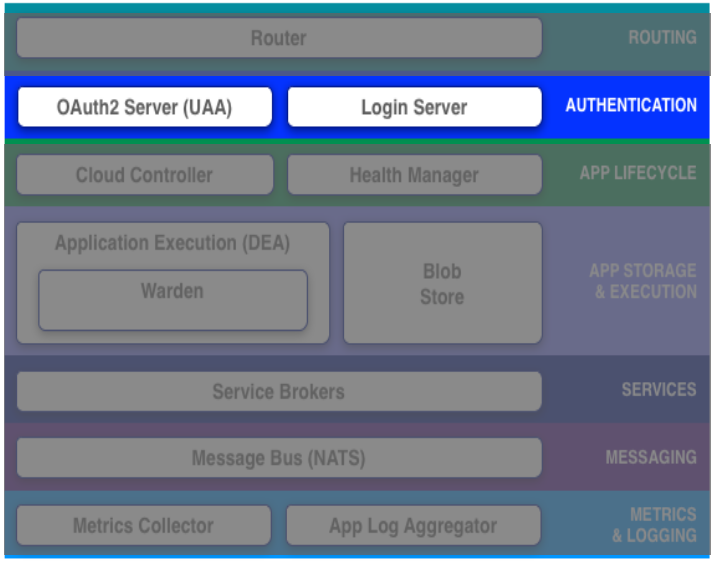
App Storage & Execution



- Droplet Execution Agent (**DEA**) manages application instances, tracks instances & broadcast state
- Applications sit in **warden** containers
  1. Provides isolated environment
  2. Limit CPU, memory, disk & network usage
- Blob Stores hold – Application code, Buildpacks, Droplets

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

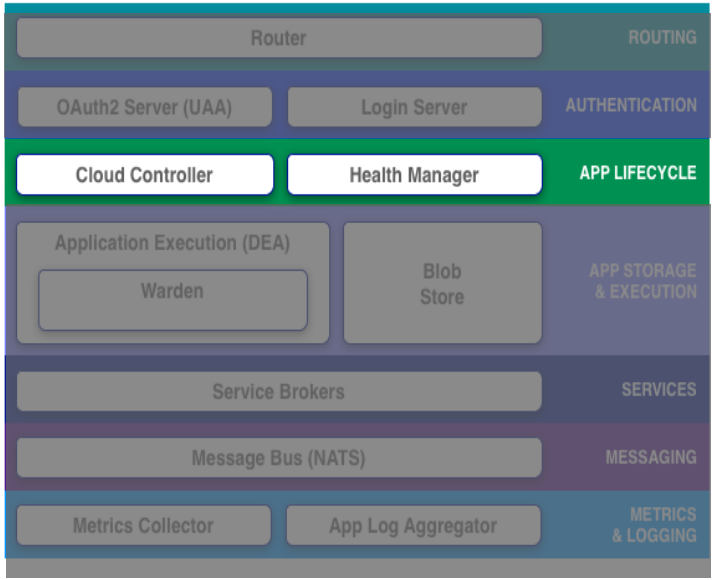
Authentication | Identity Management



- User Account & Authentication (UAA) is the identity management service
  - OAuth2 provider
  - Authenticate with login server
  - Single Sign On (SSO) for users
  - Endpoints for managing user accounts & other functions

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

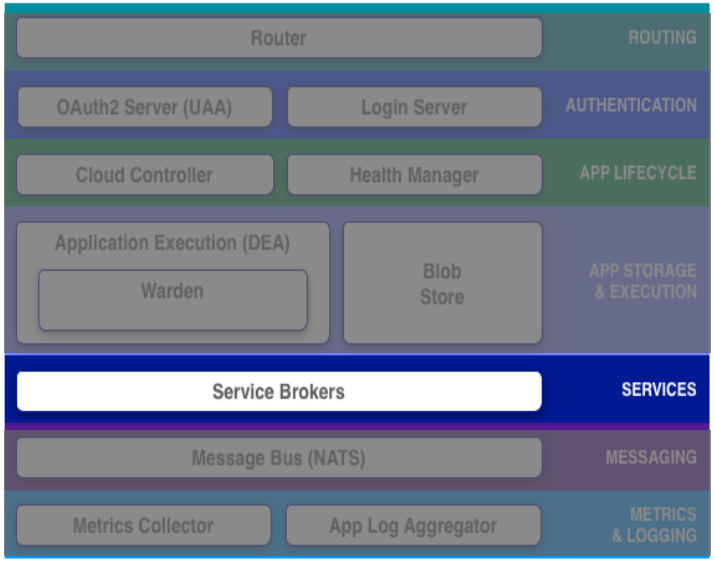
Cloud Controller & Health Manager



- Cloud Controller provides command and control system
  - Exposes the CF API
  - Maintains records of domain objects
  - Directs DEA to stage/run apps
- Health Manager
  - Monitors the health of apps
  - App state reconciliation
  - Directs CC to address issues found

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

Service Brokers

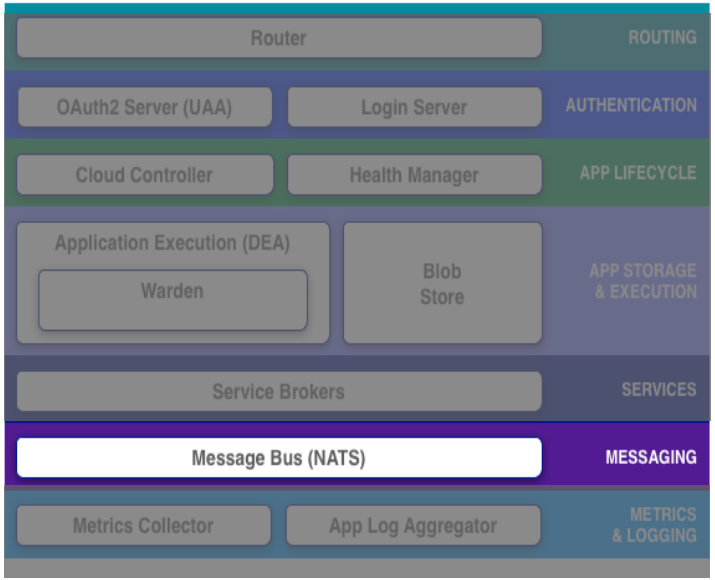


- Manages the services
  - Advertises service catalog
  - Service provisioning

    Create Bind Unbind Delete
  - Broker API for service providers

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

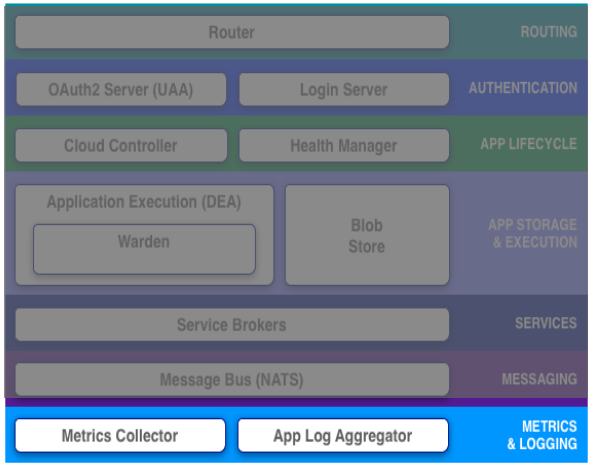
Message Bus (NATS)



- High performance messaging
  - Internal to Cloud Foundry
  - Publish/Subscribe, Queues
  - Non persistent messages

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

Metrics & Logging

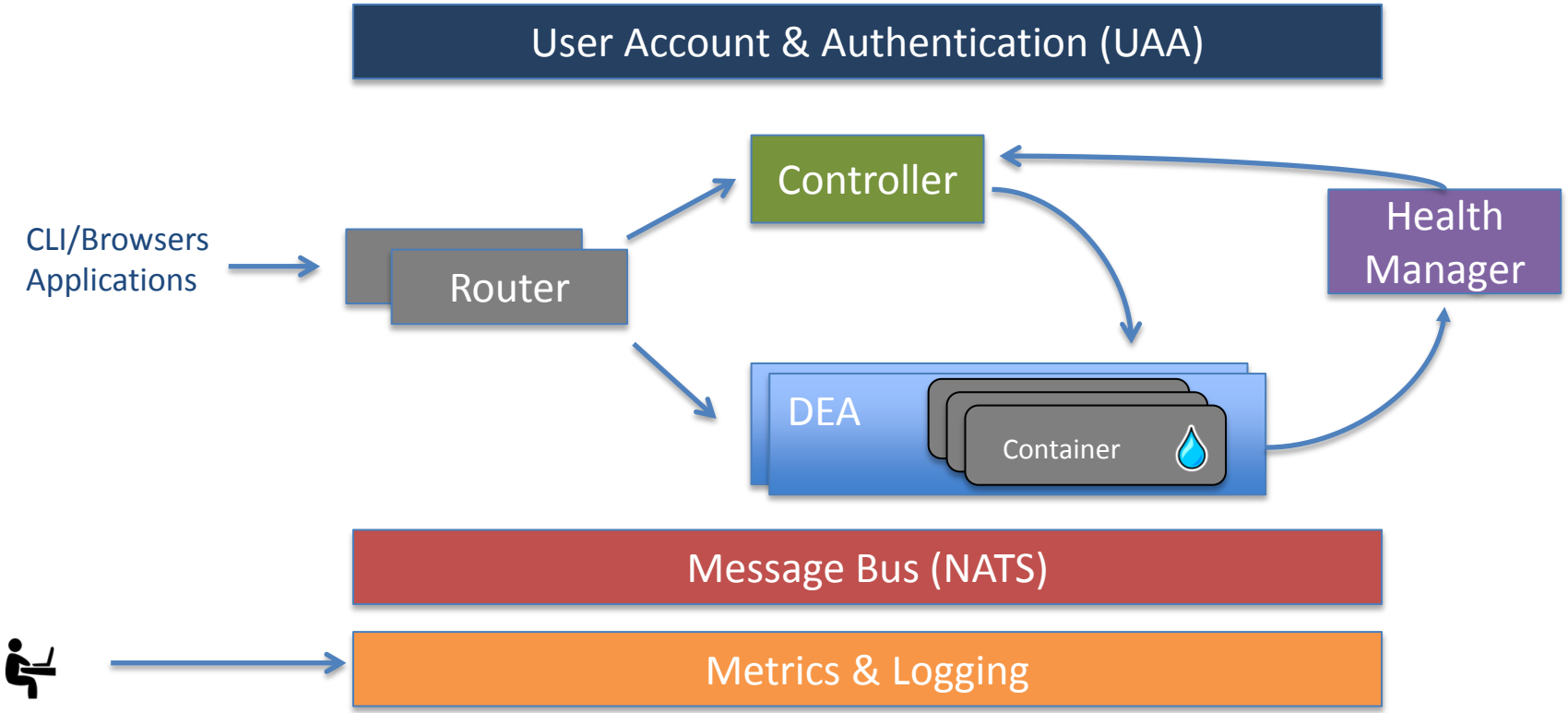


- Gather metrics from components
  - E.g., number of router.requests
  - E.g., cc.requests.completed
  - Enables CF health management
- Application Log aggregator
  - Streams app logs to developers
  - By default to terminal

```
2014-02-13T11:44:52.11-0800 [API]     OUT Updated app with guid e1ca6390-cf78-
```

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

Summary



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

CF CLI Commands

CLI Commands - basic

cf -help, -h	Prints the help
cf --version, -v	Prints CLI version
cf logout	Logs out the user
cf passwd	Change the account password

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

CLI Commands – target view, set

cf target [-o ORG] [-s SPACE]

Sets the target i.e., Org & Space for the subsequent commands

Org	Select the org
Space	Select the space in org

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

CLI Commands - Domains

cf domains Lists the domains

cf create-domain ORG domain
Creates the domain under the ORG

cf delete-domain domain -f
Deletes the domain if –f for forced deletion

cf create-shared-domain
cf delete-shared-domain
Only PaaS admin can use these commands

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

CLI Commands - Routes

cf routes	Lists all routes in the current space or organization
cf create-route	Create a url route in a space for later use
cf check-route	Checks if route exists
cf map-route	Maps the route to an application
cf unmap-route	Remove a url route from an app
cf delete-route	Delete a route
cf delete-orphaned-routes	Delete the orphaned route(s)

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

CLI Commands - Quotas

Quota plans = named sets of memory, service, and instance usage quotas.

cf quotas	Lists available usage quotas
cf quota	Show quota info
cf set-quota	Assign a quota to an org
cf create-quota	Define a new resource quota
cf delete-quota	Delete a quota
cf update-quota	Update an existing resource quota

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

CLI Commands - Organizations

cf orgs	List all orgs
cf org	Show org info
cf rename-org	Rename the org
cf delete-org	Delete the org
cf create-org	Create and org

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

CLI Commands - Spaces

cf spaces	List all spaces under the org
cf space	Show space info
cf create-space	Create space
cf rename-space	Rename the space
cf delete-space	Delete the space

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

CLI Command – Application Information

cf apps	Lists the apps under the selected target (Organization & Space)
cf app APP_NAME	Shows the health/information for selected App

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

CLI Commands - Application Management

cf start, stop	Start and stop application
cf logs, events, files	View logs, files & events related to an app
cf delete	Delete the application
cf restart-app-instance	Restart the application
cf restage	Restage an application

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

CF push & Manifest file

CLI Command - Application push

```
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]
```

Pushes the application to Bluemix/Cloud Foundry PaaS

You may push with or without the **manifest** file

You may push multiple apps with the manifest file

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

Manifest File – manifest.yml

- Manifest tells *cf* push what to do with the applications
- Manifest file contains the various arguments in YML format
- By default *cf push* uses a manifest file in current directory
- Applications may be pushed without the manifest as well

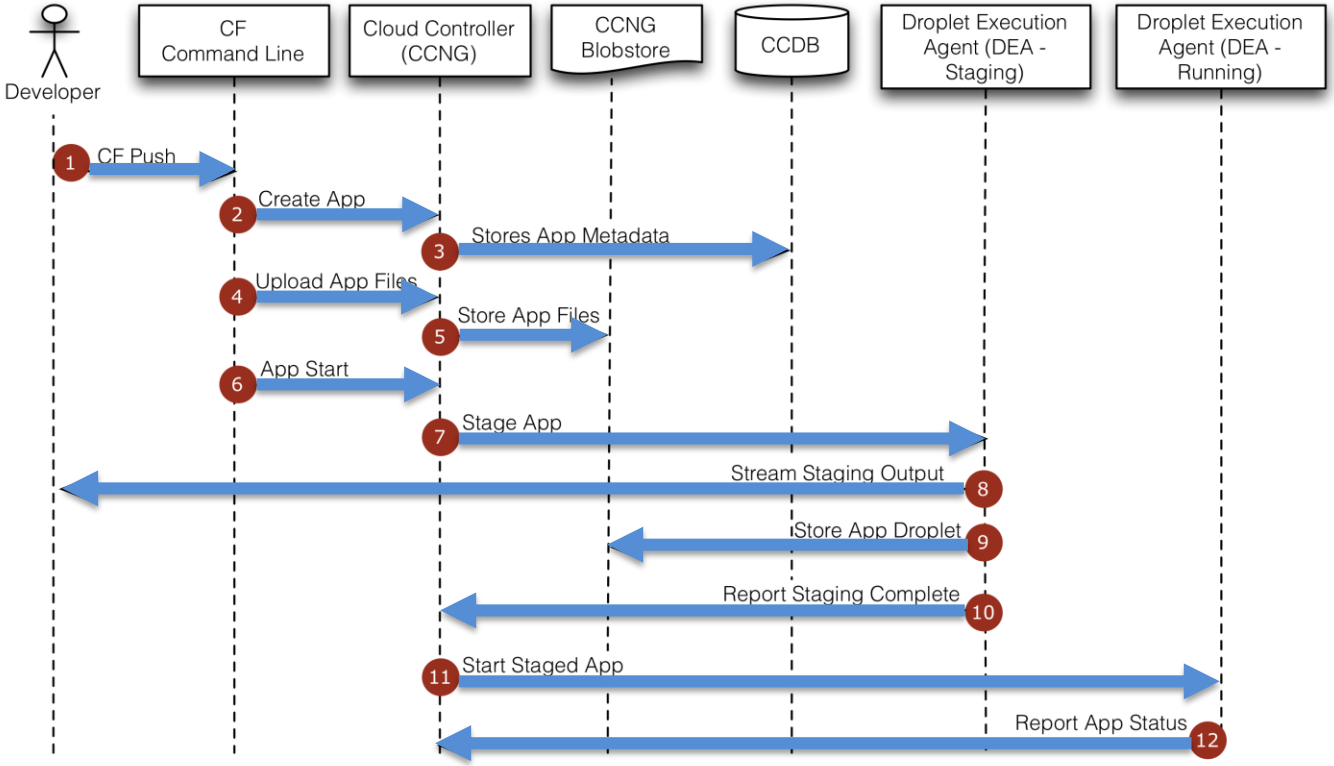
Benefits

- Automate the deployment
- Reproducibility
- Consistency

PS: CF Push command line parameters override manifest parameters

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

cf push



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

Environment Variables

CF Environment Variables

- Means by which CF runtime communicates with the application
- Environment variables are set by various CF components such as DEA, Buildpacks
- Developers can set application specific environment variables – referred to as user defined environment variables

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

Environment variables - Categories

- **Application** — Common to all instances of the app
- **Instance specific** — May be different for each instance of app
- **User Provided** — User provided environment variables
- **Environment Variable Groups** — Common across all apps/system

Application

- Common to all instances of the app - set by DEA & Buildpack during staging
  - HOME
  - MEMORY_LIMIT
  - PORT
  - PWD
  - TMPDIR
  - USER
  - VCAP_APP_HOST
  - VCAP_APPLICATION
  - ~~VCAP_APP_PORT~~ (Deprecated)
  - VCAP_SERVICES
- Buildpack may also specific variables e.g., JAVA_HOME

User Provided Env. Variables

- Application specific environment variable
- Can be set via manifest file e.g., application_stage=PROD

```
cf  set-env  APP_NAME      VAR_NAME      VAR_VALUE
Sets environment variable for the application

cf  unset-env  APP_NAME      VAR_NAME
UnSets environment variable for the application
```

Instance specific Env. Variables

- May be different for each instance of the same application
  - CF_INSTANCE_INDEX
  - CF_INSTANCE_IP
  - CF_INSTANCE_PORT
  - CF_INSTANCE_ADDR
  - CF_INSTANCE_PORTS

VCAP_APPLICATION

VCAP = *VMWare's Cloud Application Platform*

- In JSON format

GUID

host.domain

Resource limits
Defined by developer

Space name

```
{
 "VCAP_APPLICATION": {
  "application_id": "fa05c1a9-0fc1-4fbd-bae1-139850dec7a3",
  "application_name": "my-app",
  "application_uris": [
    "my-app.10.244.0.34.xip.io"
  ],
  "application_version": "fb8fbcc6-8d58-479e-bcc7-3b4ce5a7f0ca",
  "limits": {
  "disk": 1024,
  "fds": 16384,
  "mem": 256
  },
  "name": "my-app",
  "space_id": "06450c72-4669-4dc6-8096-45f9777db68a",
  "space_name": "my-space",
  "uris": [
    "my-app.10.244.0.34.xip.io"
  ],
  "users": null,
  "version": "fb8fbcc6-8d58-479e-bcc7-3b4ce5a7f0ca"
  }
```

VCAP_SERVICES

- For bindable services CF adds details to the VCAP_SERVICES in JSON format

No Route Applications

Worker apps

- Worker apps are processes that do not have a URL/route to it

  Used for background processing

  E.g., a worker app for sending email reminders to customers

- Created by defining a standalone program e.g., Java class with main()

  Pushed to cloud foundry with the *–no-route*

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

Summary

- Workers apps are background processes that do not have a URL/route to it

- Worker apps are created by pushing the app with cf push command with **–no-route option**

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