Devops for Micro-services

Anupama Pradhan,
Ryan Hunt

15 million members

Operating Blue Cross and Blue Shield plans in **FIVE** states: IL, MT, NM, OK, TX

#6 on Diversity MBA's 50 Out Front for Diversity Leadership **Best Places to Work for Women & Diverse Managers**

Over **21,000** employees

+$1 billion in IT spend

OUR PURPOSE

To do everything in our power to stand with our members in sickness and in health®
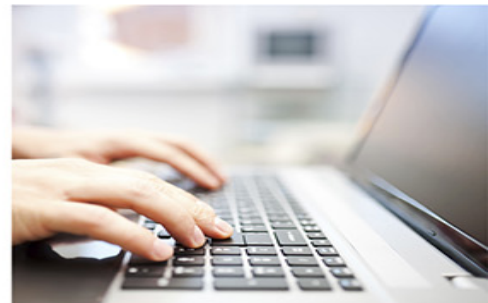
208.3 million claims processed annually

2,100 IT employees

LARGEST customer-owned health insurer in the U.S. and 4th largest overall

1936 year founded

What is CI/CD

Continuous Integration is the practice of testing each change done to your codebase automatically and as early as possible.

Continuous Deployment follows the testing that happens during Continuous Integration and pushes changes to a staging or production system. This makes sure a version of your code is accessible at all times.

Git → Build → Jar (binary) → Test → candidate → package → PCF
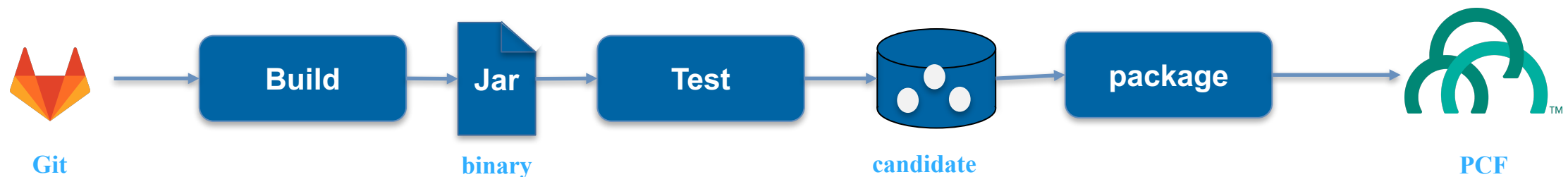
Concourse is a continuous integration/continuous delivery tool created by Pivotal. It uses pipelines to define a series of jobs that can execute the Terraform and Ansible scripts, perform Maven builds, among other things.

Ansible is an open source automation tool from RedHat, that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, etc

Docker is a customized container for developers and sysadmins to build, ship, and run distributed applications, whether on laptops, data center VMs, or the cloud.

Task: A task is a basic unit of work. It is execution of a command or script.

Resources: A resource is an entity that forms the input or output of a task

Jobs: A job is the configuration of tasks, input/output resources and execution criteria

**Pipeline:** Orchestration of multiple jobs form a pipeline.

Git → **Build** → **Jar**

Tasks: Tasks are basic ansible components that can manage system resources, like services, packages, files, execution of system commands, and etc..

Roles: reusable modules based on the well defined structure.

Inventory: Collection of servers and variables

Playbook: playbooks orchestrate the set of steps required to automate processes
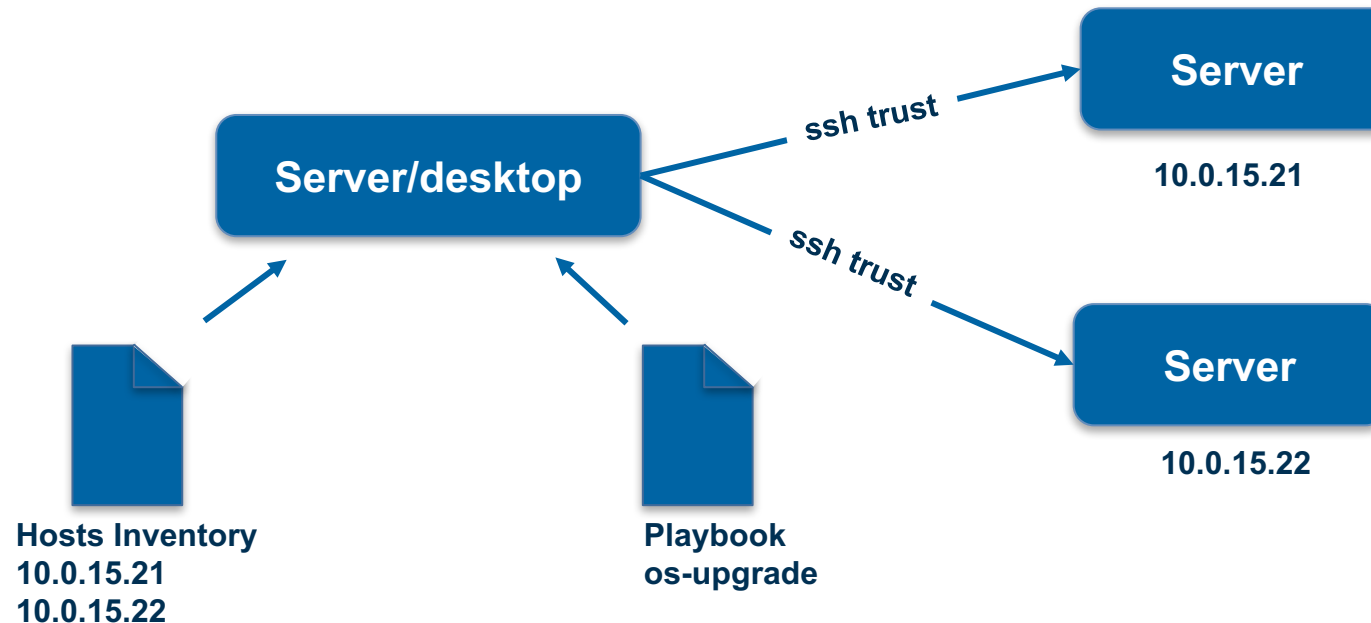
ApplicationName-<major>.<minor>.<patch>.<type>

major: Indicates significant change in the application

- Major change that is not compatible with prior versions
- Large set of new features

minor: Indicates new release with minor enhancements, bug fixes that do not
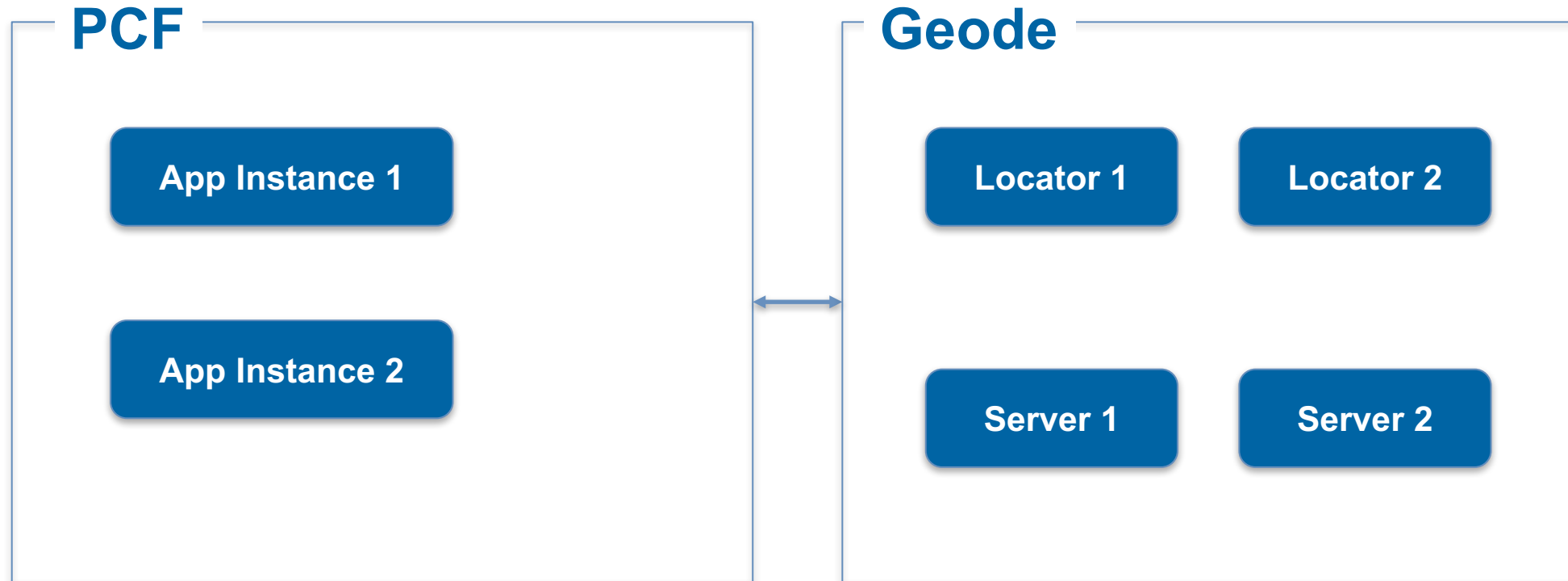
impact the backward compatibility

patch: bug fixes, no new functionality

type: keyword that represents the stability of the release

For Example: snapshot, RC (release candidates), Final

Jar movement along the environments

Semantic Version	Nexus Archive	DEV	S3 Archive (DEV)	TEST	S3 Archive (TEST)	PROD	Nexus Archive
1.0.0-SNAPSHOT.1	x	FAIL					
1.0.0-SNAPSHOT.2	x	PASS	x	FAIL			
1.0.0-SNAPSHOT.3	x	PASS	x	PASS	x	PASS	x
Version Bump -> 1.1.0							
1.0.1-SNAPSHOT.1	x	PASS	x	PASS	x	PASS	x
Version Bump -> 1.2.0							

Sample Application : Customer Order Service

PCF

App Instance 1

App Instance 2

Geode

Locator 1

Locator 2

Server 1

Server 2

Pipeline Jobs:

provision: Prepare the environment like install jdk and geode binaries.

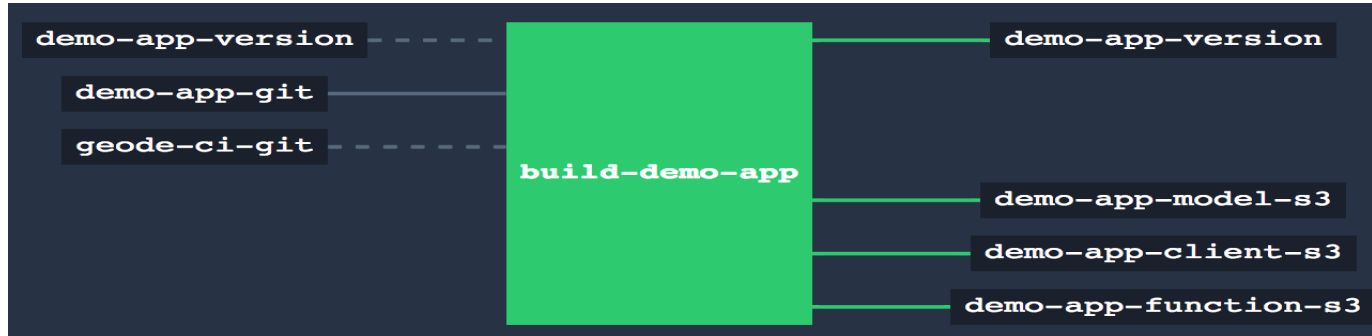This task is reused for upgrade too.

configure: setup pdx packages, create regions, setup base configuration for

locators and servers

deploy: Build application code to Gemfire server– Model, functions etc.

deploy-client: Deploy Client code to PCF (API interface)

Demo Pipeline Jobs

Build and deploy the microservice

Provision Geode

Configure Geode cluster for the microservice

- Example for Geode Upgrade

References

https://pivotal.io/concourse

https://www.ansible.com/it-automation

http://geode.apache.org/docs/guide/13/about_geode.html

https://geode.apache.org/docs/guide/11/configuring/cluster_config/gfsh_persist.html

https://www.youtube.com/watch?v=bAfEJtUVors&index=3&list=PL62pIycqXx-QfmNrUmfKoTZXKU5K90JpK

Source Code

- https://github.com/ap1256/gemfire-demoapp