



Cognizant

Spinnaker – CD Tool for K8's



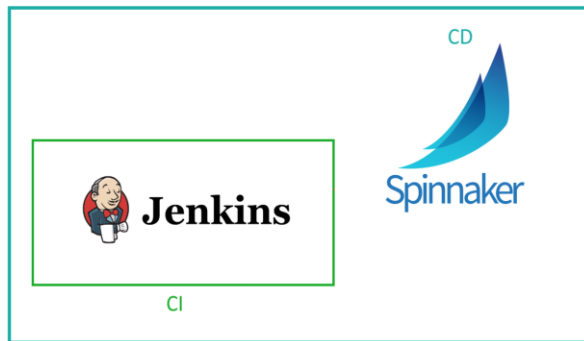
Presenter: Syed Abdul Raqib - 241026

Date: 08/03/2018

What is CI and CD?

Continuous Integration refers to the process of verifying the build, allowing teams to detect problems early and thereafter creating a container image automatically

Continuous Delivery is the extension of Continuous Integration: an approach in which teams ensure that every change to the system is releasable, and that we can release any version at the push of a button



Why Spinnaker?



Canary: Supports phased, canary deployments. Deploy to a percentage of traffic or to specific regions, and easily change your canary approach on the fly.



Roll Back: Supports “Red/Black” deployments, enabling fast roll-backs to previous application versions if needed. You define how long you want to keep previous versions of applications available, and how many past versions to keep as backups.



Multi-Cloud: Users can manage resources using cloud-agnostic constructs, abstracting deployments to IaaS providers like AWS, GCP, and Azure.



Visibility: Provides UI visibility to the Infrastructure where the application is deployed.



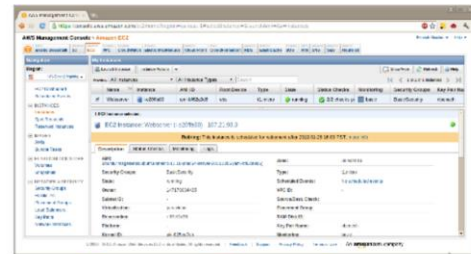
Kubernetes Support: Enables more sophisticated deployments to Kubernetes.



Flexible: Easy creation and modification of deployment pipelines (including complex pipelines) using a visual interface (or via programmatic endpoints).

Before and After

Before Spinnaker

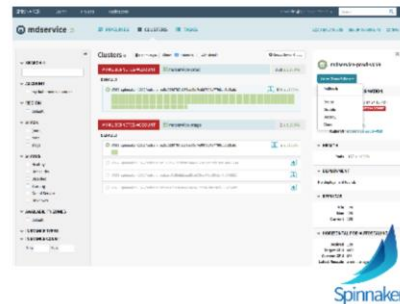
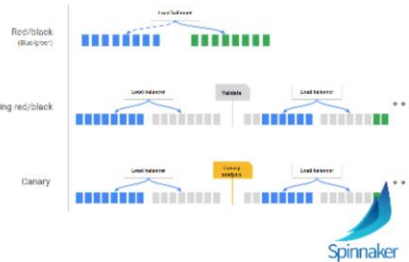
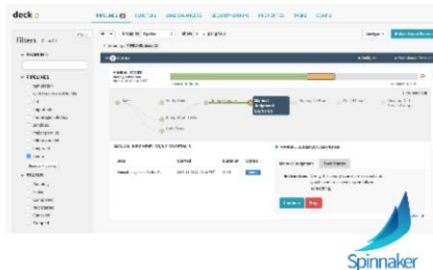


Compile/Prepare Image

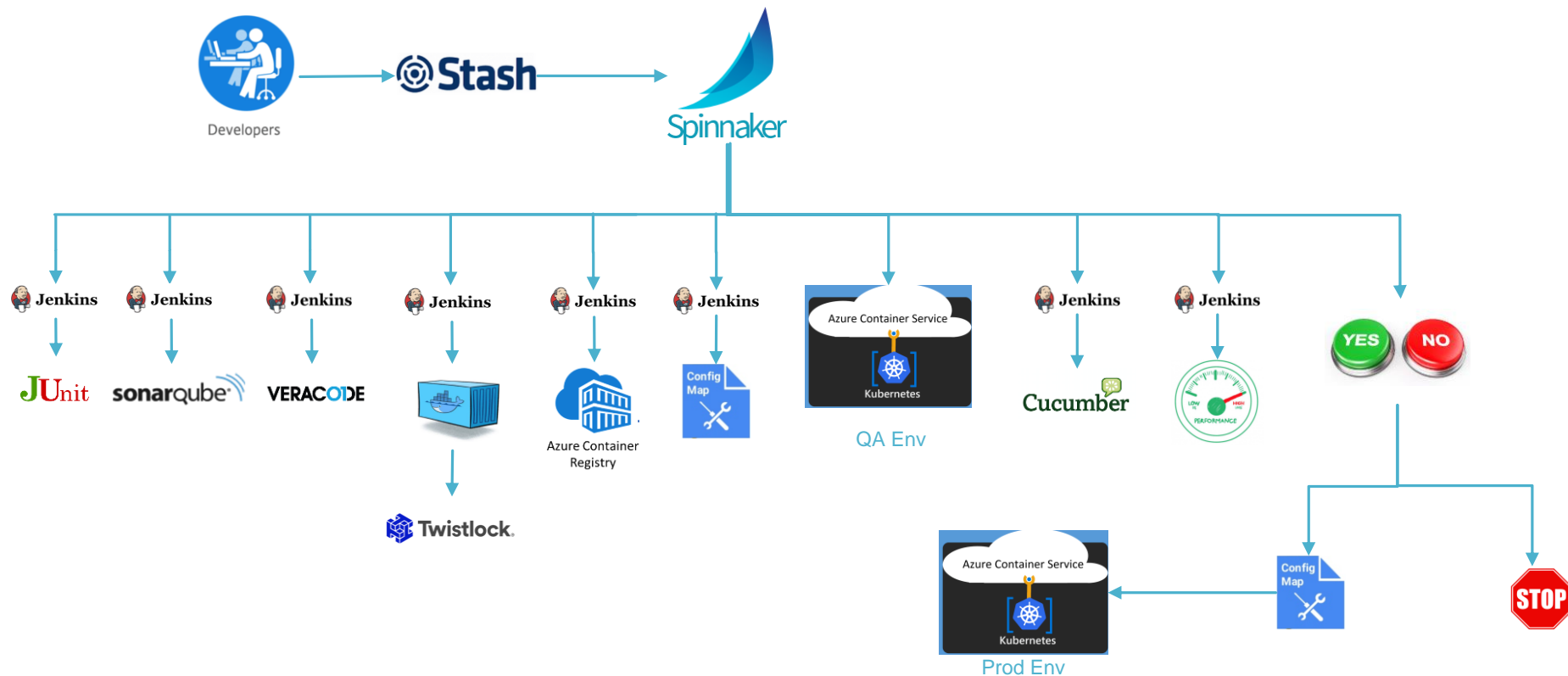
Deploy/Provision

VPC/Monitor Cloud status

With Spinnaker



Spinnaker with Jenkins

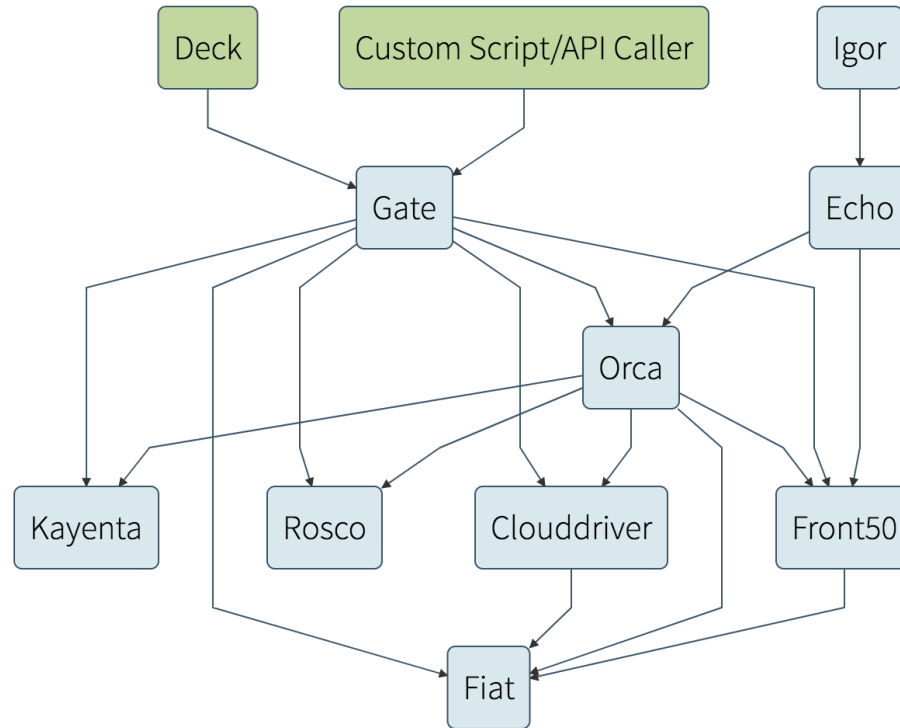


Jenkins Stages

[Enlarge](#) [Configure](#)

		Clone	Unit Test - JavaScript	SonarQube Analysis - JavaScript	Unit Test - Java	SonarQube Analysis - Java	Build - Java	Docker Build	TwistLock Scan	Push to ACR
Average stage times: (Average full run time: ~10min 43s)		4min 11s	32s	1min 46s	1min 39s	39s	19s	37s	14s	15s
#62	Jul 13 02:10 No Changes	2min 47s	31s	1min 16s	1min 20s	33s	18s	39s	13s	15s
#61	Jul 13 01:58 No Changes	4min 2s	33s	59s	1min 34s	33s	20s	42s	17s	18s
#60	Jul 11 21:04 No Changes	2min 51s	30s	1min 29s	1min 39s	33s	17s	39s	15s	15s
#59	Jul 11 10:08 No Changes	3min 18s	33s	1min 49s	1min 51s	1min 19s	20s	43s	26s	16s

Spinnaker Architecture



Spinnaker Architecture - Services

Spinnaker is composed of a number of independent micro-services

- [Deck](#) is the browser-based UI.
- [Gate](#) is the API gateway.
- The Spinnaker UI and all api callers communicate with Spinnaker via Gate.
- [Orca](#) is the orchestration engine. It handles all ad-hoc operations and pipelines.
- [Clouddriver](#) is responsible for all mutating calls to the cloud providers and for indexing/caching all deployed resources.
- [Front50](#) is used to persist the metadata of applications, pipelines, projects and notifications.
- [Rosco](#) is the bakery.
- It is used to produce machine images (for example [GCE images](#), [AWS AMIs](#), [Azure VM images](#)). It currently wraps [packer](#), but will be expanded to support additional mechanisms for producing images.
- [Igor](#) is used to trigger pipelines via continuous integration jobs in systems like Jenkins and Travis CI, and it allows Jenkins/Travis stages to be used in pipelines.
- [Echo](#) is Spinnaker's eventing bus.
- It supports sending notifications (e.g. Slack, email, Hipchat, SMS), and acts on incoming webhooks from services like Github.
- [Fiat](#) is Spinnaker's authorization service.
- It is used to query a user's access permissions for accounts, applications and service accounts.
- [Kayenta](#) provides automated canary analysis for Spinnaker.
- [Halyard](#) is Spinnaker's configuration service.
- Halyard manages the lifecycle of each of the above services. It only interacts with these services during Spinnaker startup, updates, and rollbacks.

Spinnaker Main Page

cure | dgtlspinnaker02.walgreens.com:9000/#/search ☆

SPINNAKER Search Projects Applications → List all the Project and Application in Spinnaker sraqibh3 Search What's New

Search projects, applications, clusters, load balancers, server groups, security groups → Search for Particular Project or Application Actions ▾

Recently viewed

APPLICATIONS (5)

- cactender
- rxdpfrxstatus
- doctext2clip
- caccartui ✕
- rxdprrxcheckout

INSTANCES (4)

- cactender-v004-gp5jl (dcom-dev-spinnaker)
- csfamilymgmt-v003-p9trk (dcom-dev-spinnaker)
- caccartui-v003-5n8g9 (dcom-dev-spinnaker)
- caccartui-v000-h8rmd (dcom-dev-spinnaker)

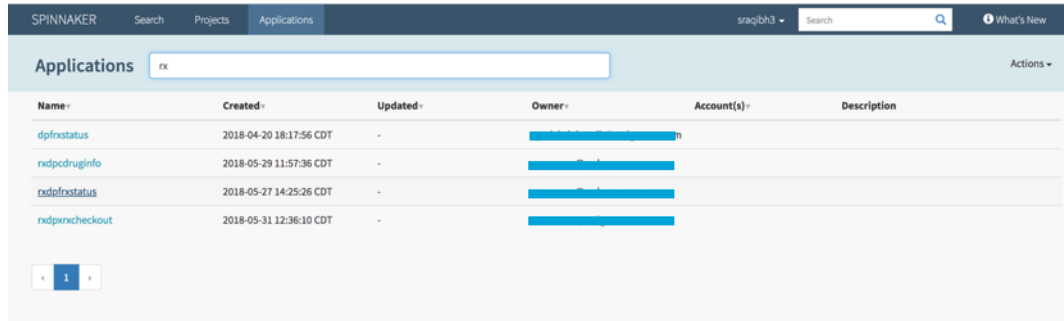
SERVER GROUPS (5)

- DCOMDEV03 rxdpfrxstatus-v014 (dcom-dev-spinnaker)
- DCOMDEV03 rxdpfrxstatus-v012 (dcom-dev-spinnaker)
- DCOMDEV03 rxdpfrxstatus-v013 (dcom-dev-spinnaker)
- DCOMDEV03 rxdpfrxstatus-v011 (dcom-dev-spinnaker)
- DCOMDEV03 rxdpfrxstatus-v010 (dcom-dev-spinnaker)

→ Recently Accessed Application, Instances and Server Groups

- To Browse all the Project and Application
- To Search Application, Project
- Body shows the recent project and application

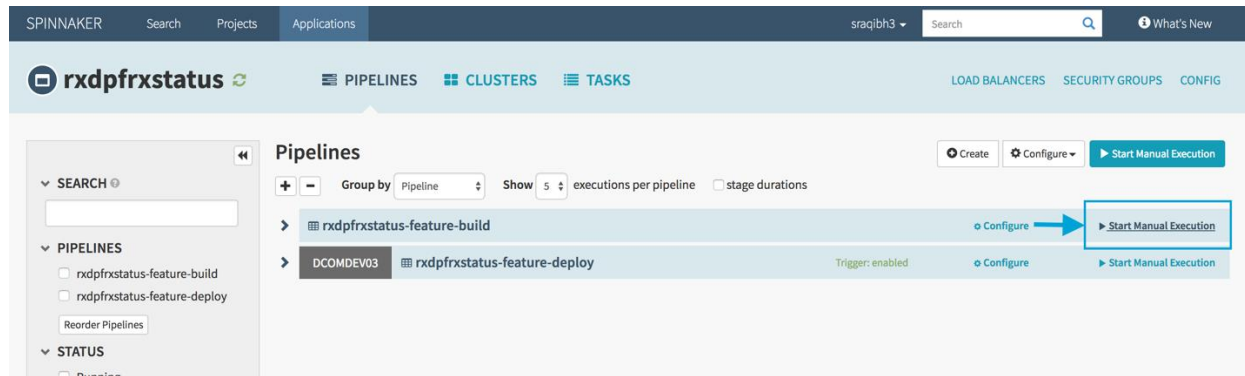
Access Application, Pipeline, Check POD instance



Name	Created	Updated	Owner	Account(s)	Description
dprfxstatus	2018-04-20 18:17:56 CDT	-	[REDACTED]		
rxdpcdruginfo	2018-05-29 11:57:36 CDT	-	[REDACTED]		
rxdprfxstatus	2018-05-27 14:25:26 CDT	-	[REDACTED]		
rxdprfxcheckout	2018-05-31 12:36:10 CDT	-	[REDACTED]		

Click on application name, the default page is clusters – click on pipelines to access them

After logging in -> Search for an application -> eg : rx . This will list the applications with rx



Pipelines

Group by Pipeline Show 5 executions per pipeline stage durations

- rxdprfxstatus-feature-build Configure Start Manual Execution
- DCOMDEV03 rxdprfxstatus-feature-deploy Trigger: enabled Configure Start Manual Execution

Spinnaker will trigger a Jenkins job in the back end

SPINNAKER Search Projects Applications sraqibh3 Search What's New

rxdpfrxstatus PIPELINES CLUSTERS TASKS LOAD BALANCERS SECURITY GROUPS CONFIG

Pipelines

Create Configure Start Manual Execution

Group by Pipeline Show 5 executions per pipeline stage durations

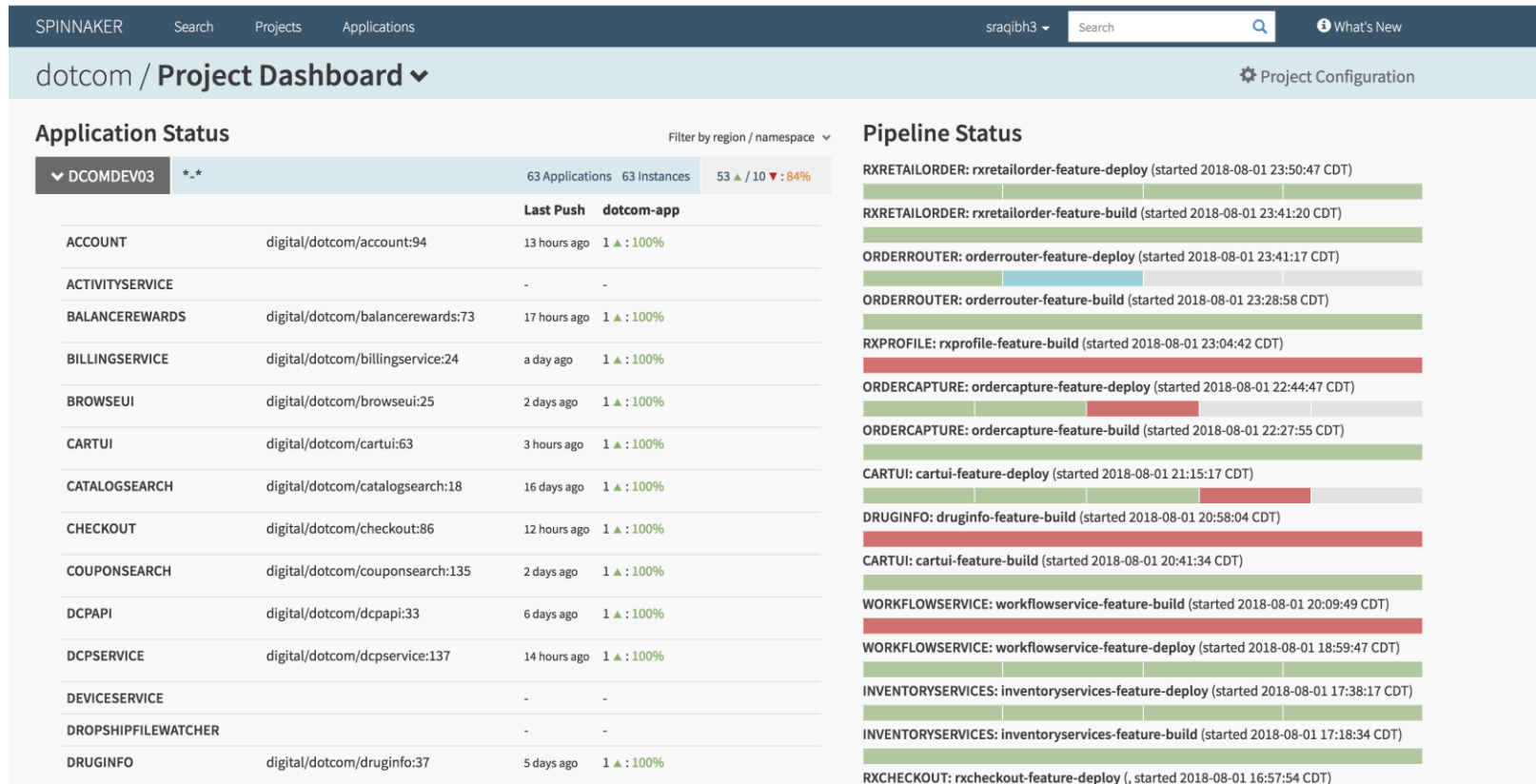
- rxdpfrxstatus-feature-build** 1 Configure Start Manual Execution
 - MANUAL START**
sraqibh3
a few seconds ago
FeatureBranch: "feature/enhancemen
t"
Status: **RUNNING** Duration: 00:05
[Details](#)
 - MANUAL START**
saruna4p
3 days ago
FeatureBranch: "feature/dotcom-rxsta
tus"
Status: **TERMINAL** Duration: 06:04
[Details](#)
 - MANUAL START**
saruna4p
5 days ago
FeatureBranch: "master"
Status: **TERMINAL** Duration: 05:23
[Details](#)
- DCOMDEV03 rxdpfrxstatus-feature-deploy** Trigger: enabled Configure Start Manual Execution

Deployment UI

The screenshot shows the Spinnaker Deployment UI. The top navigation bar includes 'SPINNAKER', 'Search', 'Projects', 'Applications', and a search bar. The 'CLUSTERS' tab is selected and highlighted with a blue box and an arrow. Below the navigation bar, the 'Clusters' section is visible, showing a list of clusters. The cluster 'DCOMDEV03' is selected, and its instances are listed. The instance 'rdpfrxstatus-v020-vc4mf' is highlighted with a blue box and an arrow. On the right side of the UI, the details for this instance are shown, including its status (Up) and a link to 'Console Output (Raw)' which is also highlighted with a blue box and an arrow.

When the Deployment completes click on clusters -> POD instance to check if the instance is up, will be indicated by red if it is down, you will be able to see the POD instance information, images, console output

Project Dashboard



K8 Deployment with Spinnaker



Questions?



Know more:

<https://www.spinnaker.io/>

<https://www.spinnaker.io/community/faqs/>

<https://www.armory.io/>



Cognizant

KEEP CHALLENGING™