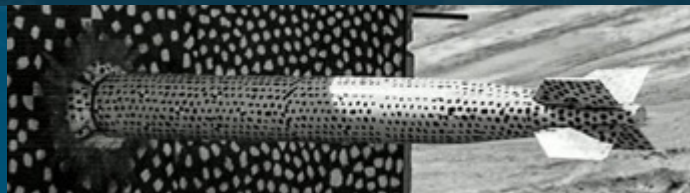
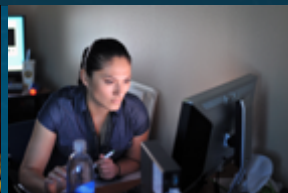


CI Tools as Lego Blocks

Build Your Ideal Custom Solution



PRESENTED BY

Jason M. Gates, David Collins, Josh Braun



- David Collins
 - DevOps Engineer
 - Scientific Applications & User Support
- Josh Braun
 - Year-Round Graduate Intern
 - Software Engineering & Research



The Continuous Integration Landscape





- Examples
 - GitHub Actions
 - GitLab CI/CD
- Advantages
 - Everything in one place
 - Free cloud computing resources
 - Emphasis on ease of use

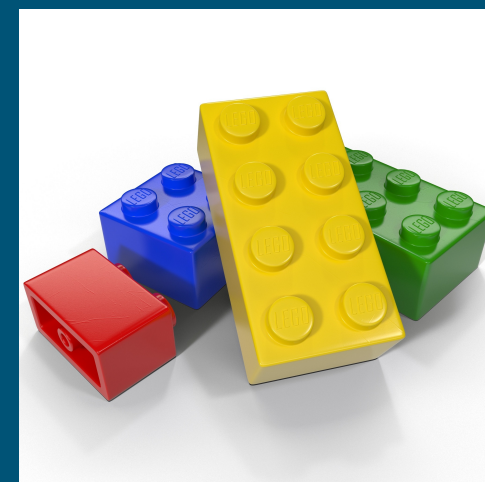


- Architected for lightweight applications
 - Requiring minimal cores/memory
 - Thousands of tests run in seconds
- Assume open source
 - What about need-to-know concerns?
 - What about restricted networks?
- Emphasis on simplicity prohibits complexity
 - No full-fledged programming language to govern the pipelines
 - No clean way to preserve state between jobs



The Lego Paradigm

- Determine the list of tasks you need to accomplish
- Figure out which tools excel in each of those spaces
- Decide how to connect the different pieces
- Snap it all together
- Create some lightweight “glue” to hold it there





Example: JOG-CI

Jenkins, OpenStack, GitLab – Continuous Integration



- From their website: “The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.”
- Jenkins Pipeline plugin suite
 - Full-fledged programming language for scripting complex control flow
 - Pipeline scripts live in version-controlled repositories
 - Ability to manage hundreds of jobs through a handful of scripts





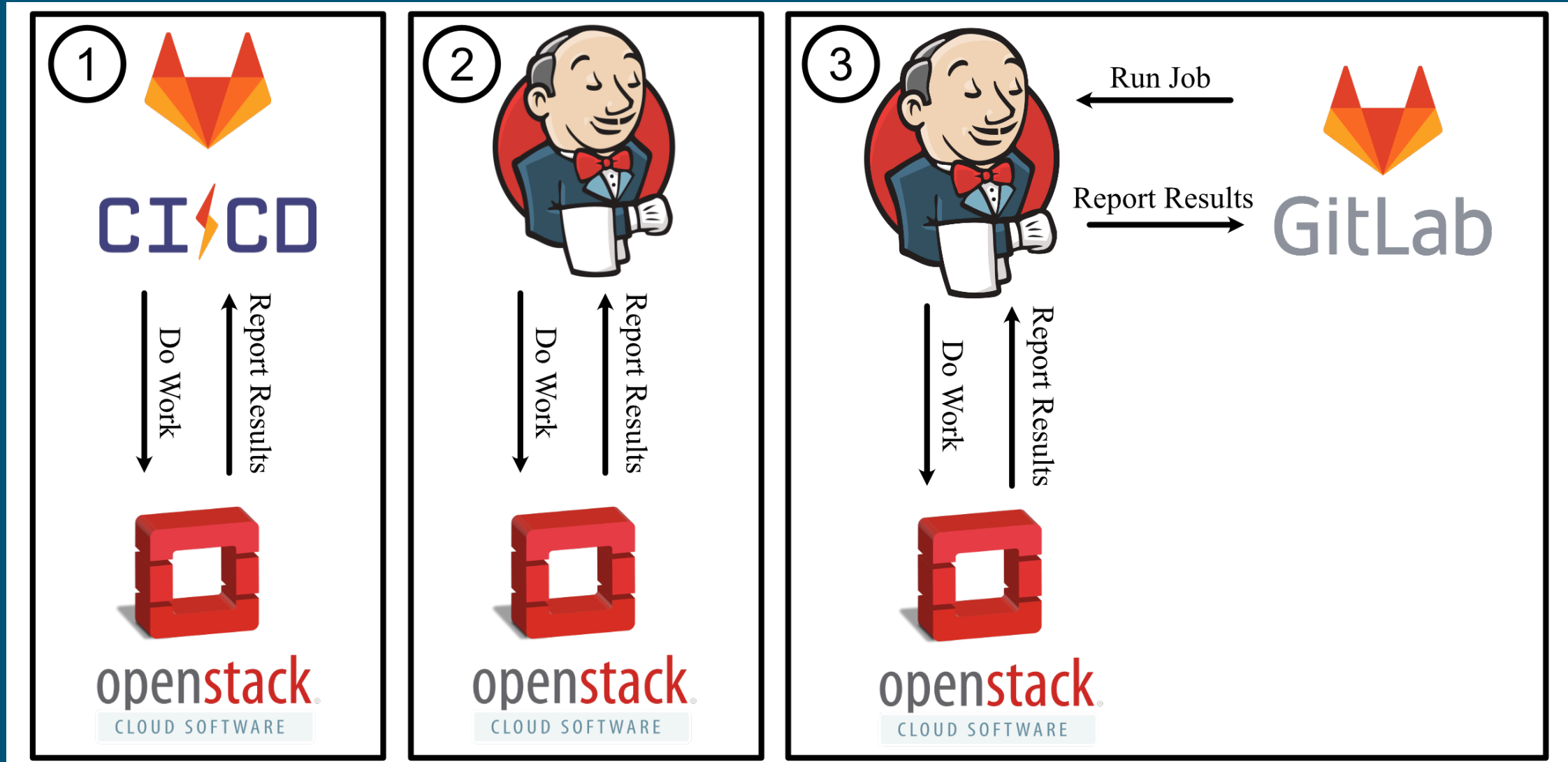
- From their website: “Deployed by thousands. Proven production at scale. OpenStack is a set of software components that provide common services for cloud infrastructure.”
- Allows teams to manage their own private cloud tenant
- Provides flexibility to stand up / tear down instances as needed
- Finer-grained control over compute resources





- From their website: “GitLab CI/CD is a tool built into GitLab for software development through the continuous methodologies: Continuous Integration (CI), Continuous Delivery (CD), Continuous Deployment (CD).”
- Where the repositories live, and the project management, design discussions, code review, etc., happens
- Want to make sure all merge requests pass initial set of testing







- This is the lightweight “glue” that sticks the block together
- Python 3.6+
- Notable packages: `openstacksdk`, `python-jenkins`
- Utilities:
 - `stand_up_ee_openstack.py`
 - `tear_down_ee_openstack.py`
 - `connect_ee_openstack_to_jenkins.py`



```
usage: python3 stand_up_cee_openstack.py [-h] [--dry] [-d] [-V]
      [--project-name PROJECT_NAME]
      [--network-name NETWORK_NAME]
      [--subnet-name SUBNET_NAME]
      [--router-name ROUTER_NAME]
      [--floating-ip-description FLOATING_IP_DESC]
      [--network-pool POOL_NAME]
      [--volume-names VOLUME_NAMES [VOLUME_NAMES ...]]
      [--stage {instance,network,subnet,router,router_interface,floating_ips,volumes,
               security_group} [{instance,network,subnet,router,router_interface,
               floating_ips,volumes,security_group} ...]]
      [--project-id PROJECT_ID]
      [--flavor FLAVOR]
      [--volume-sizes VOLUME_SIZES [VOLUME_SIZES ...]]
      [--install-gitlab-runner]
      [--gitlab-registration-token GITLAB_RT]
      [--entity-account-name ENTITY_ACCOUNT]
instance_name
```



```
usage: python3 tear_down_ee_openstack.py [-h] [--yes] [--dry] [-d] [-V]
      [--project-name PROJECT_NAME]
      [--network-name NETWORK_NAME]
      [--subnet-name SUBNET_NAME]
      [--router-name ROUTER_NAME]
      [--floating-ip-description FLOATING_IP_DESC]
      [--network-pool POOL_NAME]
      [--volume-names VOLUME_NAMES [VOLUME_NAMES ...]]
      [--stage {instance,network,subnet,router,router_interface,floating_ips,volumes,
               security_group} [{instance,network,subnet,router,router_interface,
               floating_ips,volumes,security_group} ...]]
      [--project-id PROJECT_ID]
instance_name
```



pipeline

passed

coverage

97.00%

docs

latest

passed

#150556

latest



P master -> 1f20f2ed

Merge branch '11-need-to-re...



00:00:23

1 week ago

passed

#150554



P 11-need-to-... -> e55b44af

Added functionality to remov...



00:00:23

1 week ago

passed

#150551



P 11-need-to-... -> 0010c497

Fixed tests and added a cou...



00:00:26

1 week ago



pipeline

passed

coverage

97.00%

docs

latest

Module	statements	missing	excluded	coverage
stand_up_cee_openstack.py	334	4	0	99%
tear_down_cee_openstack.py	171	2	0	99%
src/cee_openstack.py	264	11	0	96%
src/connect.py	50	2	0	96%
connect_cee_openstack_to_jenkins.py	144	12	0	92%
Total	963	31	0	97%
Total	963	31	0	97%



JOG-CI: Jenkins OpenStack GitLab – Continuous Integration



pipeline

passed

coverage

97.00%

docs

latest

JOG-CI



+



+



Search docs

CONTENTS:

- Stand Up CEE OpenStack Utility
- Connect OpenStack to Jenkins Utility
- Tear Down CEE OpenStack Utility
- CEE OpenStack Base Class
- Connect to CEE OpenStack Module

» Welcome to JOG-CI's documentation!

Welcome to JOG-CI's documentation!

Contents:

- Stand Up CEE OpenStack Utility
 - Required
 - Name Customization
 - Other Customization
 - Miscellaneous



GitLab

PAGES

Thanks Again



- David Collins
- Josh Braun
- Paul Wolfenbarger
- Dena Vigil
- Vadim Dyadechko