Zuul as a build system

lessons learned in the Tungsten Fabric infra

Jarosław Łukow and Łukasz Łukasiewicz

Who we are

About

- admins of Tungsten Fabric CI/CB system
- started deploying Zuul v3 in November 2017
- we work at CodiLime
  - DevOps, SDN, NFV, Cloud-Native services
  - 200 engineers, 48 270 coffees/year*
- contacts:
  - diabelko: lukasz@codilime.com
  - jluk: jaroslaw.lukow@codilime.com

* at least the espresso machines say so

Agenda

- **about**
- build system
- reusing Zuul jobs
- testing jobs
- cool to see in Zuul

What is Tungsten Fabric

Intro

- formerly known as OpenContrail
- multicloud, multistack SDN solution
- integrates with OpenStack, Kubernetes, OpenShift, VMware

Project specifics

Intro

- C, Go, Python
- single build of all components (30 repos)
- Android Repo tool
- services deployed as containers
- platforms:
  - CentOS (mostly)
  - RHEL
  - Windows Server

Agenda

- about
- **build system**
- reusing Zuul jobs
- testing jobs
- cool to see in Zuul

The starting point - Jenkins CB system

Build system

- separate from CI (which was running on Zuul 2.5 at that time)
- different locations of dependencies
- different scripts
- different slave pool
- single-job pipeline

Then comes Zuul v3

Build system

- upgrade CI from 2.5
- accent on openness
- unify CI, build and release pipelines

The pipeline

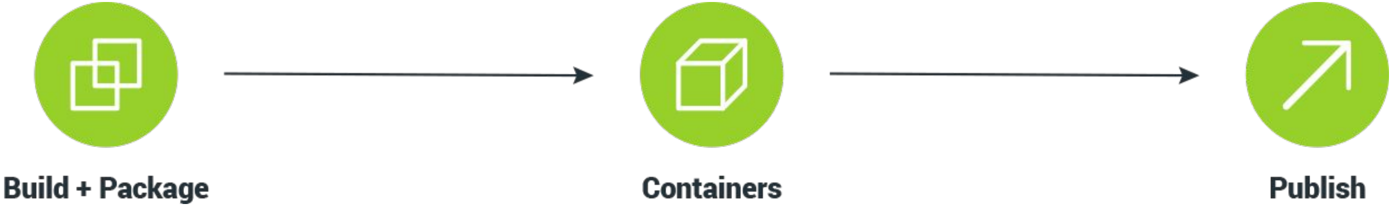
Build system

- compile and package
- containerize
- publish

The pipeline

Build system

- compile and package
- containerize
- publish



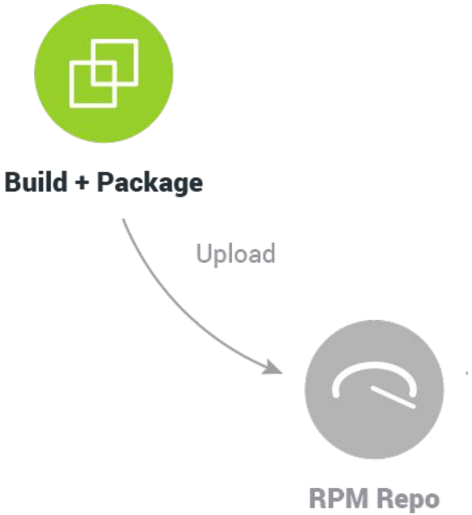
The pipeline

Build system

Build + Package

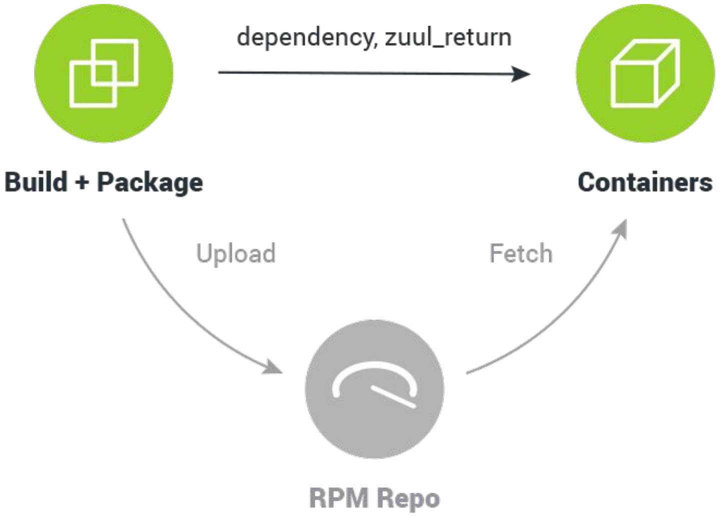
The pipeline

Build system



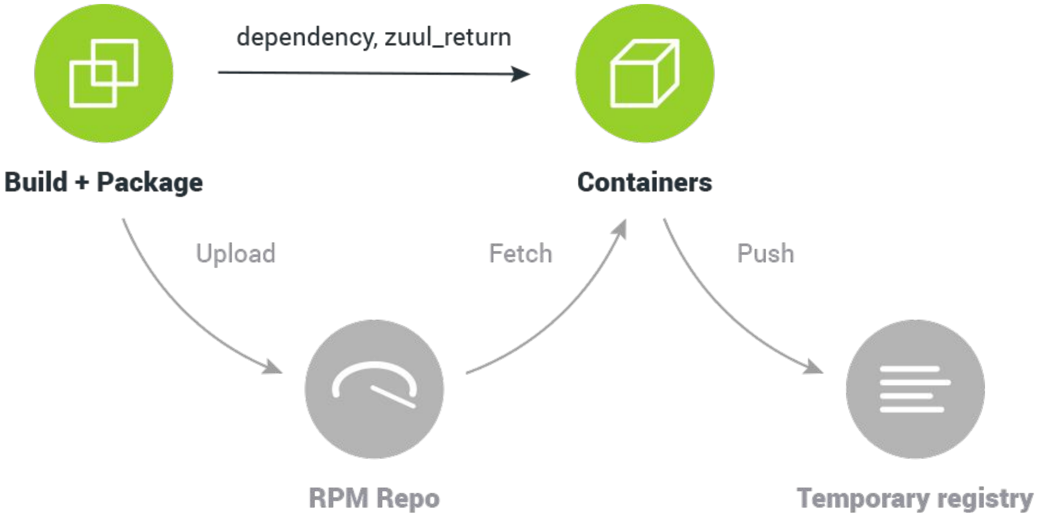
The pipeline

Build system



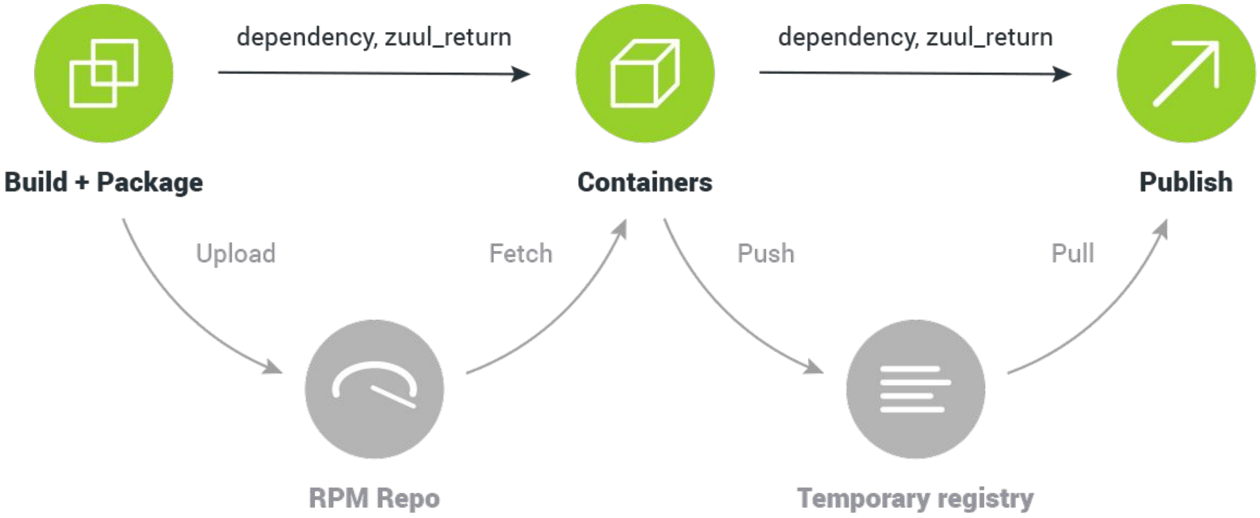
The pipeline

Build system



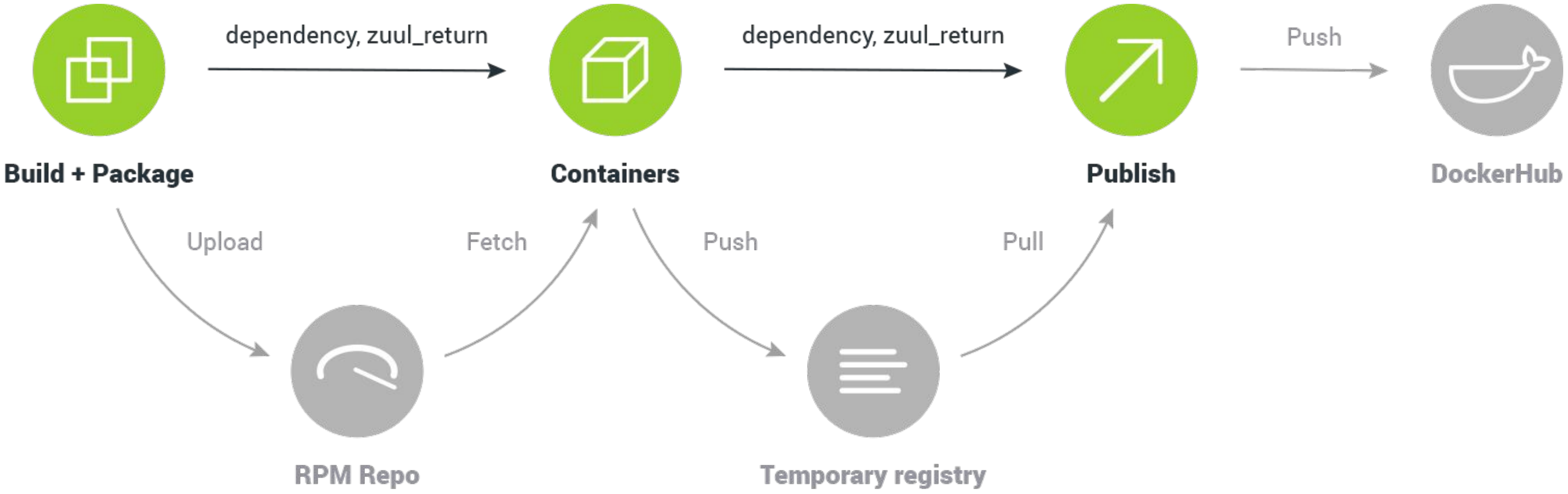
The pipeline

Build system



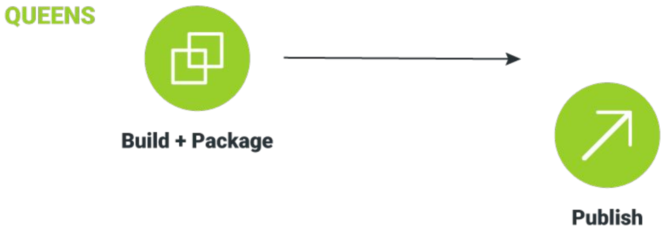
The pipeline

Build system



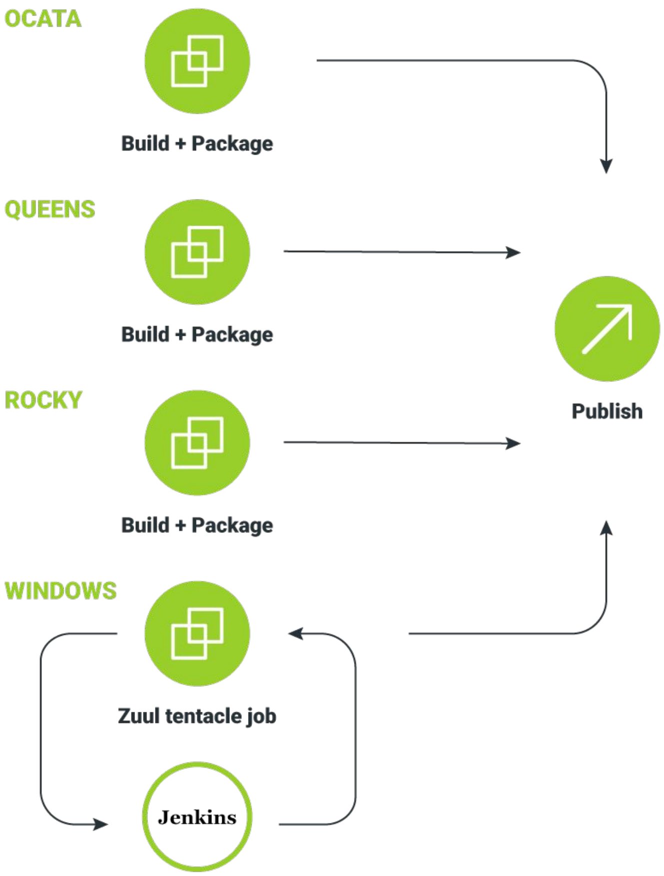
The pipeline

Build system



The pipeline

Build system



Surroundings

Build system

- mirrors
  - RPMs, DEBs
  - PyPI
  - Maven
- DockerHub cache

Builder VM images

Build system

- minimal approach
  - OS base
  - Zuul SSH key
- all the dependencies are installed during build
- the devs were disappointed with the fact that it's not a way to cache builds

Triggering

Build system

- scheduled (periodic)

```
trigger:
  timer:
    - time: "0 7 * * *"
```

- on every merge

```
trigger:
  gerrit:
    - event: ref-updated
```

- on-demand

```
zuul enqueue-ref --ref refs/heads/master ...
```

Triggering

Build system

- scheduled (periodic)

```
trigger:
  timer:
    - time: "0 7 * * *"
```

daily builds

- on every merge

```
trigger:
  gerrit:
    - event: ref-updated
```

docs, third party packages

- on-demand

```
zuul enqueue-ref --ref refs/heads/master ...
```

retrying builds

Our extensions

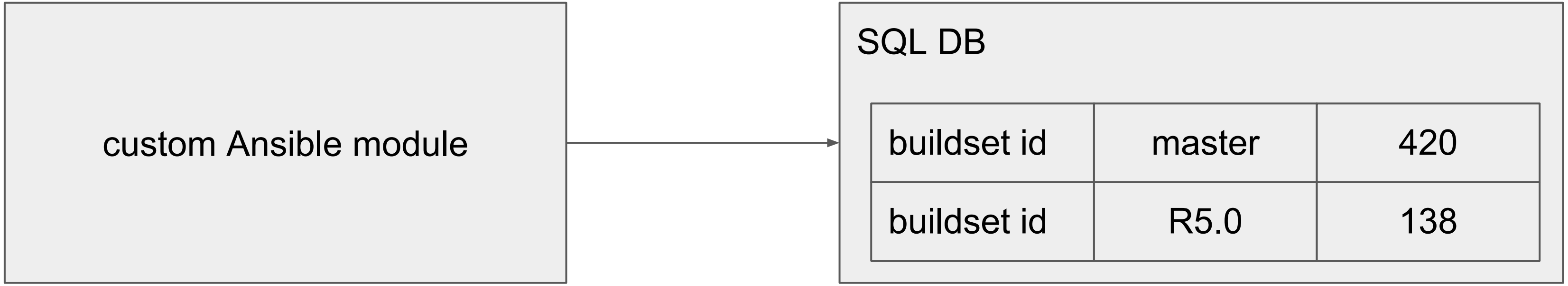
Build system

- consecutive build numbers
- dumping exact commit information
- dumping information about artifacts
- generating lists of changes included in builds (changelog)

Our extensions

Build system

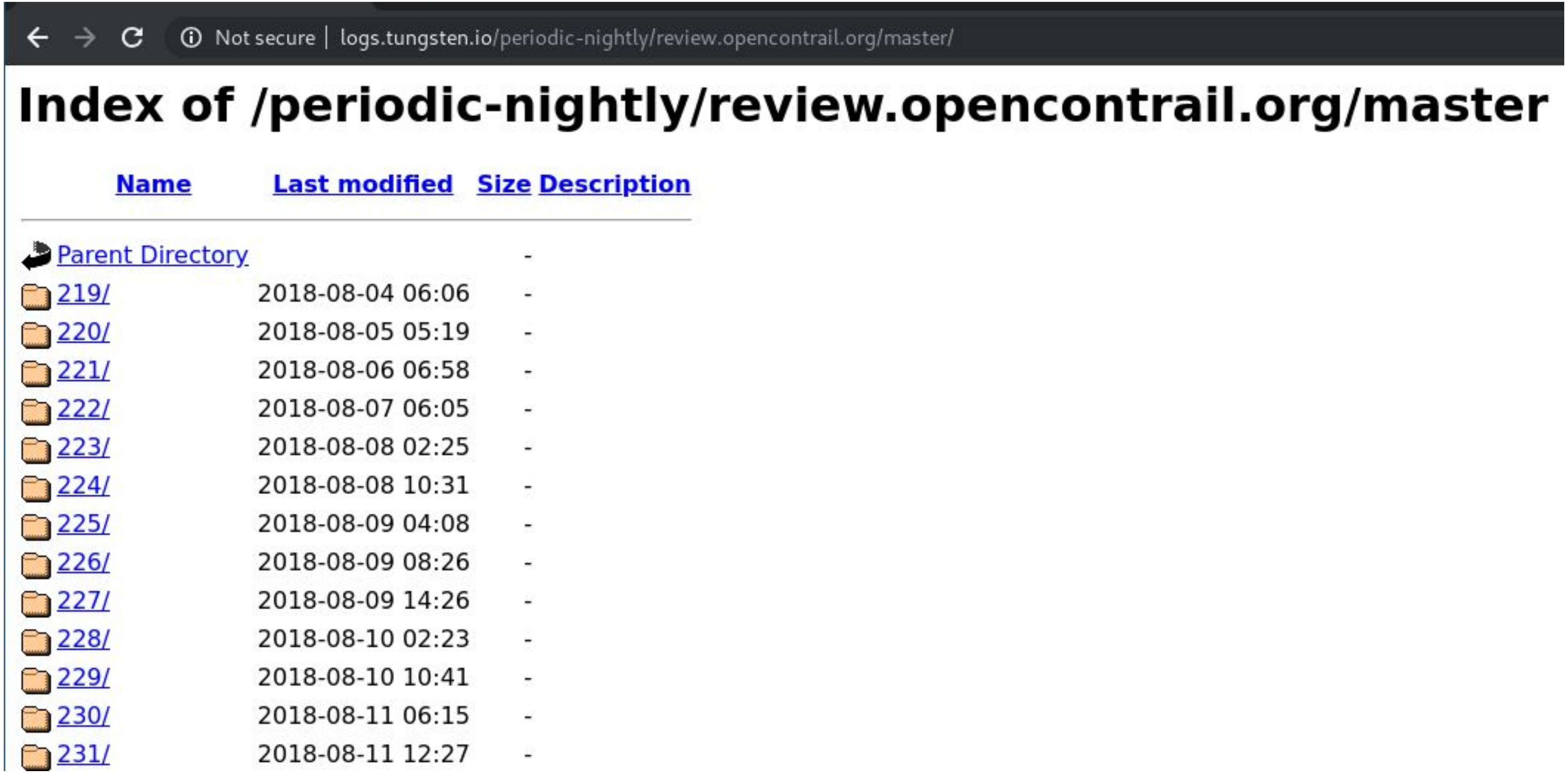
- consecutive build numbers



custom Ansible module

SQL DB

buildset id	master	420
buildset id	R5.0	138


Not secure | logs.tungsten.io/periodic-nightly/review.opencontrail.org/master/

Index of /periodic-nightly/review.opencontrail.org/master

Name	Last modified	Size	Description
Parent Directory		-	
219/	2018-08-04 06:06	-	
220/	2018-08-05 05:19	-	
221/	2018-08-06 06:58	-	
222/	2018-08-07 06:05	-	
223/	2018-08-08 02:25	-	
224/	2018-08-08 10:31	-	
225/	2018-08-09 04:08	-	
226/	2018-08-09 08:26	-	
227/	2018-08-09 14:26	-	
228/	2018-08-10 02:23	-	
229/	2018-08-10 10:41	-	
230/	2018-08-11 06:15	-	
231/	2018-08-11 12:27	-	

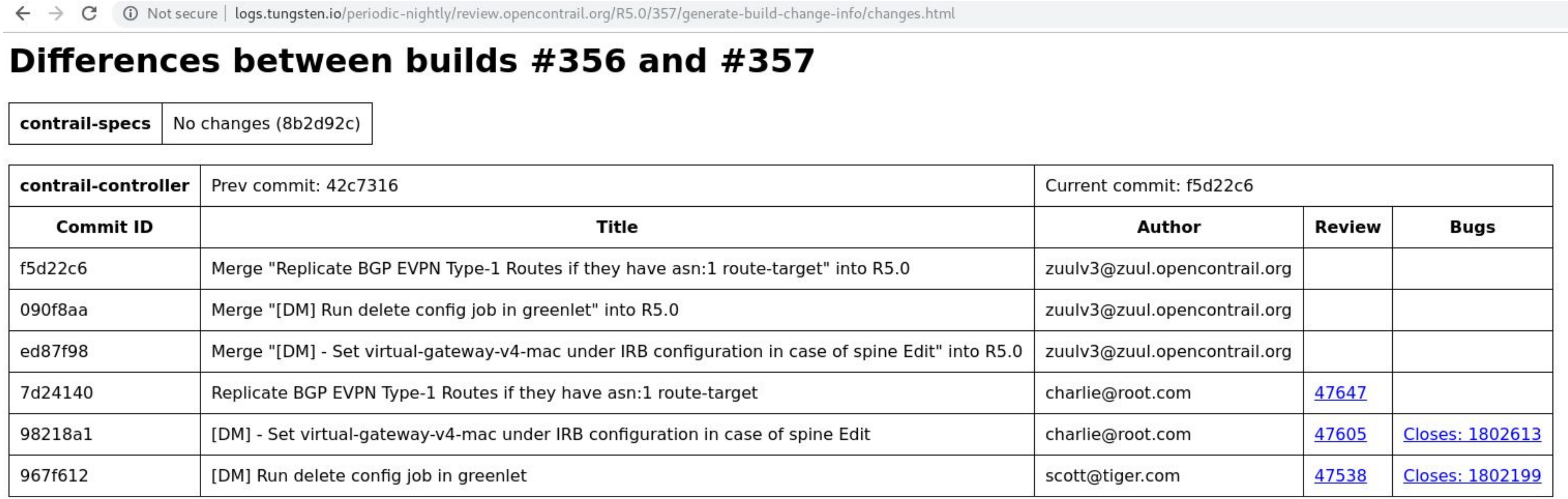
Our extensions

dumping exact commit information

```
{"review.opencontrail.org/Juniper/contrail-controller": {
        "revisions": {
            "current": "f5d22c6",
            "previous": "42c7316"
        }
        "changes": [{
                "title": "Replicate BGP EVPN Type-1 Routes...",
                "timestamp": 1542144758,
                "author": {...},
                "bugs": [],
                "sha": "7d24140f16b6d066f9802e0547b41deb2a846893",
                "message": "...",
                "change": {
                    "number": 47647,
                    "id": "I4387030ca62495afe949f78b5fc391049f4783d5"
                }
            },
...
```

Our extensions

Build system



logs.tungsten.io/periodic-nightly/review.opencontrail.org/R5.0/357/generate-build-change-info/changes.html

Differences between builds #356 and #357

contrail-specs	No changes (8b2d92c)

contrail-controller	Prev commit: 42c7316	Current commit: f5d22c6		
Commit ID	**Title**	**Author**	**Review**	**Bugs**
f5d22c6	Merge "Replicate BGP EVPN Type-1 Routes if they have asn:1 route-target" into R5.0	zuulv3@zuul.opencontrail.org		
090f8aa	Merge "[DM] Run delete config job in greenlet" into R5.0	zuulv3@zuul.opencontrail.org		
ed87f98	Merge "[DM] - Set virtual-gateway-v4-mac under IRB configuration in case of spine Edit" into R5.0	zuulv3@zuul.opencontrail.org		
7d24140	Replicate BGP EVPN Type-1 Routes if they have asn:1 route-target	charlie@root.com	47647	
98218a1	[DM] - Set virtual-gateway-v4-mac under IRB configuration in case of spine Edit	charlie@root.com	47605	Closes: 1802613
967f612	[DM] Run delete config job in greenlet	scott@tiger.com	47538	Closes: 1802199

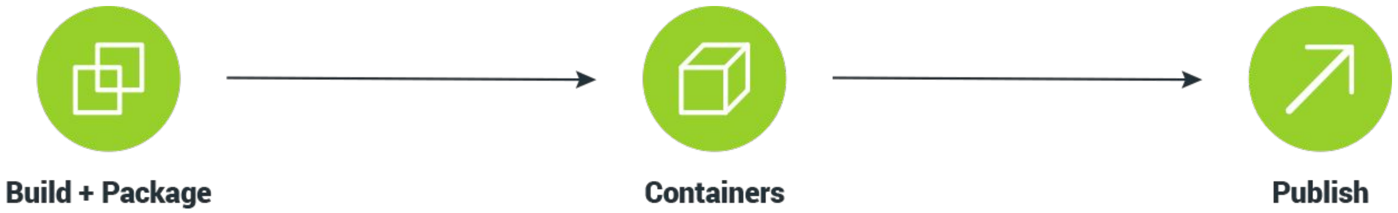
Build pipeline meets ‘check’

Build system

- uses the same jobs as periodic pipeline
- sanity jobs use containers built in previous jobs
- publishing artifacts at the end is not needed

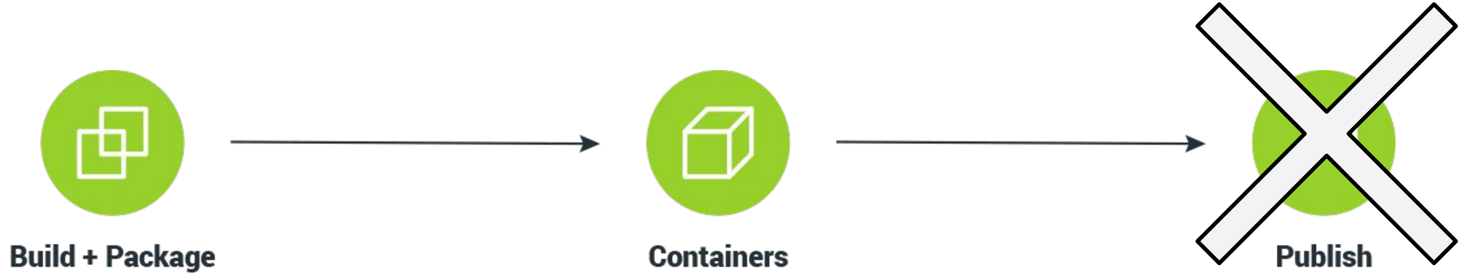
Build pipeline meets ‘check’

Build system



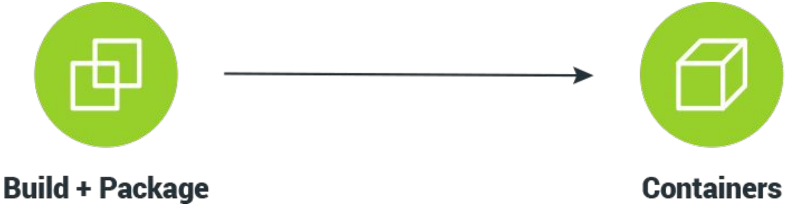
Build pipeline meets ‘check’

Build system



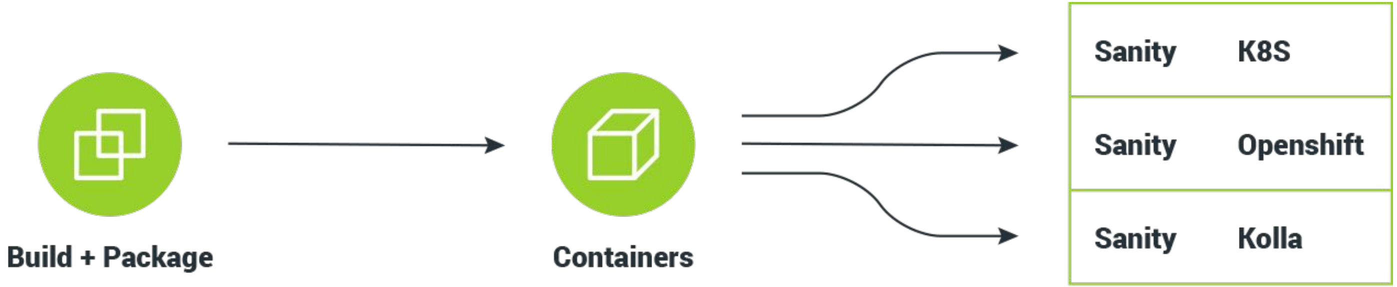
Build pipeline meets ‘check’

Build system



Build pipeline meets ‘check’

Build system



Build + Package

Containers

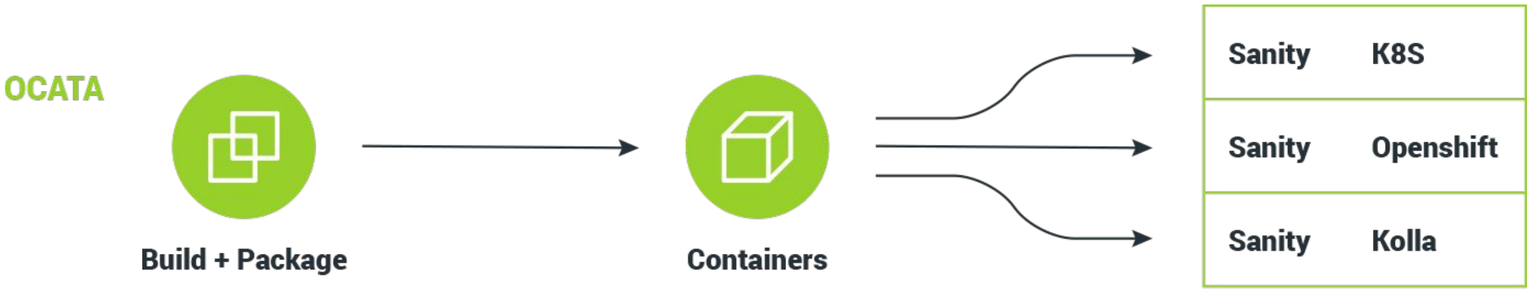
Sanity K8S

Sanity Openshift

Sanity Kolla

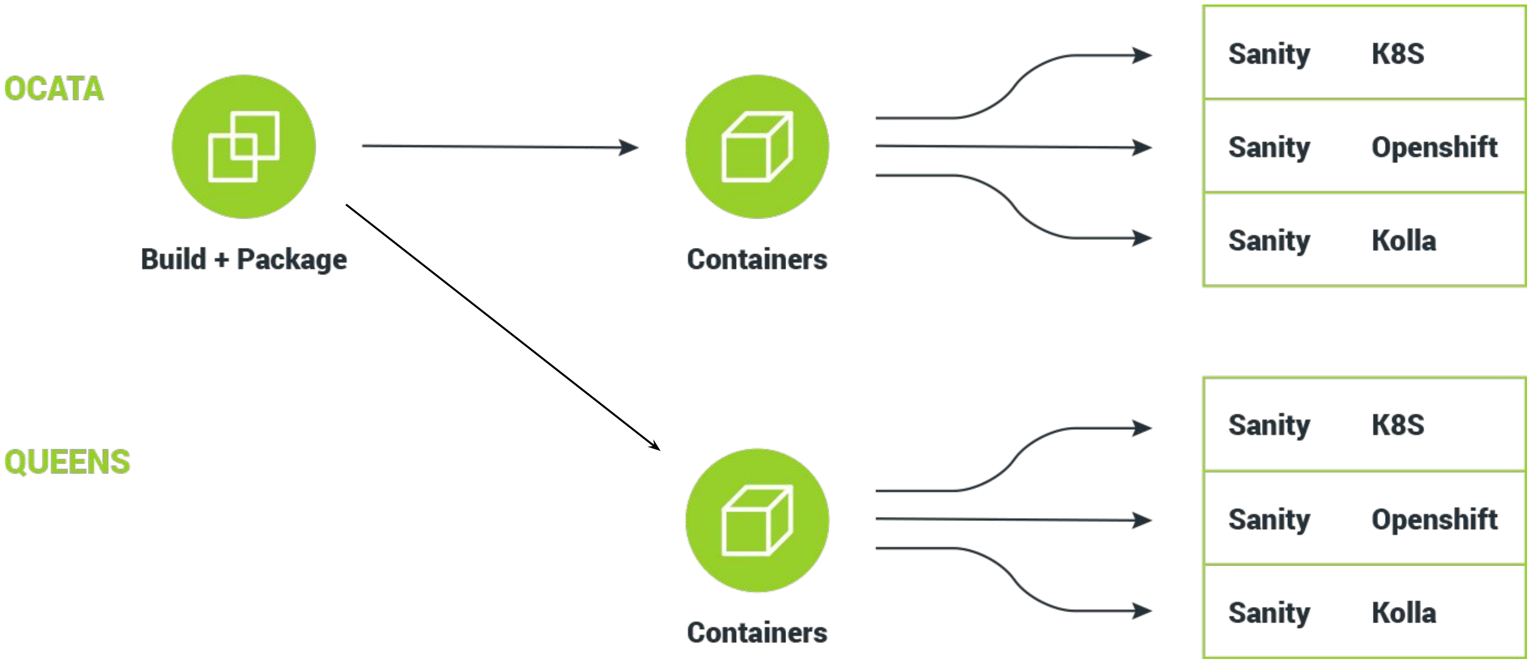
Build pipeline meets ‘check’

Build system



Build pipeline meets ‘check’

Build system



Agenda

- about
- build system
- **reusing Zuul playbooks**
- testing jobs
- cool to see in Zuul

Original idea

Reusing Zuul playbooks

- jobs are already shared by the CI and Build jobs
- perhaps they can also be used in developer environment
- so… let’s create Zuul-agnostic playbooks and roles

Why

Reusing Zuul playbooks

- so developers can reproduce the CI environment with ‘one-click’
- to save us some time
- because it’s cool to reuse stuff

The Zuul job dilemma

Reusing Zuul playbooks

- reusable playbooks vs. convenient usage of variables
- good ARA visibility vs. single "shell" entrypoint

Why it failed

Reusing Zuul playbooks

- run playbook has to do all the work
- can’t leverage pre- and post- playbooks
- too hard to draw a strict line between Ansible and Zuul
- too hard to mock Zuul outside of Zuul (you don't want to parse config on your own)

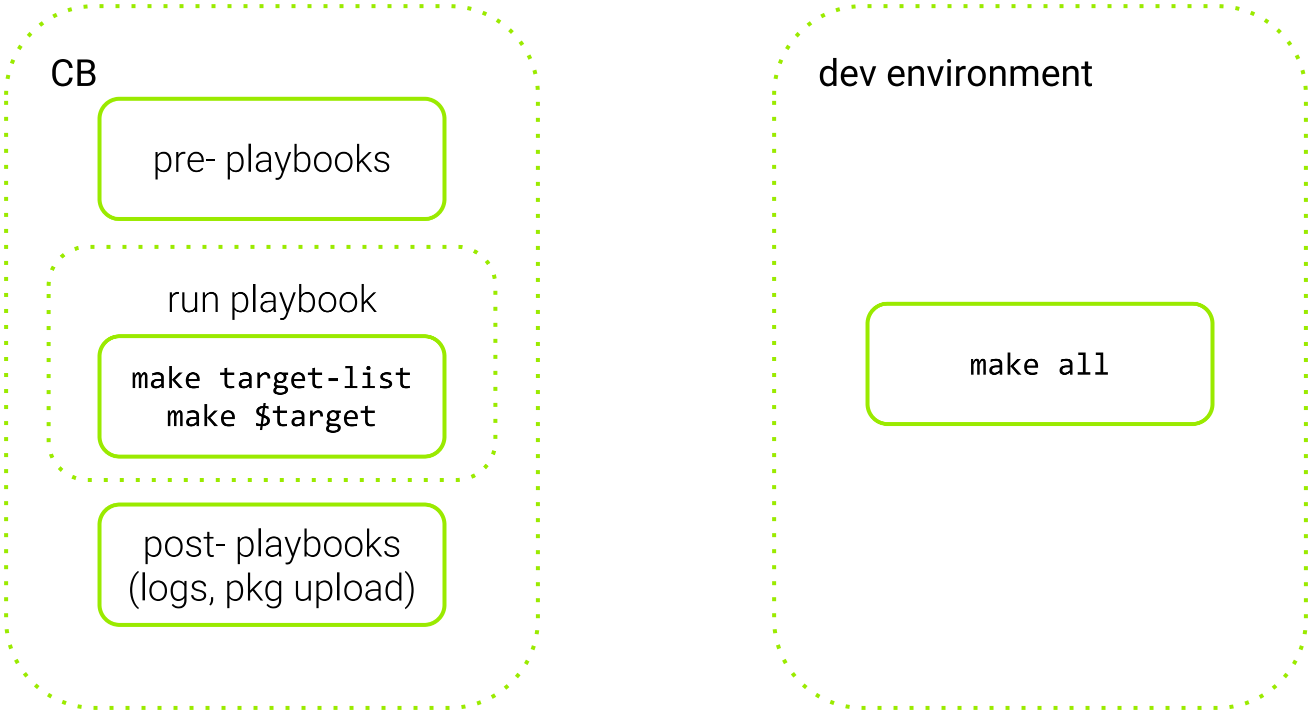
Aftermath

Reusing Zuul playbooks

- packaging/building logic inside the code, instead of the CI
- simple Makefiles
- still visible in ARA

Aftermath

Reusing Zuul playbooks



CB

pre- playbooks

run playbook

```
make target-list
make $target
```

post- playbooks
(logs, pkg upload)

dev environment

```
make all
```

Agenda

- about
- build system
- reusing Zuul jobs
- **testing jobs**
- cool to see in Zuul

CI of CI

Testing jobs

- your CI jobs are stored in a repo as code, so…
- you should test them like everything else
- but, some things are not testable in Zuul (for a good reason)
- you can take the risk or...

Ideas for testing jobs

Testing jobs

- setting all pipelines as post-review (‘disable security’)
- separate development environment (Zuul, Gerrit, Nodepool)
- Zuul on a laptop
- unit testing roles
- running copies/mocks of jobs

Mocking your jobs

Testing jobs

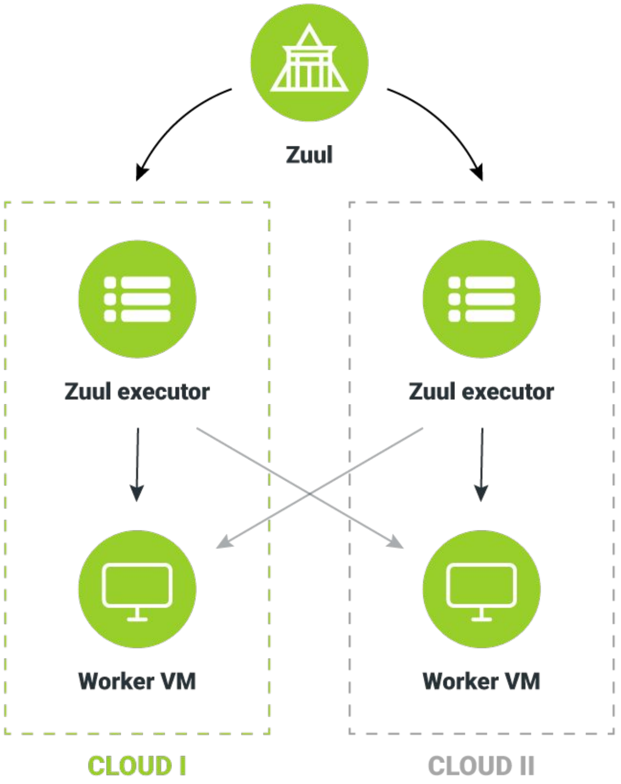
- review to an untrusted repo
- secrets as variables (dummy values)
- changing Ansible host

Agenda

- about
- build system
- reusing Zuul jobs
- testing jobs
- **cool to see in Zuul**

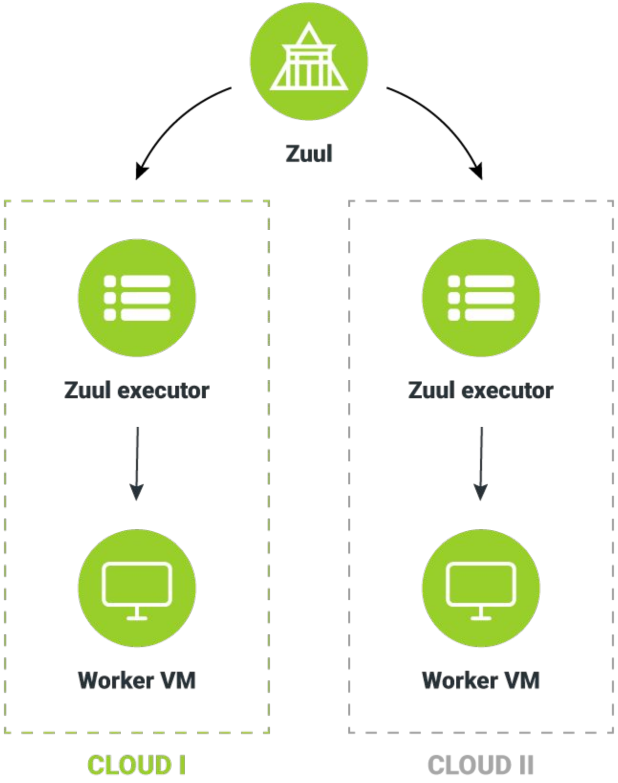
Matching executor with its cloud

Cool to see in Zuul



Matching executor with its cloud

Cool to see in Zuul



Matrix build definitions

Cool to see in Zuul

```
- job:
    name: contrail-build-containers-centos7-newton
    parent: contrail-build-containers-base
    vars:
      openstack_version: newton

- job:
    name: contrail-build-containers-centos7-ocata
    parent: contrail-build-containers-base
    vars:
      openstack_version: ocata

- job:
    name: contrail-build-containers-centos7-queens
    parent: contrail-build-containers-base
    vars:
      openstack_version: queens

- project:
    name: Juniper/contrail-analytics
    check:
      jobs:
        - contrail-build-containers-centos7-newton
        - contrail-build-containers-centos7-ocata
        - contrail-build-containers-centos7-queens
```

```
- job:
    name: contrail-build-containers-centos7-{openstack_version}
    parent: contrail-build-containers-base

- project:
    name: Juniper/contrail-analytics
    check:
      jobs:
        - contrail-build-containers-centos7-{openstack_version}:
            vars:
              openstack_version: newton
        - contrail-build-containers-centos7-{openstack_version}:
            vars:
              openstack_version: ocata
        - contrail-build-containers-centos7-{openstack_version}:
            vars:
               openstack_version: queens
```

Wrapping up

Takeaways

Zuul as a build system

- Tungsten Fabric has a cool CI/CB system
- how to handle build artifacts with Zuul
- reusing your jobs is the key
- you can test your jobs not-in-the-production

Future plans

Zuul as a build system

- continuous upgrade of Zuul
- running build and unittest jobs inside containers instead of VMs
- supercedent pipeline manager

Thank you

codilime®
CREATING VALUE

tungstenfabric