

Continuous Deployment into the Unknown with Artifactory, Bintray, Docker and Mesos

Gilad Garon
Kiril Nesenko

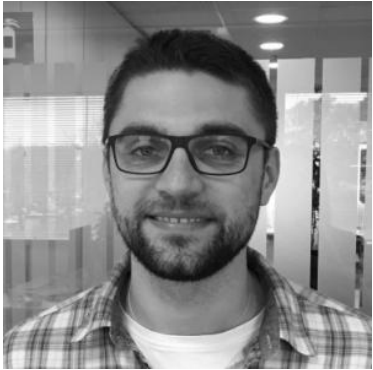
vmware®

© 2015 VMware Inc. All rights reserved.

Agenda

- What is the Common SaaS Platform (CSP)
- CI/CD processes for CSP
- Upgrading CSP
- Xenon - Distributed Control Plane (If we have the time)

Who are we ?



Kiril Nesenko

DevOps Lead

knesenko@vmware.com



Gilad Garon

Architect

ggaron@vmware.com , *Twitter @giladgaron*

VMware's SaaS Transition

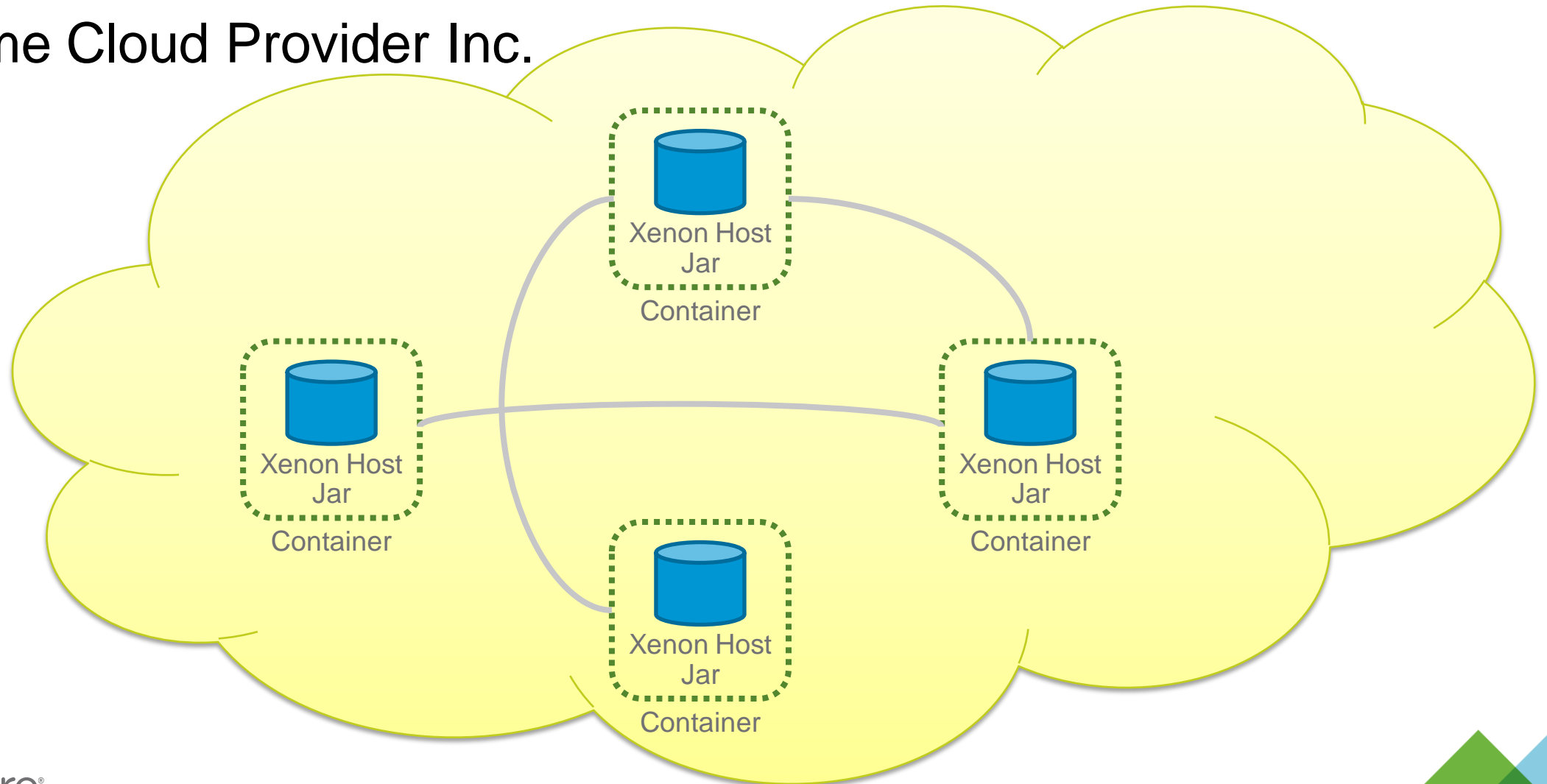
- VMware is developing many SaaS offerings
- Many services have the same common requirements (Billing, Identity, etc.)
- Like other good engineers, we like to reuse code wherever possible
- VMware's Common SaaS Platform (CSP) is platform that internal SaaS offerings are using to leverage existing internal components

Designing a SaaS platform

Design Principles	In Practice
Cloud Agnostic	Infrastructure needs to support containers
Highly Available	Dynamic, Stateful and Distributed cluster
Scalable	Tunable consistency helps to achieve availability & scalability
Great Public APIs	No internal APIs
Modular	Capabilities as libraries, Coupling is done with APIs
Ease of operability / development	Single JAR, limited classpath dependencies set

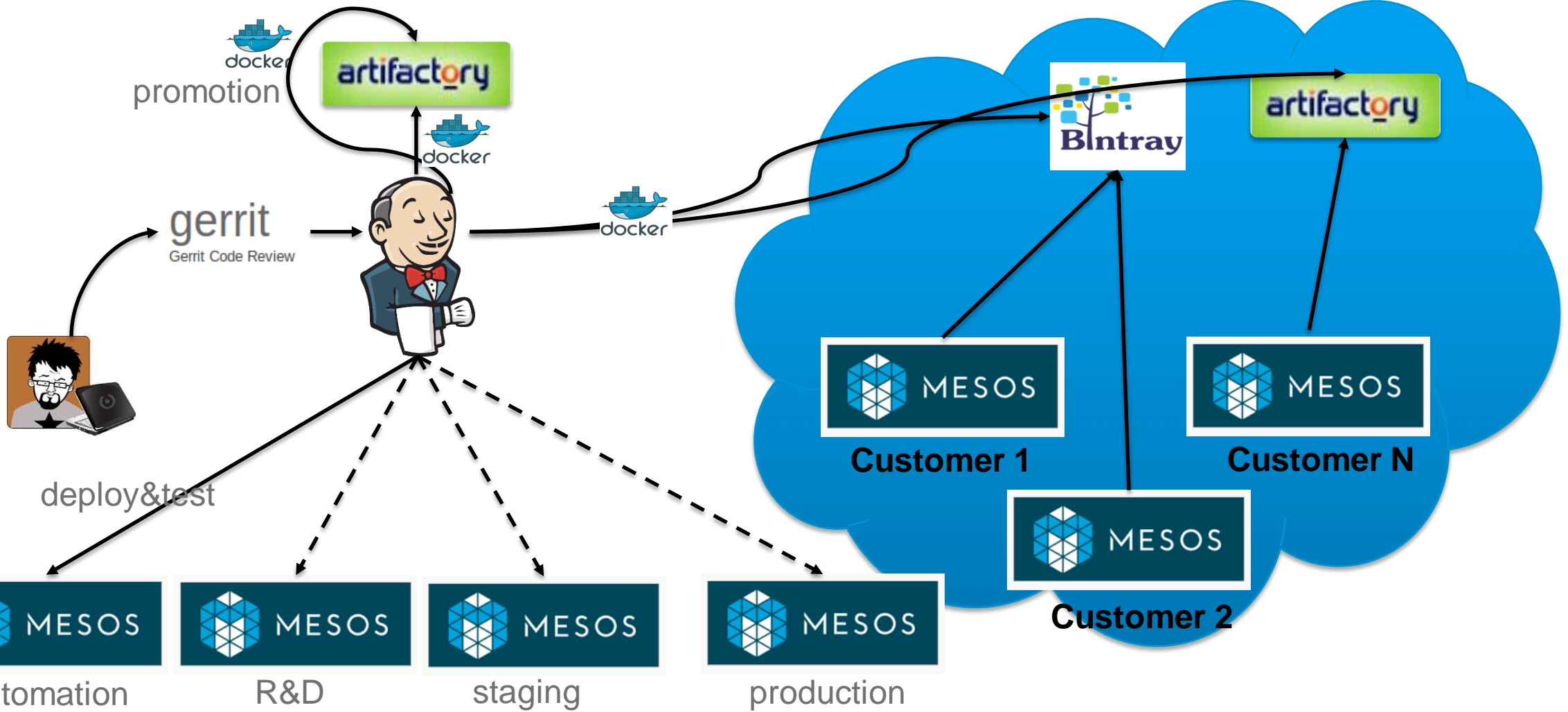
Deployment Architecture. yep that's it.

Some Cloud Provider Inc.

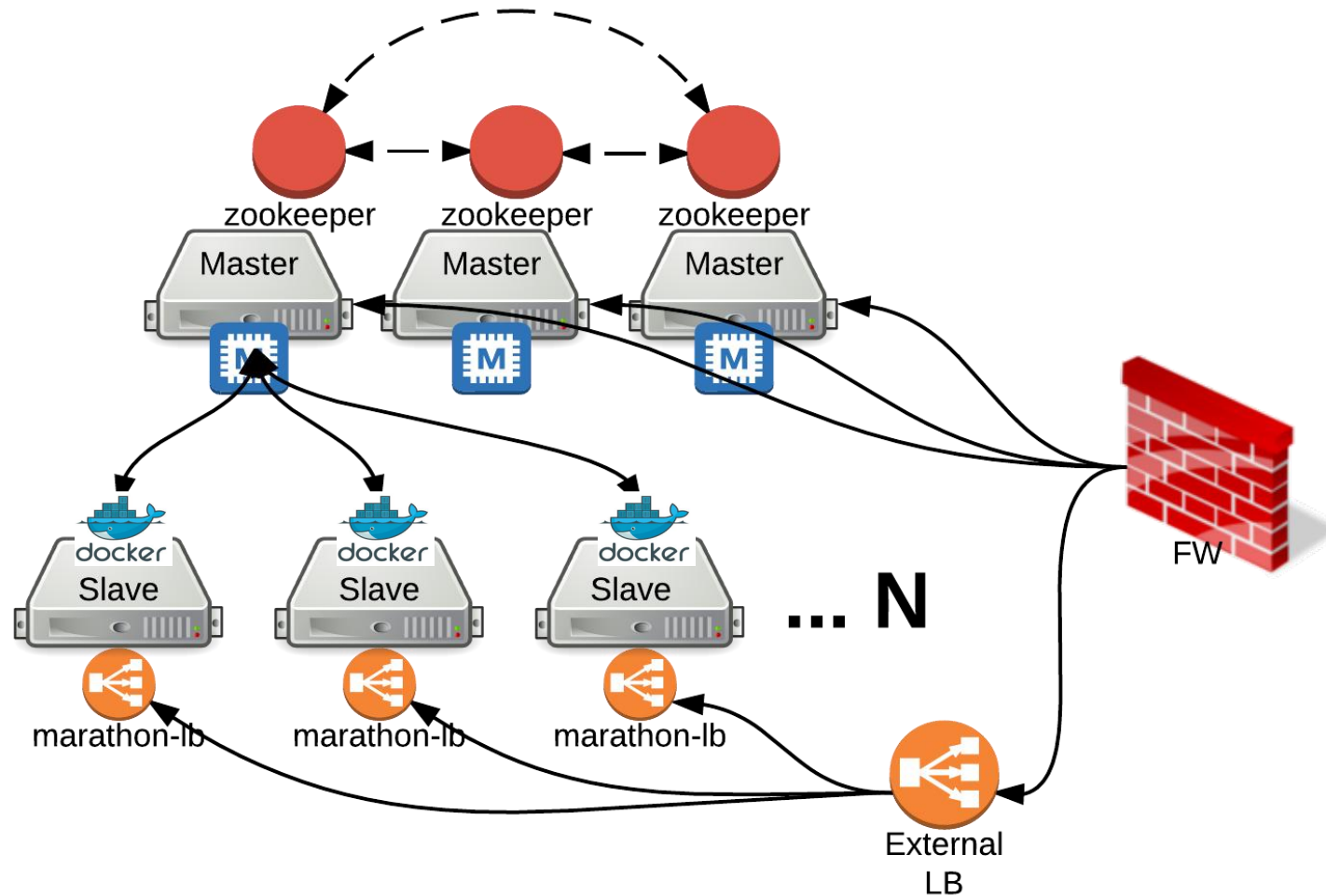


Infrastructure and Patch Life Cycle

CI/CD Overview

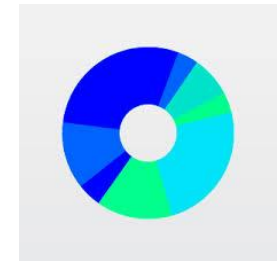


CSP Mesos Infrastructure



CI/CD Tools

- Artifacts: Artifactory, Bintray
- CI: Jenkins
- Source Control: git
- Code review: gerrit
- Slaves: dockers
- Infrastructure: mesos, dockers
- Code Analysis: Sonar
- Build: gradle, Makefiles
- Languages: Java, JS, Python, Go
- Communication: Slack



CI Infrastructure

- ~300 jenkins jobs
- 20 git repositories
- On the fly jenkins slaves
- Jenkins and Slack integration
- Mesos cluster (Marathon, marathon-lb, mesos-dns, Calico, chronos)



Jenkins Jobs Management

Jenkins Job Builder



Jenkins job builder to the rescue!

Jenkins Job Builder

- Developed by OpenStack folks
- Configuration as code (yaml format)
- Easy to review changes
- Configuration de-duplication
- Include shell/groovy/python... scripts
- Test before deploying
- Easier to organize (per directory, per file)
- Serves as backup (easy to replicate to another jenkins)

```
job:
  name: docker_clean_images_containers
  concurrent: true
  parameters:
    - gerrit-params:
      branch: master
    - bool:
      name: CLEAN_IMAGES
      default: true
      description: "Clean <none> images"
    - bool:
      name: CLEAN_CONTAINERS
      default: true
      description: "Clean old containers"
    - string:
      name: REMOVE_IMAGE
      default: ''
      description: "Image name|tag to remove"
    - node:
      name: NODES
      description: "Select slave"
      default-slaves:
        - csp-docker
        - csp-docker-01
      allowed-multiselect: true
  triggers:
    - timed: "H 0 * * *"
  scm:
    - csp-cd-gerrit
  builders:
    - shell: !include-raw: shell-scripts/docker_clean_containers_wrapper.sh
```

Project docker_clean_images_containers

This build requires parameters:

GERRIT_REFSPEC	<input type="text" value="refs/heads/master"/>
	Git refspec to build
GERRIT_BRANCH	<input type="text" value="origin/master"/>
	Git branch to build
CLEAN_IMAGES	<input checked="" type="checkbox"/>
	Clean images
CLEAN_CONTAINERS	<input checked="" type="checkbox"/>
	Clean old containers
REMOVE_IMAGE	<input type="text"/>
	Image name/tag to remove
NODES	<input type="text" value="master"/> <input type="text" value="csp-docker"/> <input type="text" value="csp-docker-01"/>
	Select slave

Build

Templates

- For nearly identical jobs better to use templates

```
job-template:  
  name: '{build-project}_build'  
  node: docker-slave  
  properties:  
    - throttle:  
      enabled: true  
      max-per-node: 1  
      max-total: 3  
      option: project  
  parameters:  
    - gerrit-params:  
      branch: '{branch}'  
    - string:  
      name: GERRIT_EVENT_ACCOUNT_EMAIL  
      default: ''  
      description: 'Email of the person who triggered the build'  
  scm:  
    - csp-cd-master  
    - '{build-project}-gerrit'  
  builders:  
    - gradle:  
      gradle-name: 'gradle-docker'  
      wrapper: false  
      use-root-dir: false  
      root-build-script-dir: '{root-build-script-dir}'  
      switches:  
        - '{switches}'  
      tasks: |  
        {tasks}
```

Templates

```
project:
  name: csp-account-management_build
  branch: master
  build-project: csp-account-management
  root-build-script-dir: '${workspace}/csp-account-management/'
  tasks: |
    clean
    build
    sonarRunner
    artifactoryPublish
  switches: '--refresh-dependencies'
  jobs:
    - '{build-project}_build'
```

Jobs Update

```
2016-04-28 12:34:22.474 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_csp-telemetry_build
2016-04-28 12:34:23.417 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_csp-ui_build
2016-04-28 12:34:24.369 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_ars_base_build
2016-04-28 12:34:26.032 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_ars_build
2016-04-28 12:34:27.713 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_chronos_build
2016-04-28 12:34:29.621 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_collectd_build
2016-04-28 12:34:30.688 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-certs_build
2016-04-28 12:34:32.370 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-jenkins-ansible_build
2016-04-28 12:34:34.146 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-jenkins-base_build
2016-04-28 12:34:36.002 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-jenkins-slave-django_build
2016-04-28 12:34:37.048 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-jenkins-slave-go_build
2016-04-28 12:34:38.096 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-jenkins-slave_build
2016-04-28 12:34:39.173 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-marathon_build
2016-04-28 12:34:40.111 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-mesos-base_build
2016-04-28 12:34:41.306 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-mesos-master_build
2016-04-28 12:34:42.741 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-mesos-slave_build
2016-04-28 12:34:43.819 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-photon-base_build
2016-04-28 12:34:44.871 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_csp-zookeeper_build
2016-04-28 12:34:47.122 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_docker_mesos-dns_build
2016-04-28 12:34:48.706 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job listener_gerrit-admin_rpm
2016-04-28 12:34:50.610 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job ntp_sync
2016-04-28 12:34:52.405 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job osstpmgt-gradle-plugin_build
2016-04-28 12:34:53.453 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job update_ars
2016-04-28 12:34:54.497 | INFO:jenkins_jobs.builder:Reconfiguring jenkins job upload_docker_image
```



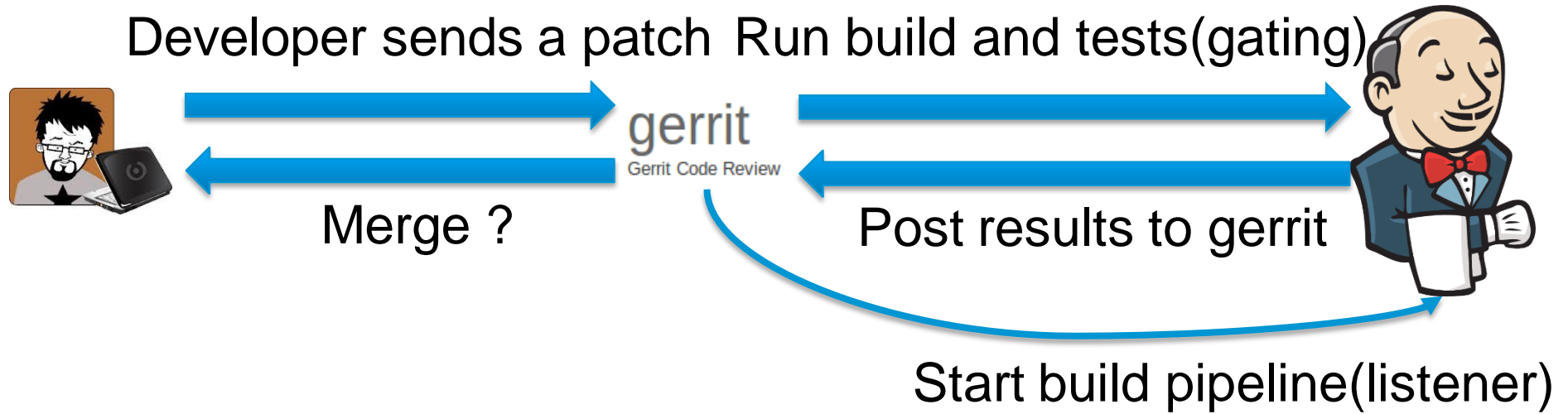
Jenkins Jobs Types

- Gating – listens for patch-set-created events
- Build – for building purposes (gradle, docker etc)
- Listeners – listens for change-merged events on gerrit (orchestrators for the pipelines)

Gating Jobs

- For each patch we run a gating job
- Each git project has its own gating job
- Build + test + post results to gerrit

Gating Jobs



Gerrit

- web-based code review tool built on top of the git

```
csp-cd / jenkins/confs/yaml/jobs/builds/build/csp-host_build.yaml  
Patch Set Base 1 2 ... skipped 7 common lines ...  
8     max-total: 3  
9     option: project  
10    - ownership:  
11      owner: knesenko  
12      co-owners:  
13        - dolininal  
14    parameters:  
15      - gerrit-params:  
16        branch: master  
17      - string:  
  
18      name: GERRIT_CHANGE_ID  
19      default: ''  
20      description: 'Gerrit change id'  
  
Patch Set 1 2 ... skipped 7 common lines ...  
8     max-total: 3  
9     option: project  
10    - ownership:  
11      owner: knesenko  
12      co-owners:  
13        - dolininal  
14    parameters:  
15      - gerrit-params:  
16        branch: master  
17      - string:  
18      name: GERRIT_EVENT_ACCOUNT_EMAIL  
19      default: ''  
20      description: 'Email of the person who triggered the build'  
21    - string:  
22      name: GERRIT_CHANGE_ID  
23      default: ''  
24      description: 'Gerrit change id'
```

Jenkins Failure

All Projects Documentation

Open Merged Abandoned

Change **1503** - Not Verified

Reply...

csp-am: Add Json Web Token to authentication - AM side

Changes:

1. Fix cookie in UserLoginService
2. Add UserLoginService to AM bootstrap
3. AuthenticationInterceptor to support JWT
4. getLoggedInUser to user link instead of token

Change-Id: Ia2879fb16dfd8539893140343045606363967ecb

Owner [Sharon Harel](#)
Reviewers [CSP Gerrit](#) [CSP Jenkins](#)
Project [csp-account-management](#)
Branch [master](#)
Topic [JWT](#)
Strategy [Rebase if Necessary](#)
Updated 28 hours ago

Code-Review +1 [CSP Gerrit](#)

Verified -1 [CSP Jenkins](#)

Sonar Failure

Change 1502 - Needs Code-Review

csp-common: Swagger to support property inheritance

Add support for inherited @ApiModelPropertyProperties.
Inherited annotated properties will appear
in the swagger documentation of the model
that contains them.

Change-Id: I0c7b80ef7951901f1eaf0fd721525e1a5a36eafb

Reply...

Owner [Doron Goldgevicht](#)

Reviewers [CSP Gerrit](#) [CSP Jenkins](#) [CSP Sonar](#)

Project [csp-common](#)

Branch [master](#)

Topic

Strategy [Rebase if Necessary](#)

Updated 28 hours ago

Code-Review -1 [CSP Sonar](#)

+1 [CSP Gerrit](#)

Verified +1 [CSP Jenkins](#)

Gerrit Failure

Gerrit hooks

- Executed on the server side
- Execute per event type
- Various checks: commit message style, trailing white spaces, etc.
- Integrations with external systems: bugzilla, jira, etc.

CSP Gerrit

Patch Set 1: Code-Review-1

- `patchset-created-all.1::ERROR::please remove trailing whitespaces`
- `patchset-created-all.1::INFO::for more info run: git show HEAD --check`
- `patchset-created-all.1::INFO::for autofix run: git rebase --whitespace=fix`

Dynamic Pipelines

Listener Jobs

- Executed on patch-merged event
- Orchestrating the build and delivery pipeline dynamically
- Orchestration done via the BuildFlow plugin (groovy)
- All listeners run the same code base
- On failure, user is notified on slack channel



jenkins BOT 07:15

@isomech seems like you broke the build

Job:https://csp-jenkins.eng.vmware.com/job/csp-host_build/687/

Please fix ASAP



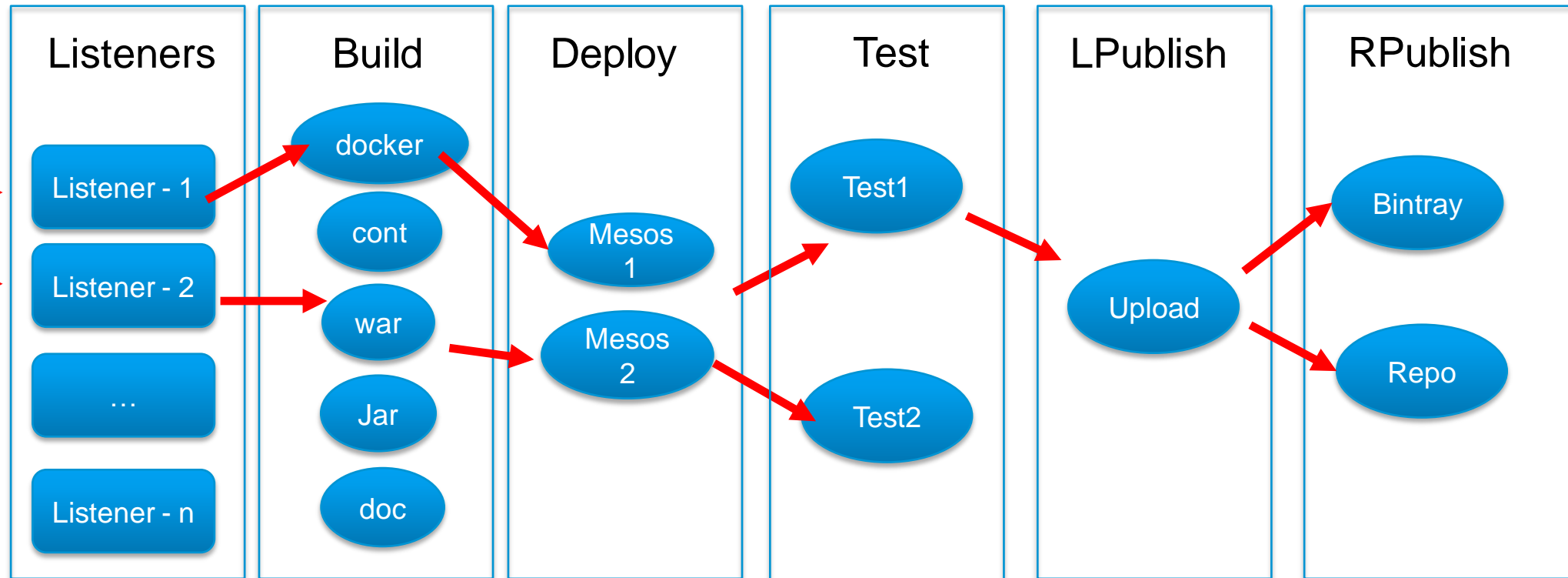
Kiril Nesenko 09:45

@isomech: ^^ FIX !!

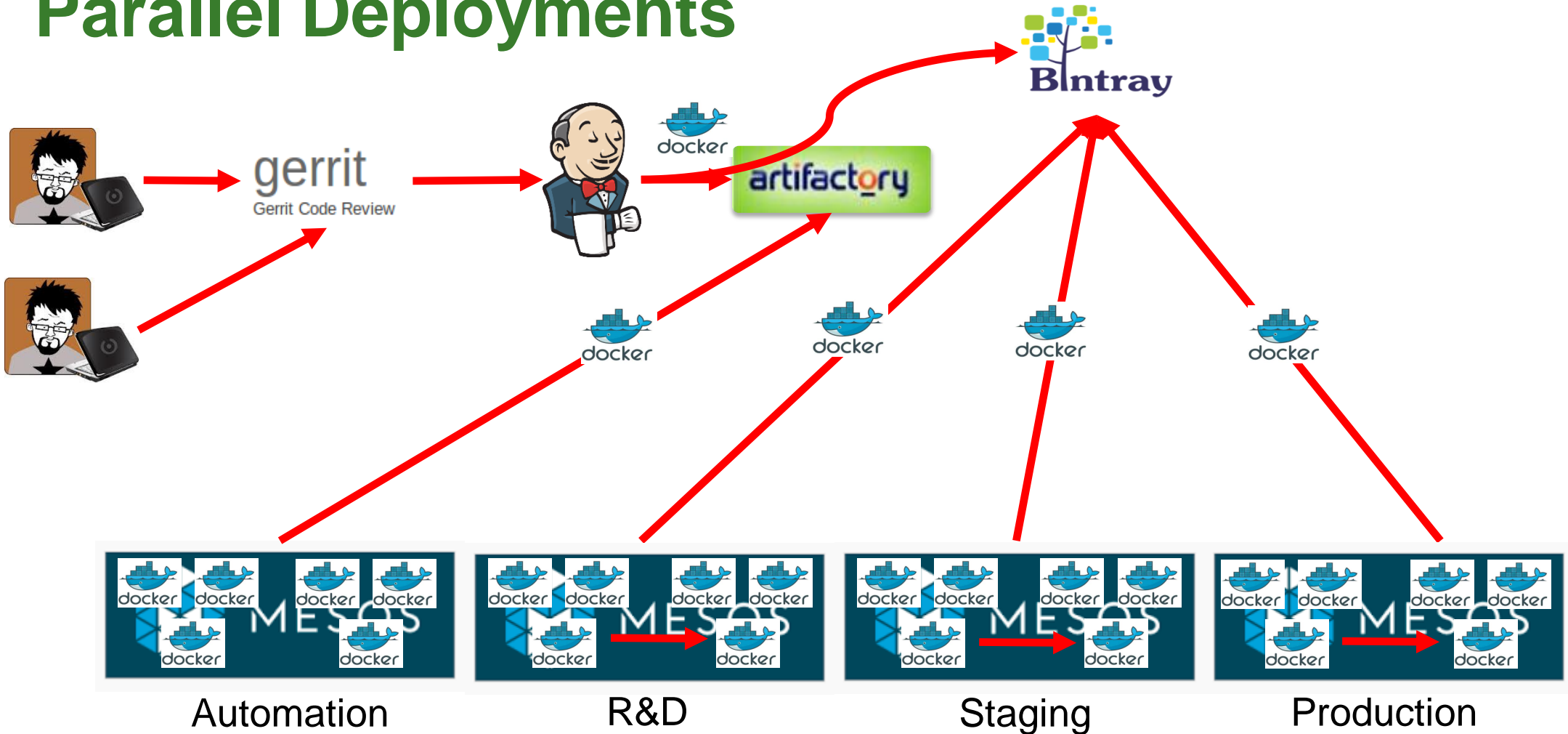
Dynamic Flows

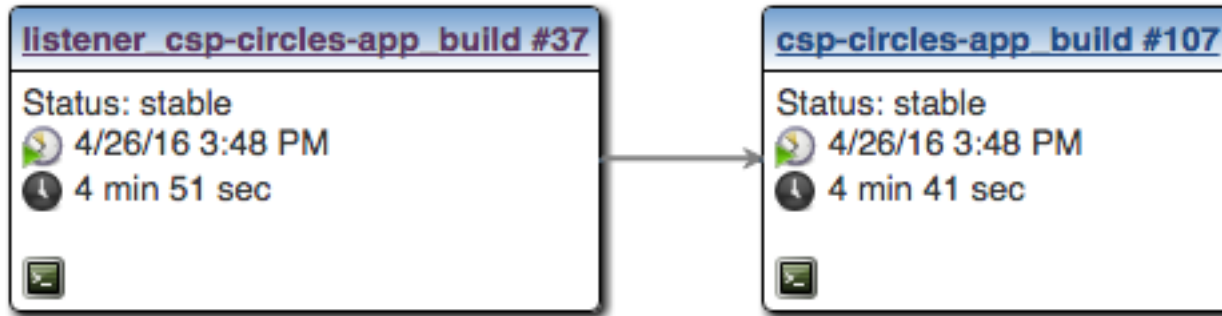


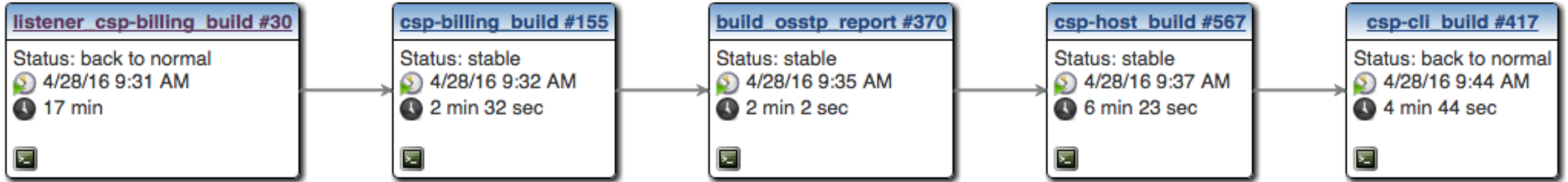
gerrit
Gerrit Code Review



Parallel Deployments







listener_csp-common_build #331
Status: stable
● 4/28/16 11:47 AM
● 32 min

csp-common_build #328
Status: stable
● 4/28/16 11:48 AM
● 2 min 38 sec

build_osstp_report #374
Status: stable
● 4/28/16 11:51 AM
● 2 min 24 sec

csp-public-api_build #280
Status: stable
● 4/28/16 11:54 AM
● 4 min 35 sec

csp-host_build #571
Status: stable
● 4/28/16 12:03 PM
● 5 min 38 sec

csp-cli_build #422
Status: stable
● 4/28/16 12:13 PM
● 3 min 40 sec

destroy_csp #8
Status: back to normal
● 4/28/16 12:18 PM
● 6.3 sec

deploy_csp #139
Status: stable
● 4/28/16 12:19 PM
● 56 sec

csp-account-management_build #365
Status: stable
● 4/28/16 11:54 AM
● 6 min 42 sec

csp-billing_build #158
Status: stable
● 4/28/16 11:54 AM
● 5 min 26 sec

csp-build-plugins_build #256
Status: stable
● 4/28/16 11:54 AM
● 1 min 49 sec

csp-email-management_build #260
Status: stable
● 4/28/16 11:54 AM
● 5 min 4 sec

csp-insight_build #261
Status: stable
● 4/28/16 11:54 AM
● 6 min 2 sec

csp-reporting_build #233
Status: stable
● 4/28/16 11:54 AM
● 3 min 46 sec

csp-service-lifecycle_build #299
Status: stable
● 4/28/16 11:54 AM
● 6 min 25 sec

csp-telemetry_build #240
Status: stable
● 4/28/16 11:54 AM
● 4 min 19 sec

csp-ui_build #303
Status: stable
● 4/28/16 11:55 AM
● 8 min 28 sec

csp-upgrade-plugin_build #79
Status: stable
● 4/28/16 11:55 AM
● 5 min 36 sec

Upgrading a Stateful platform

Goals:

- Minimal service interruptions
- Support schema changes

Challenges:

- Symmetrical cluster: Can't refactor / add API paths
- State & Business Logic in the same tier: can't separate schema upgrade from BL changes

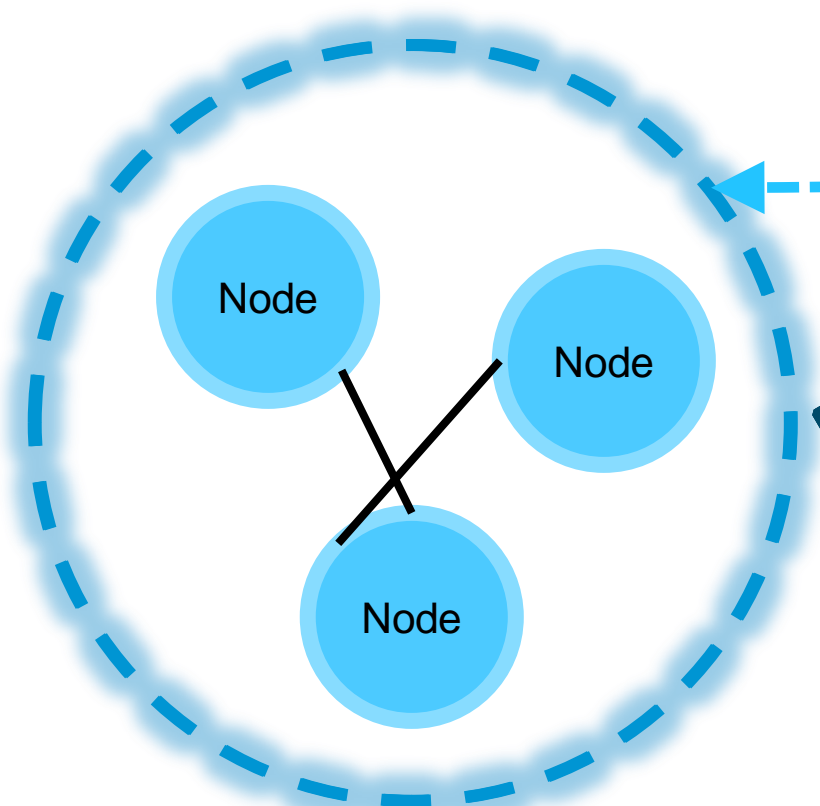
Upgrading a Stateful platform

Design:

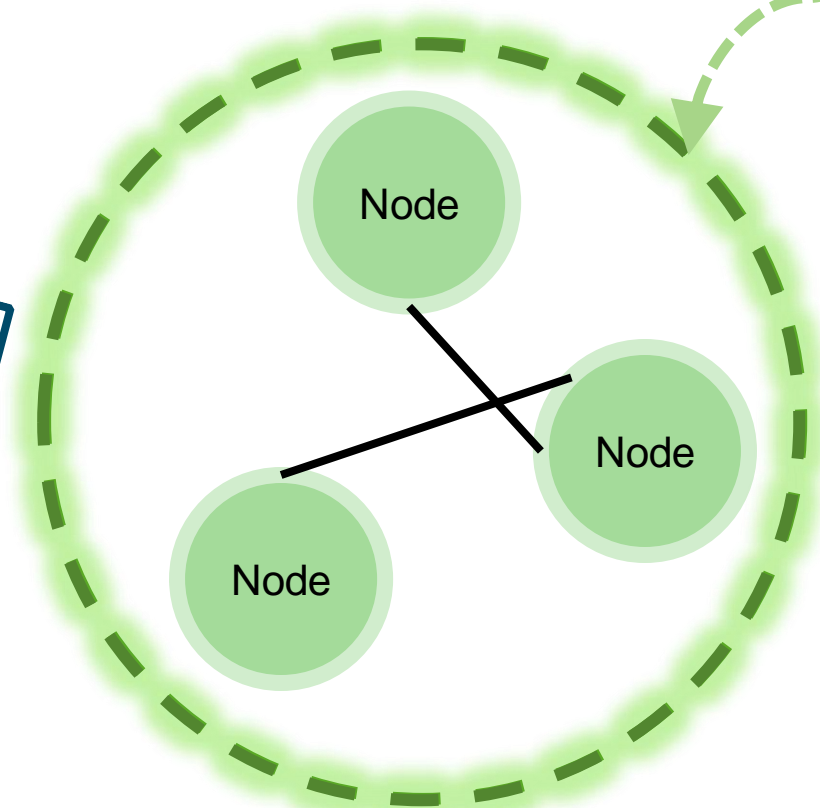
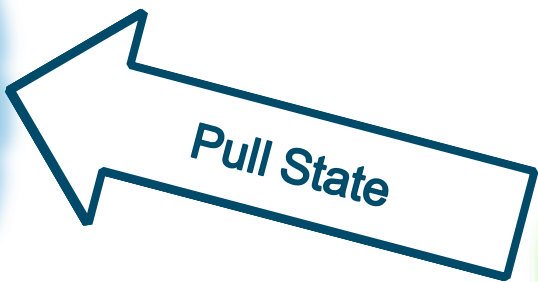
- Work in cycles, get meaningful metrics per cycle
- Each cycle migrates and transforms state
- Use a Threshold to determine progress and cutoff point
- Smartly queue external traffic
- Reroute traffic to new cluster

External Clients

Reroute Traffic



Blue NodeGroup



Green NodeGroup

```
{  
  "documents": "10K",  
  "duration": "0.2S"  
}
```

Xenon – Distributed Control Plane

<https://github.com/vmware/xenon>

- A design pattern and runtime for scalable orchestration and management logic
- A runtime powering tiny REST services
- IO Pipeline integrates key building blocks within each service operation
- Production ready code with continuous integration tests, design documents

The Popular Way

Stand up N nodes for each of:

- Orchestration code & container (Spring Boot)
- Your HA persistency layer (Cassandra, Mongo)
- Your translation layer (ORM)
- Your arbitration/leader election (ZK, etcd, consul)
- Your UI server (node.js, tomcat, apache)
- Your cache layer (Redis, memcached)
- Your message bus, event broker

The Xenon Way

Stand up N nodes running Xenon services:

- Orchestration as stateless or stateful REST endpoints
- Persist, replicate state independently
- Manage concurrency with a single JVM and one thread per core across ALL services
- Provide per operation owner selection (leader)
- Pub / Sub
- Stats
- UI
- Tracing

Links

- Jenkins Jobs Builder - <http://docs.openstack.org/infra/jenkins-job-builder>
- Xenon - <https://github.com/vmware/xenon>

Thank you!

Q&A

Decentralized Model

- Scalable to lots of nodes
 - SWIM node discovery and maintenance
 - Replication with Eventual OR Strong Consistency (choose!)
- Every node in a node group has the same core services
 - Operational simplicity

Indexing/Queries

- Multi version, fully indexed, replicated document store
 - Lucene!
- Query services with rich document query support modeled as tasks
 - Real time or historical
- Collections are just queries

Programming Model

- Isolated, asynchronous components listening on URIs
- Each service instance represents a “living” document
 - All side effects happen through REST actions on document
 - Replication, consensus, notifications all leveraging symmetric model
- Stateless handlers are offered latest state and request body
- Developer declares requirements through Service options
 - Replication with Strong (Eager) or Eventual consistency
 - Scale out (Owner selection)
 - Instrumentation
 - Persistence (with deep indexing)
 - And more ...