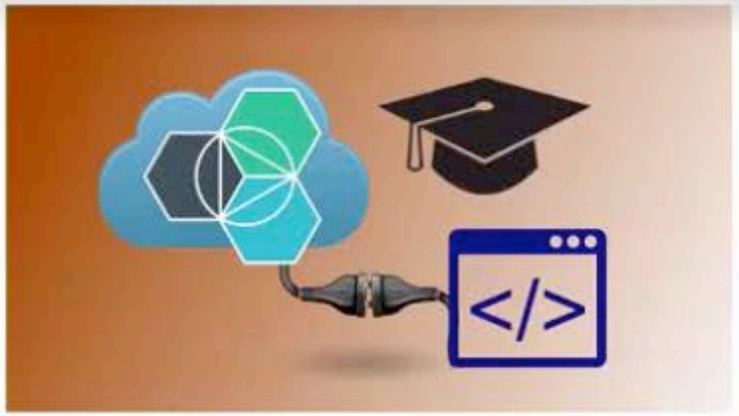
12 Factor App #5 - 8



Learning Objectives:

#5 Build, Release, Run

#6 Stateless Processes

#7 Port Binding

#8 Concurrency

#5 Build, Release & Run

"Strictly separate the build and run stages"

Build

- Compile code and package
  e.g., maven used to just create the war file
- One build many deploys

Release

- Droplet created by *cf push*
- War/Jar glued with whatever else is needed e.g., JDK for JAVA, Liberty for container , Tomcat

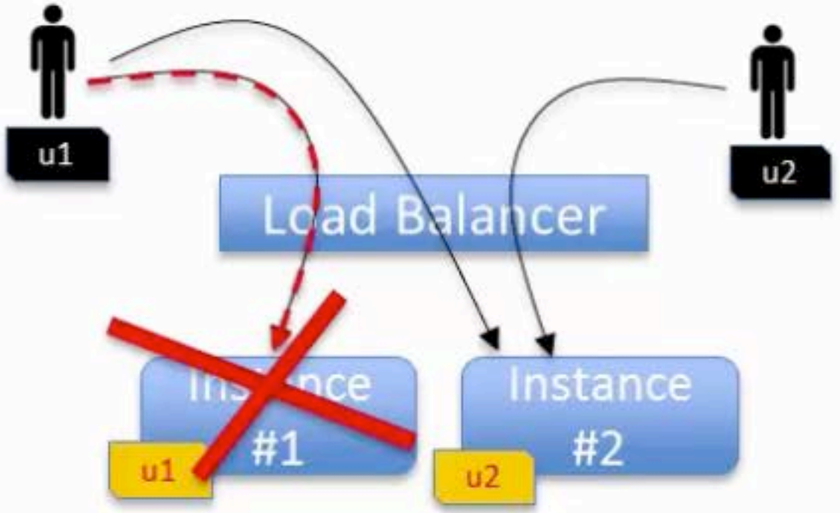
Run

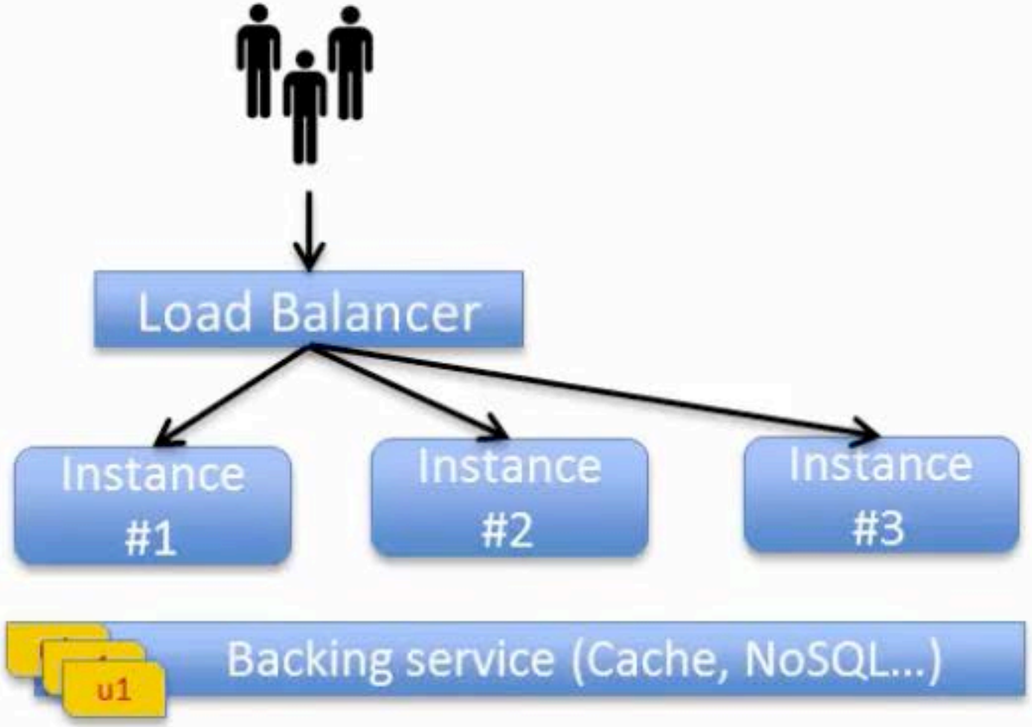
- Run using a single command *cf start*
- Container provides the runtime

#6 Processes

THE TWELVE-FACTOR APP

"Execute the apps as one or more stateless processes"





- No state info in memory or local file system
- State of the system is defined by the data is backing service such as a database

#7 Port Binding

"Export services via port binding"

- Expose the app like a self contained service with a URL
    - E.g., Java Springboot, Node JS are used to create self contained apps that is they do not need an external web container
- This way one app becomes the backing service for another app

#8 Concurrency

"Scale out via the process "

- An application process can benefit from vertical scaling up to a certain limit; at some point achieving higher levels of request processing concurrency is not possible
- The application should be designed in such as way that additional process instances may be created to cater to the increased traffic/load

Summary

- **Build, Release, Run** – strictly separate out the stages that convert the codebase to a running application process
- **Stateless Processes** – Apps should be built as stateless processes; use a backing service for managing state or session data
- **Port Biding**– application is self contained; listens for incoming requests on port bound to it
- **Concurrency** – application are built to use scale out strategy for scaling