Continuous Integration and Continuous Delivery for Networks

Pete Lumbis - @PeteCCDE

Cumulus Networks Systems Engineer
CCIE #28677, CCDE 2012::3

WTF is CI/CD?

Continuous Integration (CI):

A system where all changes are automatically tested before pushed to production or seen by others.

WTF is CI/CD?

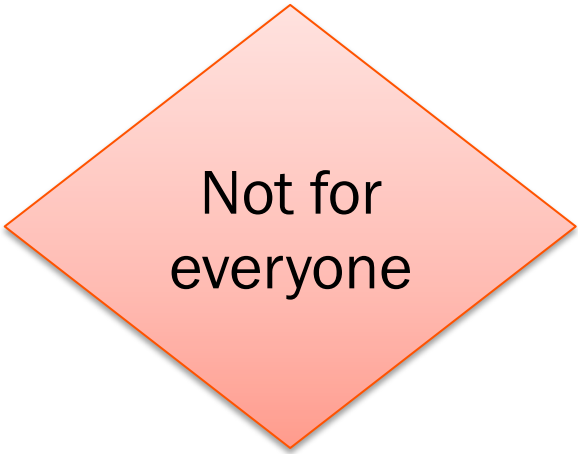
Continuous Integration (CI):
A system where all changes are automatically tested before pushed to production or seen by others.

Continuous Delivery (CD):
Built on a CI system where changes are made multiple times a day.

WTF is CI/CD?

Continuous Integration (CI):
A system where all changes are automatically tested before pushed to production or seen by others.

Continuous Delivery (CD):
Built on a CI system where changes are made multiple times a day.



WTF is CI/CD?

Continuous Integration (CI):

A system where all changes are automatically tested before pushed to production or seen by others.

Continuous Delivery (CD):

Built on a CI system where changes are made multiple times a day.

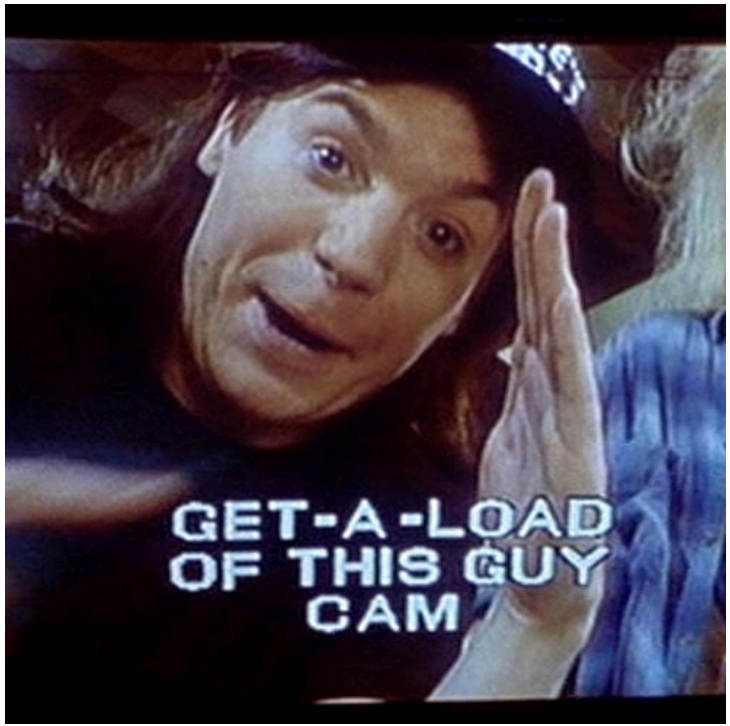
Agenda

Change Management Today

The Problems Today

NetDevOps Toolkit

Pete Who?



CCIE R&S, CCDE

Former Cisco TAC Routing Escalation

Current Cumulus Networks SE

Network Engineer, Not a Programmer

Leading Questions

Who thinks their change system works well?

Leading Questions

Who thinks their change system works well?

- I assume your manager is here

Leading Questions

Who thinks their change system works well?

- I assume your manager is here

Who lab tests 100% of changes?

Leading Questions

Who thinks their change system works well?

- I assume your manager is here

Who lab tests 100% of changes?

Who checks servers + apps on changes?

Leading Questions

Who thinks their change system works well?

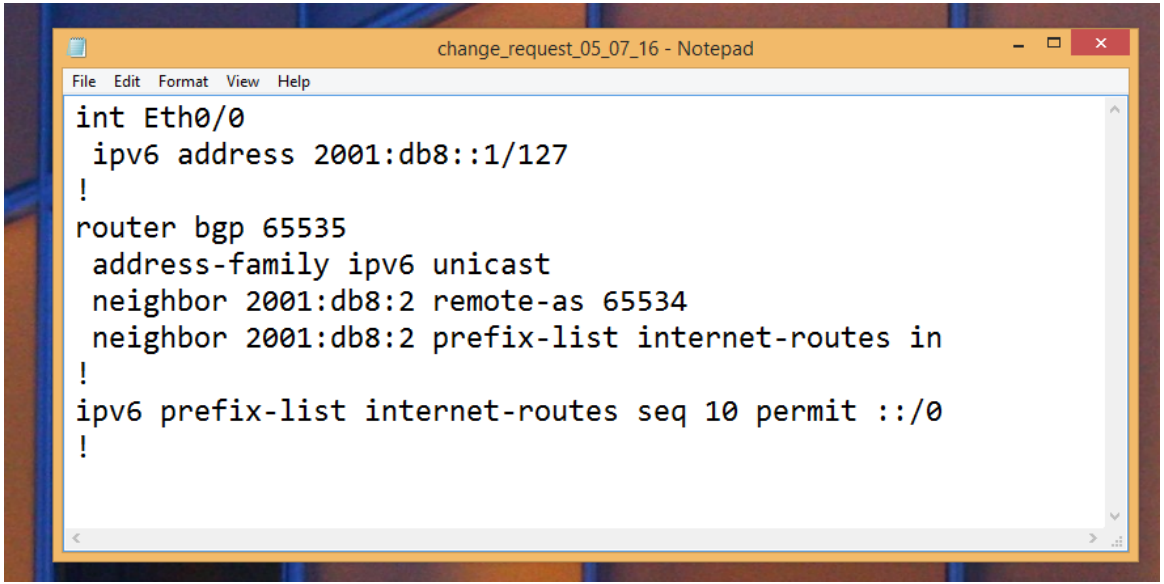
- I assume your manager is here

Who lab tests 100% of changes?

Who checks servers + apps on changes?

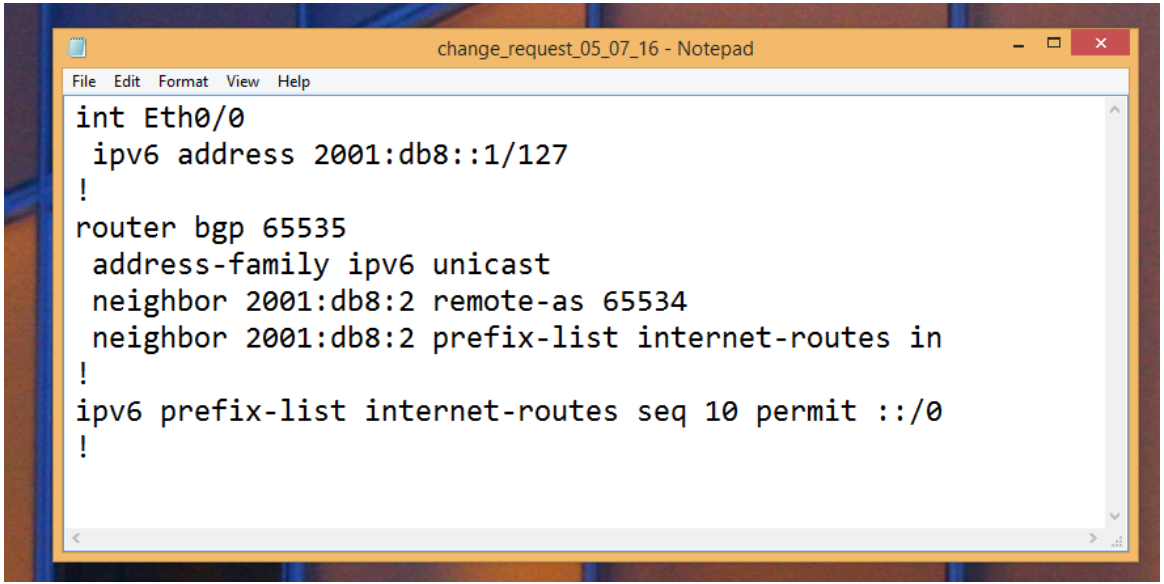
- Why is it your users?

Change Management Today

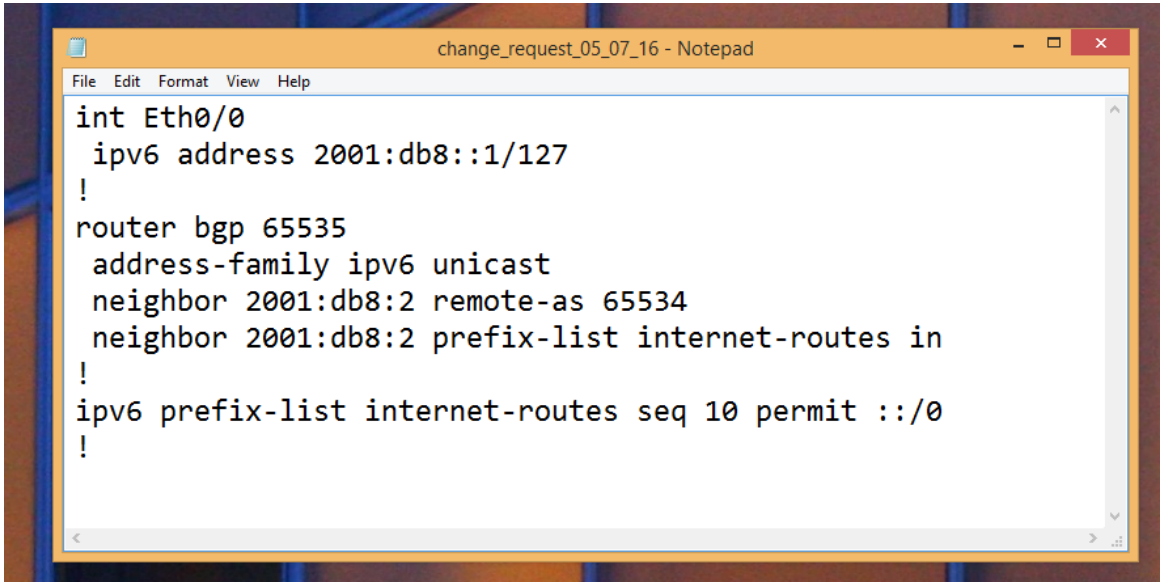


Change Management Today

Monday, 9:30am:
Write Change, Email Team



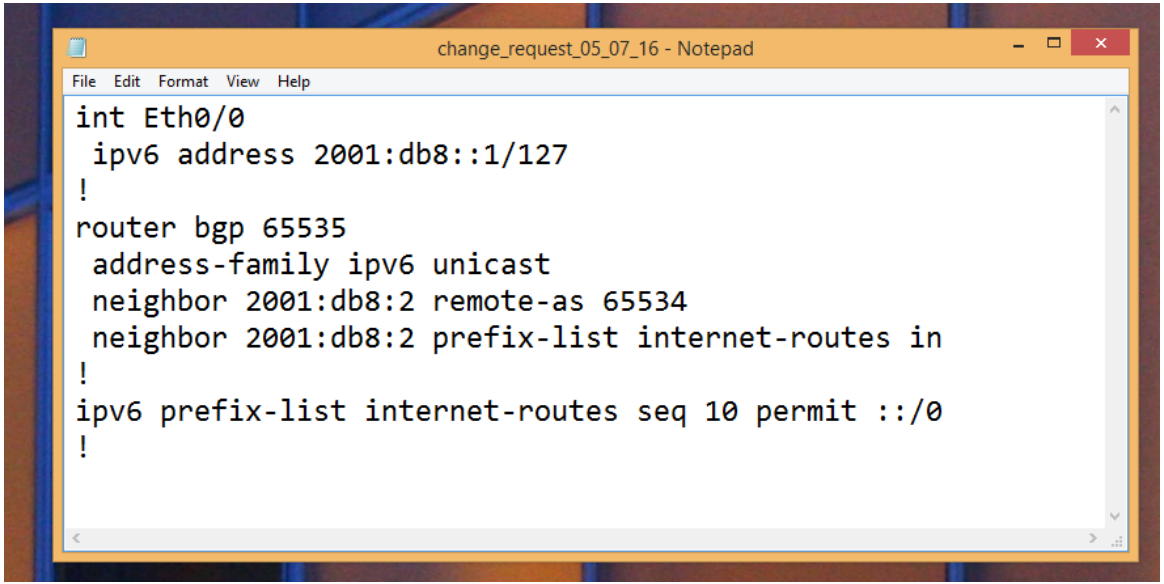
Change Management Today



Monday, 9:30am:
Write Change, Email Team

Thursday, 9:00am:
Ask team if they read email

Change Management Today

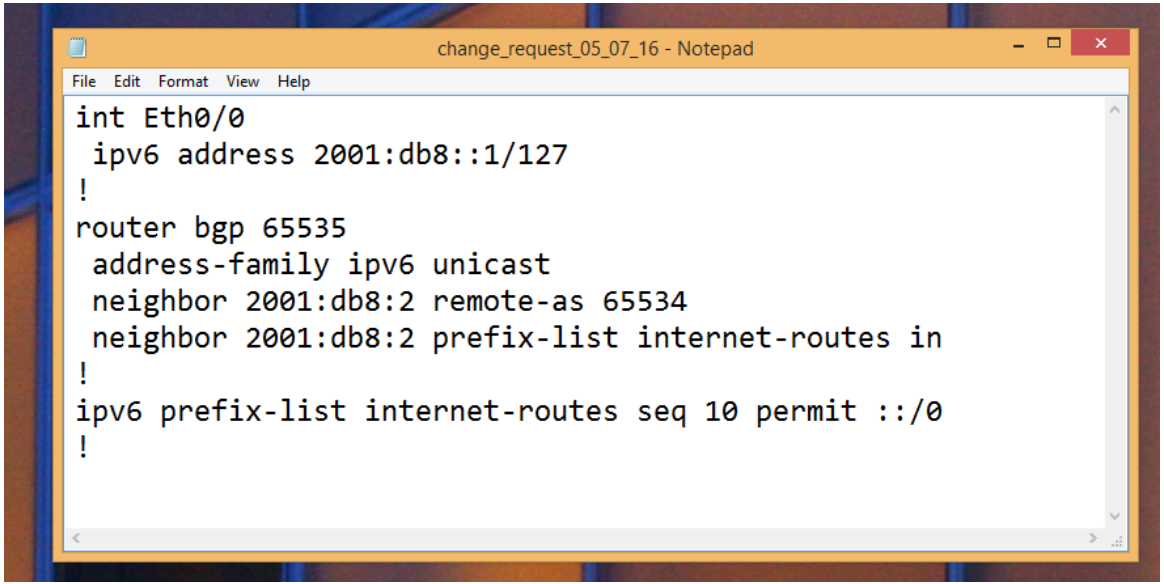


Monday, 9:30am:
Write Change, Email Team

Thursday, 9:00am:
Ask team if they read email

Friday, 11:00am:
Receive “looks good” from lead

Change Management Today



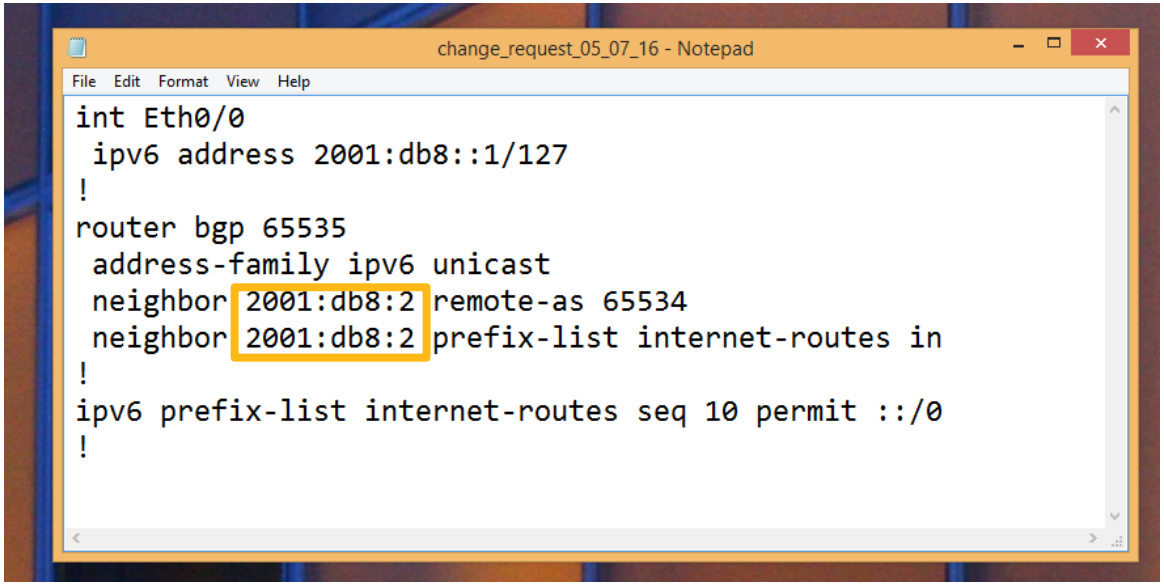
Monday, 9:30am:
Write Change, Email Team

Thursday, 9:00am:
Ask team if they read email

Friday, 11:00am:
Receive “looks good” from lead

Saturday, 10:00pm:
Implement Change in 2hr window

Change Management Today



Monday, 9:30am:
Write Change, Email Team

Thursday, 9:00am:
Ask team if they read email

Friday, 11:00am:
Receive “looks good” from lead

Saturday, 10:00pm:
Implement Change in 2hr window

Sunday, 8:30am:
Notice Typo, Fix. Go home.

Change Validation

Change Validation

Validation? What Validation?

Validation? What Validation?

Best Case

- 30 person bridge
- Every department signs off

Validation? What Validation?

Best Case

- 30 person bridge
- Every department signs off

Worst Case

- Ping some things
- Watch for tickets

Validation? What Validation?

Validation? What Validation?

Is ping your business application?

Recap: What's Broken?

Recap: What's Broken?

Poor change communication

Recap: What's Broken?

Poor change communication

Difficult to test outside of production

Recap: What's Broken?

Poor change communication

Difficult to test outside of production

Little change validation

Recap: What's Broken?

Poor change communication

Difficult to test outside of production

Little change validation

Manual validation doesn't scale

Recap: What's Broken?

Poor change communication

Difficult to test outside of production

Little change validation

Manual validation doesn't scale

Silos of Excellence

Imagine a World...

Imagine a World...

Configs are easily shared

Imagine a World...

Configs are easily shared

All changes are automatically tested

Imagine a World...

Configs are easily shared

All changes are automatically tested

Tests include servers and apps

Imagine a World...

Configs are easily shared

All changes are automatically tested

Tests include servers and apps

Tests must pass to reach approvers

Imagine a World...

Configs are easily shared

All changes are automatically tested

Tests include servers and apps

Tests must pass to reach approvers

More drinks with umbrellas

NetDevOps Toolbox

NetDevOps Toolbox

Git – Provides config management and collaboration

Automation Tools - Git

~~Source Code~~ text file repository

Automatic file revision/change management

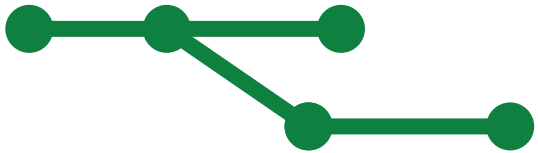
Built for teams to work on the same files

Easy to get started, lots of knobs for advanced users

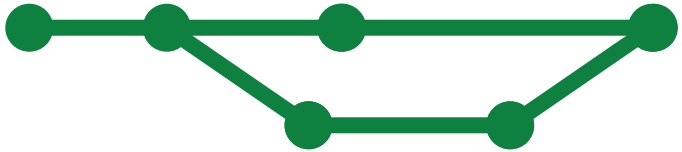
Automation Tools - Git Cont'd

Remote Repository – Central Server hosting files

Local Repository – Your local copy of the remote. Where you change things



Branch – Isolated copy for new features, risky changes



Merge – Apply branch changes to master repository copy
(see also: "Pull Request")

Automation Tools - Git Cont'd

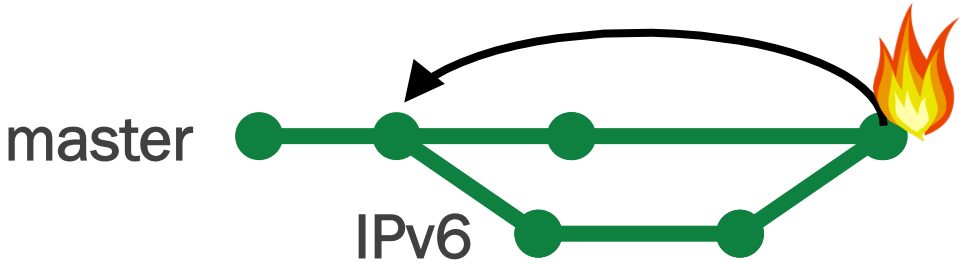
Create Changes in Branches

Use merge/pull request as an opportunity to review impact of changes

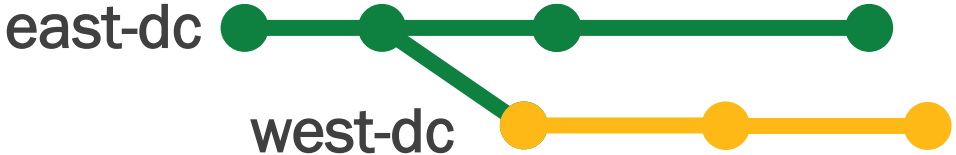
Merge and Review Changes

Automation Tools – Git Cont'd

Rollback failures after merge



Fork for new project/datacenter/office



NetDevOps Toolbox Alternatives

Git

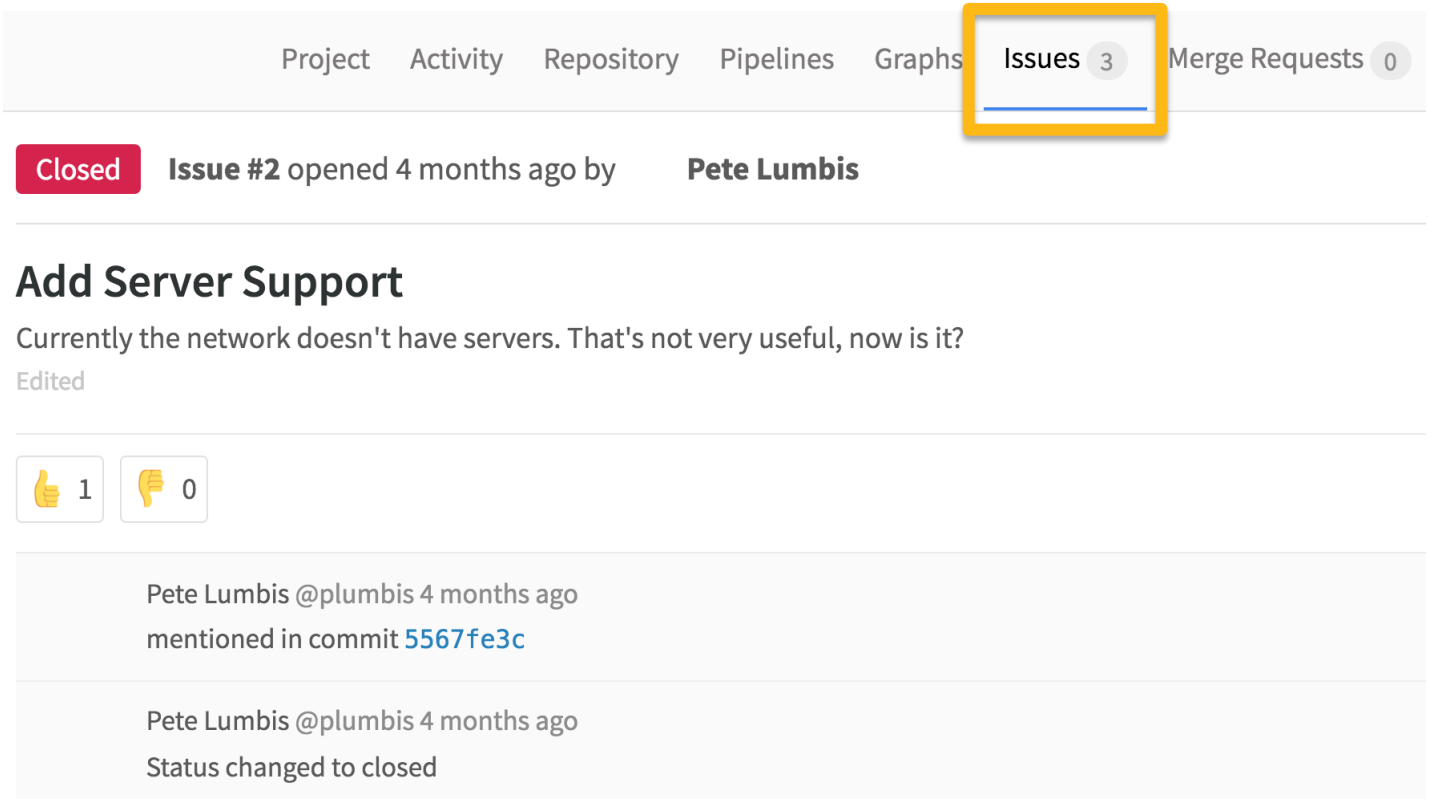
Atlassian Bitbucket

GitLab

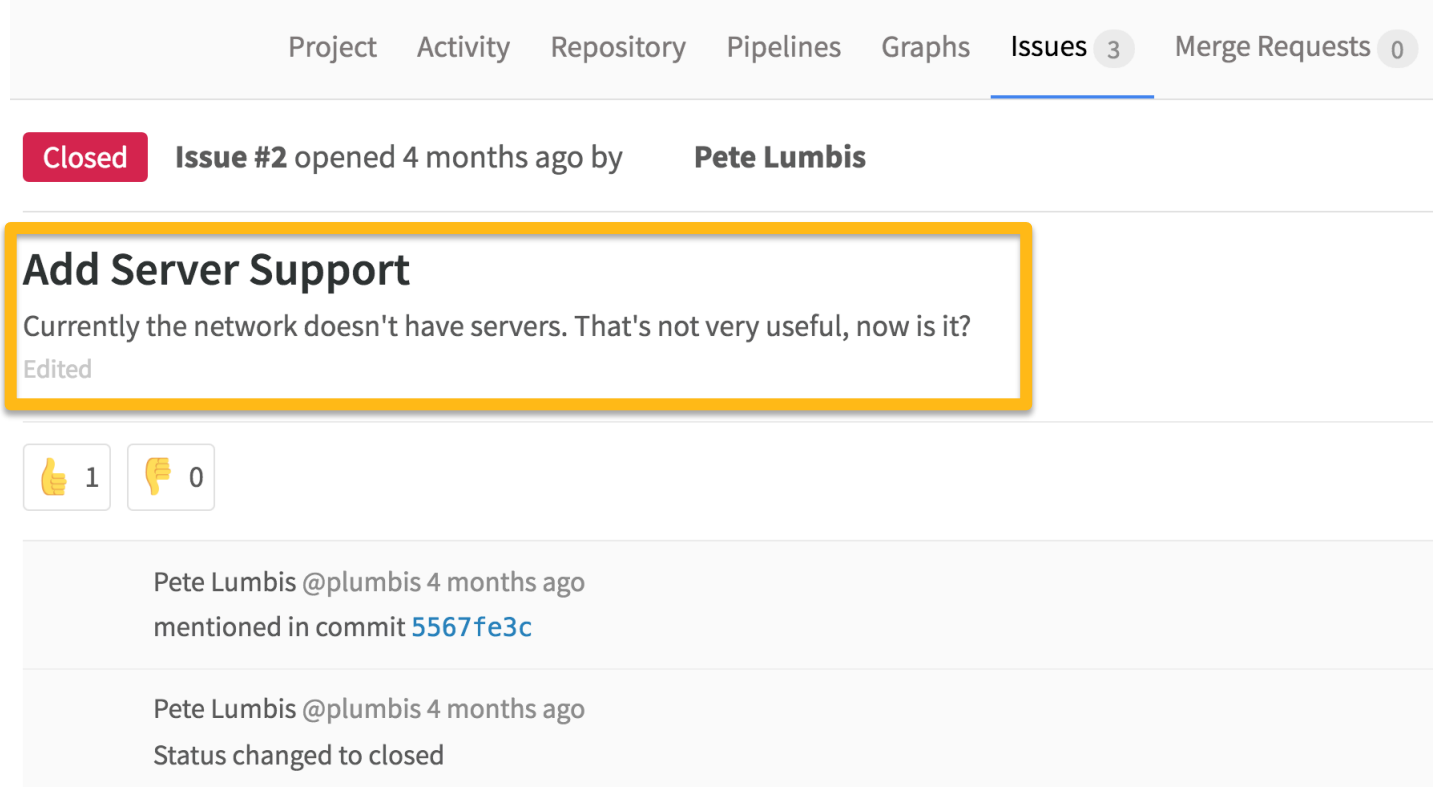
NetDevOps Toolbox

GitLab – Change management and automated testing

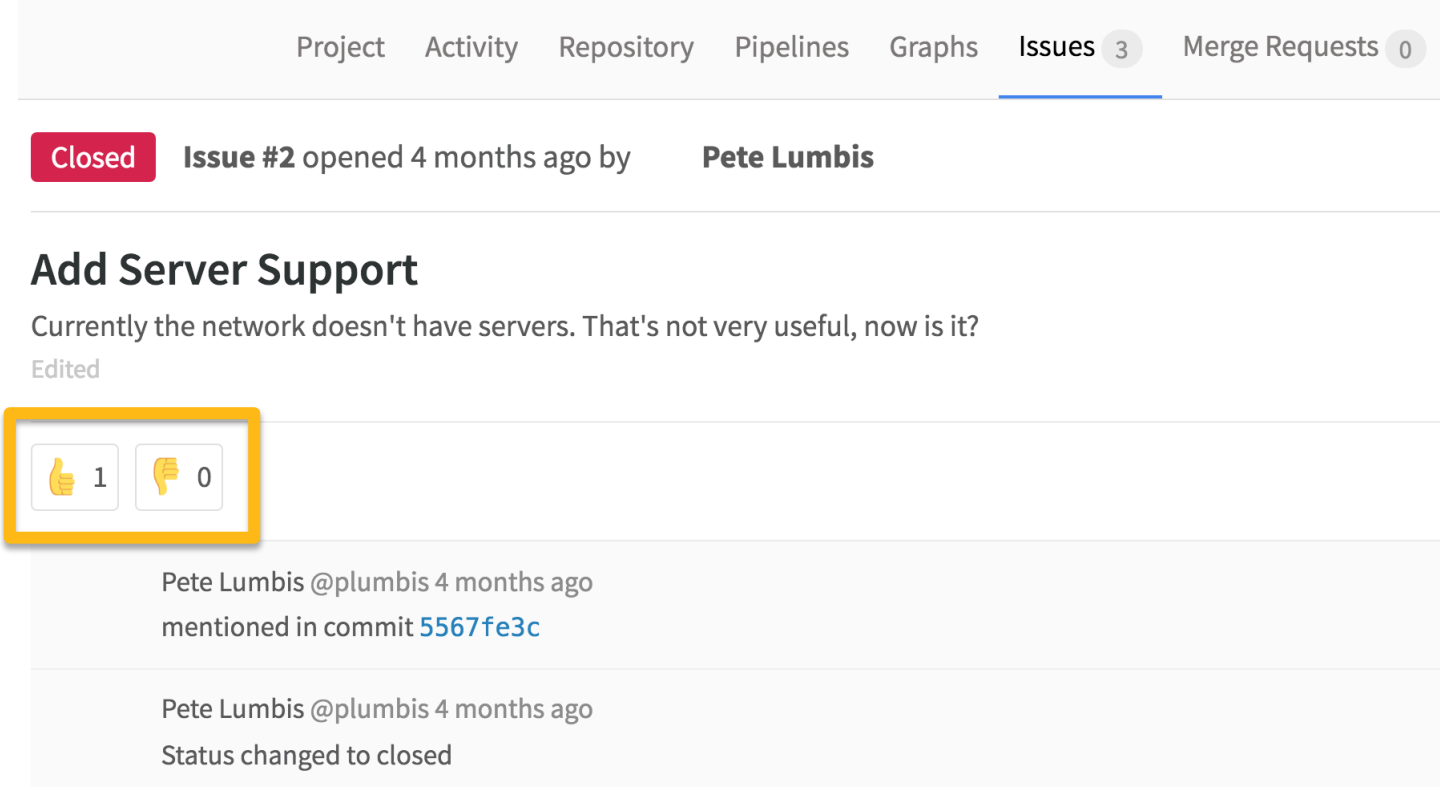
GitLab Ticketing



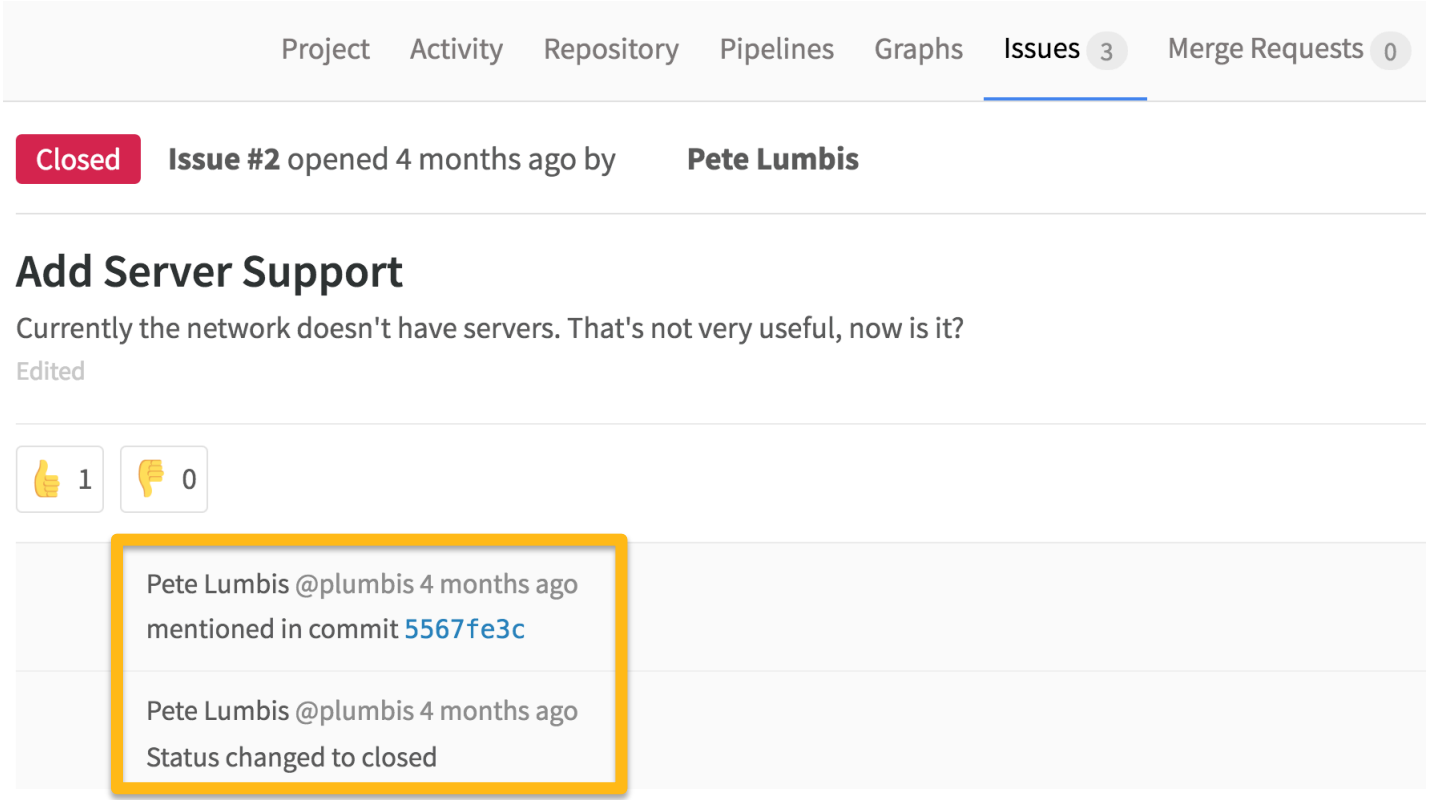
GitLab Ticketing



GitLab Ticketing



GitLab Ticketing



GitLab Changes

Authored by **Pete Lumbis** 4 months ago

Options

5567fe3c 1 parent 67018bfb master ...

Builds for 1 pipeline passed

Servers added as CumulusVx Nodes. Each server runs apache and BGP unnumbered. Resolves #2

Changes 5 Builds 3

Showing **5 changed files** with **11 additions** and **25 deletions**

Hide whitespace changes | Inline | Side-by-side

GitLab Changes

Authored by **Pete Lumbis** 4 months ago

Options

5567fe3c 1 parent 67018bfb master ...

Builds for 1 pipeline passed

Servers added as CumulusVx Nodes. Each server runs apache and BGP unnumbered. Resolves #2

Changes 5 Builds 3

Showing **5 changed files** with **11 additions** and **25 deletions**

Hide whitespace changes | Inline | Side-by-side

GitLab Changes

Authored by **Pete Lumbis** 4 months ago

Options

5567fe3c 1 parent 67018bfb master ...

Builds for 1 pipeline passed

Servers added as CumulusVx Nodes. Each server runs apache and BGP unnumbered. Resolves #2

Changes 5 Builds 3

Showing **5 changed files** with **11 additions** an

```
49  46      leaf4:
50  47        lo:
51  48          ipv6: "fd::1:4/128"
52      -       ipv4: "172.16.0.44/32"
    49  +       ipv4: "172.16.0.4/32"
```

NetDevOps Toolbox

Virtual Box – Hypervisor for lab network + server VMs

Vagrant – Simplifies large Virtual Box environments

NetDevOps Toolbox Alternatives

Virtual Box

vmware

Microsoft
Hyper-V

KVM

NetDevOps Toolbox

Ansible – Applies configuration and manages automation

Automation Tools - Ansible

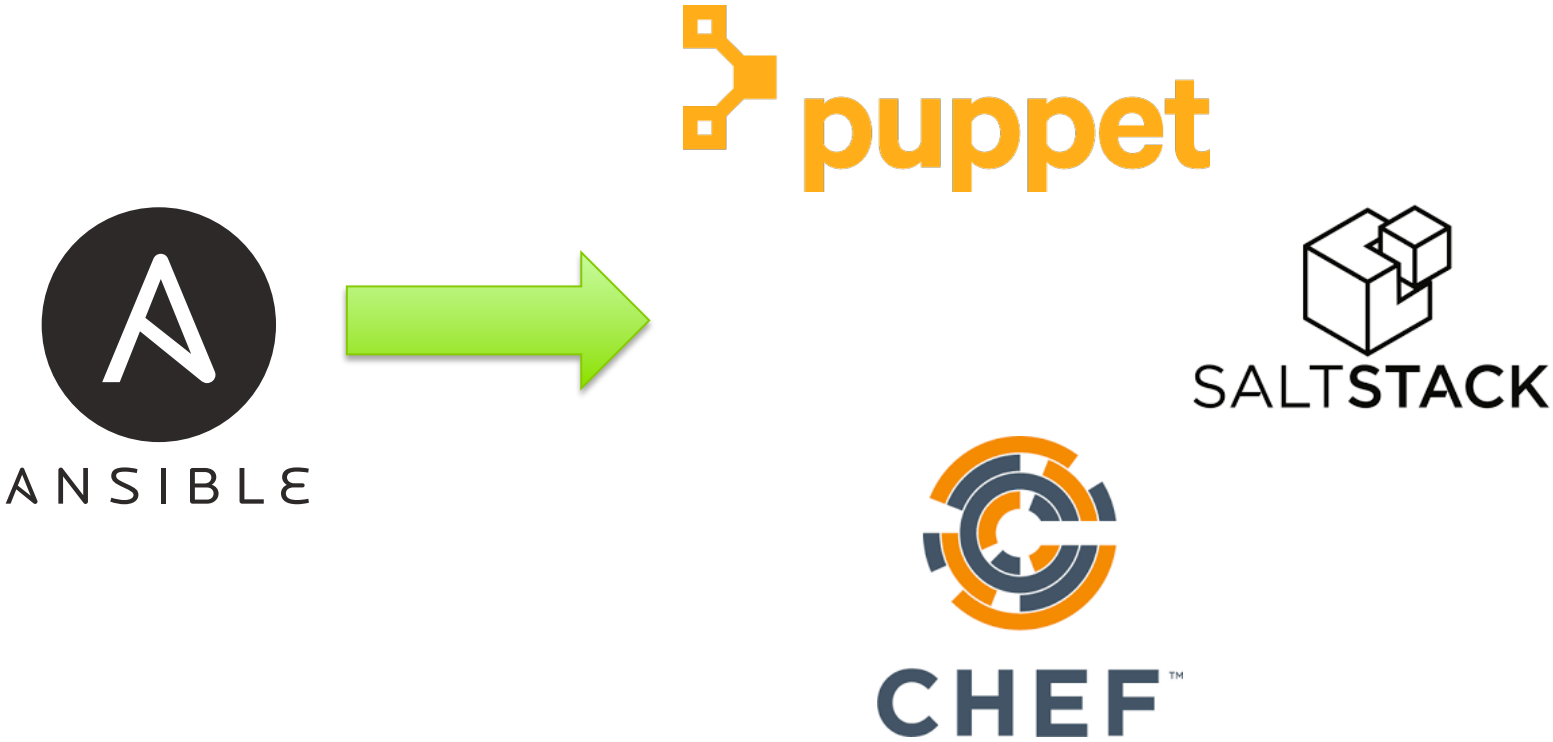
Configuration Management Software

Manages SSH connections

Large collection of libraries for common tasks

Human Readable

NetDevOps Toolbox Alternatives



NetDevOps Toolbox

Git

GitLab

Virtual Box

VAGRANT

ANSIBLE

Behave – English based testing language

Behave Overview

```
plumbis:validation plumbis$ behave
Feature: Validate BGP # bgp.feature:1

  Scenario: Check BGP Neighbors     # bgp.feature:3
    Given BGP is enabled            # steps/bgp.py:219 1.760s
    When neighbors are configured   # steps/bgp.py:243 0.001s
    Then the neighbors should be up # steps/bgp.py:341 0.000s

Feature: Validate Interfaces are up and IPs are applied # interfaces.feature:1

  Scenario: Check interfaces are up  # interfaces.feature:3
    Given an interface is configured # steps/interfaces.py:156 1.889s
    Then the interfaces should be up # steps/interfaces.py:180 0.001s

Feature: Validate the webservers can be reached. # website.feature:1
  This will validate each server has apache2 configured and running.
  Then each server will try to reach every other server and fetch the index page
  Scenario: Validate Web Server Access    # website.feature:5
    Given a webserver is configured       # steps/website.py:225 0.596s
    When apache is running                # steps/website.py:237 0.577s
    Then the website should be accessible # steps/website.py:245 4.564s

3 features passed, 0 failed, 0 skipped
3 scenarios passed, 0 failed, 0 skipped
8 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m9.388s
```

Behave Overview

```
plumbis:validation plumbis$ behave
Feature: Validate BGP # bgp.feature:1

  Scenario: Check BGP Neighbors      # bgp.feature:3
    Given BGP is enabled             # steps/bgp.py:219 1.760s
    When neighbors are configured    # steps/bgp.py:243 0.001s
    Then the neighbors should be up # steps/bgp.py:341 0.000s

Feature: Validate Interfaces are up and IPs are applied # interfaces.feature:1

  Scenario: Check interfaces are up  # interfaces.feature:3
    Given an interface is configured # steps/interfaces.py:156 1.889s
    Then the interfaces should be up # steps/interfaces.py:180 0.001s

Feature: Validate the webservers can be reached. # website.feature:1
  This will validate each server has apache2 configured and running.
  Then each server will try to reach every other server and fetch the index page
  Scenario: Validate Web Server Access     # website.feature:5
    Given a webserver is configured        # steps/website.py:225 0.596s
    When apache is running                 # steps/website.py:237 0.577s
    Then the website should be accessible # steps/website.py:245 4.564s

3 features passed, 0 failed, 0 skipped
3 scenarios passed, 0 failed, 0 skipped
8 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m9.388s
```

```
plumbis:validation plumbis$ behave
Feature: Validate BGP # bgp.feature:1

  Scenario: Check BGP Neighbors      # bgp.feature:3
    Given BGP is enabled             # steps/bgp.py:196 0.579s
    When neighbors are configured    # steps/bgp.py:220 0.001s
    Then the neighbors should be up # steps/bgp.py:317 0.000s
      Assertion Failed: spine1 peer swp1 not Established. Current state: Idle


Feature: Validate Interfaces are up and IPs are applied # interfaces.feature:1

  Scenario: Check interfaces are up  # interfaces.feature:3
    Given an interface is configured # steps/interfaces.py:138 0.759s
    Then the interfaces should be up # steps/interfaces.py:162 0.000s
      Assertion Failed: Interface swp1 on spine1 is in state ADMDN


Feature: Validate the webservers can be reached. # website.feature:1
  This will validate each server has apache2 configured and running.
  Then each server will try to reach every other server and fetch the index page
  Scenario: Validate Web Server Access     # website.feature:5
    Given a webserver is configured        # steps/website.py:124 0.233s
    When apache is running                 # steps/website.py:136 0.164s
    Then the website should be accessable # steps/website.py:144 0.895s


Failing scenarios:
  bgp.feature:3  Check BGP Neighbors
  interfaces.feature:3  Check interfaces are up

1 feature passed, 2 failed, 0 skipped
1 scenario passed, 2 failed, 0 skipped
6 steps passed, 2 failed, 0 skipped, 0 undefined
Took 0m2.632s
```

Behave Overview

Human readable policy checking

Calls python code under the covers

Network engineers are policy experts

Treat it as a back office application

Works on network and systems

NetDevOps Toolbox Alternatives

Beaker

puppet

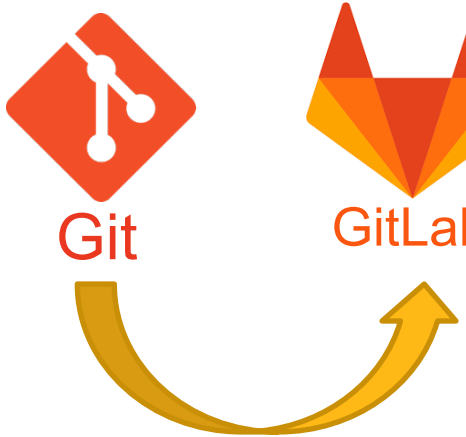
Matt Oswalt's ToDD

/?

KitchenCI

NetDevOps Toolbox – How It Happens

NetDevOps Toolbox – How It Happens



Configuration Saved to GitLab (`git push`)

Virtual Box

VAGRANT

ANSIBLE

NetDevOps Toolbox – How It Happens

Git

GitLab

Virtual Box

VAGRANT

ANSIBLE

GitLab Launches Lab
with new configs

NetDevOps Toolbox – How It Happens

Vagrant configures lab with Ansible

NetDevOps Toolbox – How It Happens

GitLab runs Behave tests against infrastructure

NetDevOps Toolbox – Automated Validation

GitLab executes behave checks automatically

Only checks that pass are merged in for others to see

Status	Commit
passed	#3753351 master f1a4ebeb latest Correcting BGP configuration mistake
failed	#3753256 master 66be74eb Modifying BGP peer settings

NetDevOps Toolbox – Automated Validation

failed Build **#2513447** for commit **66be74eb** from `master` by **@plumbis** about an hour ago

```
Feature: Validate BGP # bgp.feature:1

  Scenario: Check BGP Neighbors     # bgp.feature:3
    Given BGP is enabled            # steps/bgp.py:219 1.777s
    When neighbors are configured   # steps/bgp.py:243 0.001s
    Then the neighbors should be up # steps/bgp.py:341 0.000s

Feature: Validate Interfaces are up and IPs are applied # interfaces.feature:1

  Scenario: Check interfaces are up  # interfaces.feature:3
    Given an interface is configured # steps/interfaces.py:156 1.947s
    Then the interfaces should be up # steps/interfaces.py:180 0.001s

Feature: Validate the webservers can be reached. # website.feature:1
  This will validate each server has apache2 configured and running.
  Then each server will try to reach every other server and fetch the index page
  Scenario: Validate Web Server Access    # website.feature:5
    Given a webserver is configured       # steps/website.py:225 0.609s
    When apache is running                # steps/website.py:237 0.583s
    Then the website should be accessible # steps/website.py:245 5.181s
      Assertion Failed: Error on server1 trying to access http://10.0.0.2 : Status code was not [200]: Request failed
: <urlopen error [Errno 101] Network is unreachable>



Failing scenarios:
  website.feature:5  Validate Web Server Access
```

NetDevOps Toolbox – Automated Validation

Modifying BGP peer settings

Changes 1 Builds 2

Showing **1 changed file** with **0 additions** and **1 deletions**

roles/spines/templates/Quagga.conf.j2

```
...  ...  @@ -25,7 +25,6 @@ router bgp {{ bgpvars.asn }}
25   25      neighbor fabric timers 1 3
26   26      neighbor fabric timers connect 3
27   27      neighbor fabric remote-as external
28        -  neighbor fabric capability extended-nexthop
29   28     {### Configure prefix lists ###}
30   29      {% if bgpvars.fabric_prefix_list_out is defined %}
31   30      neighbor fabric prefix-list {{bgpvars.fabric_prefix_list_out}} out
...  ...
```

Wrap Up

Wrap Up

There’s a better way!

Wrap Up

There’s a better way!

Infrastructure as code

Wrap Up

There’s a better way!

Infrastructure as code

Automate all the things

Wrap Up

There's a better way!

Infrastructure as code

Automate all the things

Testing must include applications

Wrap Up

There's a better way!

Infrastructure as code

Automate all the things

Testing must include applications

Push your vendor. Vote with $$

Questions?

Thank You!

@PeteCCDE
http://gitlab.com/plumbis/interop-2016