MONASH University

Developing and Operating MyTardis at Monash University

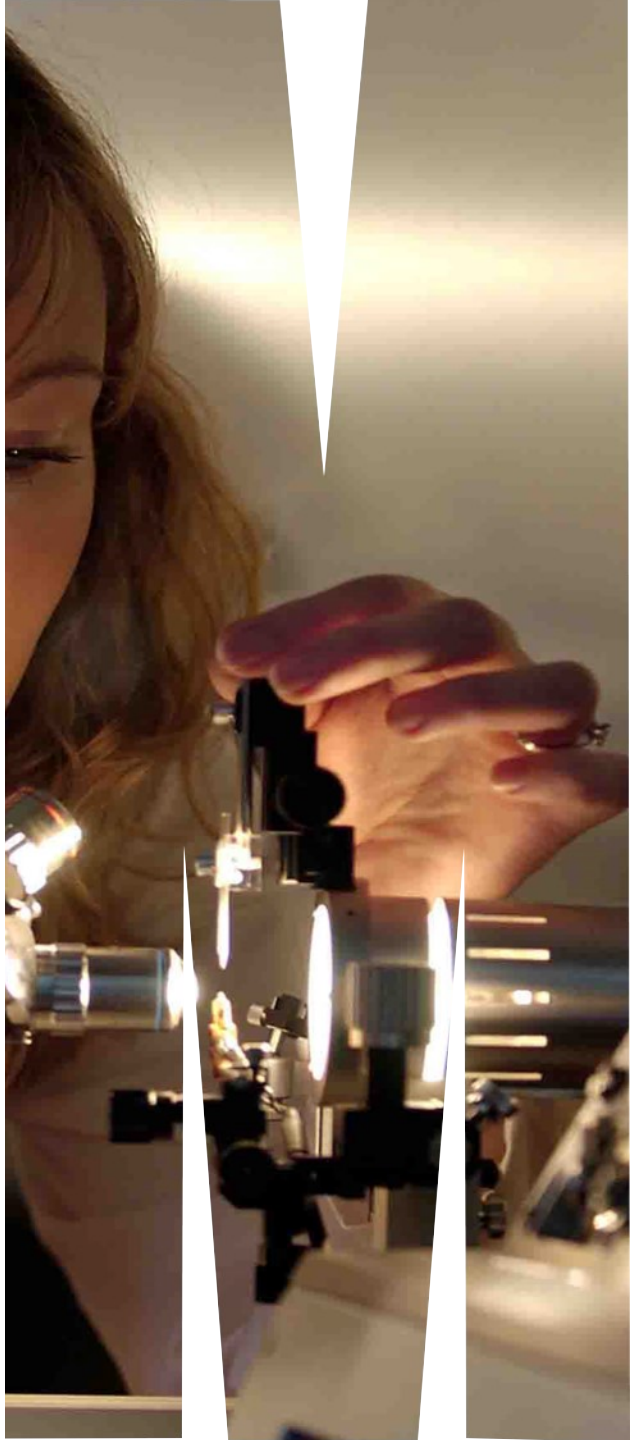
James Wettenhall and Manish Kumar

Monash eResearch Centre

https://www.monash.edu/eresearch

In collaboration with:

Amr Hassan, Wojtek Goscinski, Samitha Amarapathy, Keith Schulze, Chaitanya Manapragada, Grischa Meyer, Steve Androulakis, Jason Rigby, Andrew Perry, Anitha Kannan, Lance Wilson, Andrew Mehnert, Andrew Janke



Objectives

Developing MyTardis

MyTardis Codebase

Contributing to Mytardis

Maintaining MyTardis

Continuous Integration

Operating MyTardis

Infrastructure as code

Managing Mytardis deployment at Monash

Developing MyTardis at Monash University

MyTardis

- Open source project hosted at https://github.com/mytardis/mytardis
- Licensed under GPL V3.0
- Latest release is v3.9.0
- Documentation http://mytardis.readthedocs.org
  - User documentation
    - Organising data, Acessing data, sharing and publishing data
  - Administrator documentation
    - Installation, Settings, Migration, enable/disable apps
  - Developer documentation
    - Architecture, Tests, source code

Developing MyTardis at Monash University

Contributing to MyTardis

- Feature requests
- Reporting Bugs
  - https://github.com/mytardis/mytardis/issues
- Submitting Pull Request
  - https://github.com/mytardis/mytardis/blob/develop/CONTRIBUTING.rst
- Help users and contributors via Slack/Github
  - https://mytardis.slack.com/

User password change and reset functions via UI and API auth-rewrite feature ux improvement
#324 opened on 2 Dec 2013 by grischa future

Redirect to next page is not working bug
#1196 opened on 26 Jul by manishkumr 4.0

Tell elasticsearch tokenizer to split on underscores
#1285 by wettenhj was merged on 10 Sep

Developing MyTardis at Monash University

Maintaining MyTardis

Automated Python Security and Dependency Updates

Dependencies defined in **requirements-base.txt** 37

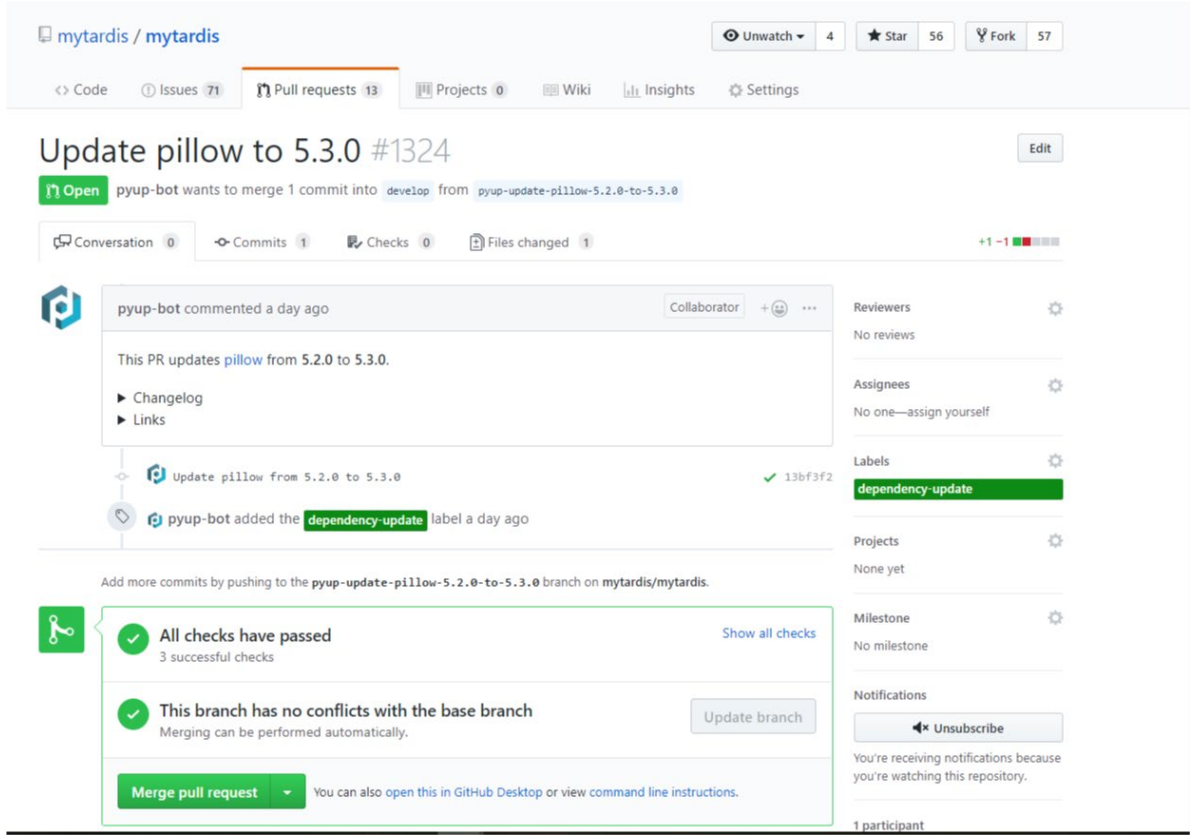
paramiko / **paramiko**	2.4.2
celery / **py-amqp** amqp	1.4.9
waylan / **beautifulsoup** beautifulsoup4	4.6.3
celery / **billiard**	3.3.0.23
mozilla / **bleach**	2.1.4
celery / **celery**	3.1.26.post2
django / **django**	1.11.16
django-extensions / **django-extensions**	1.8.1
django-haystack / **django-haystack**	2.8.1
mjumbewu / **django-jstemplate**	1.3.8
kevin1024 / **django-npm**	1.0.0
macropin / **django-registration** django-registration-redux	2.4

Developing MyTardis at Monash University

Maintaining MyTardis

Automated Python Security and Dependency Updates

- https://pyup.io/



Developing MyTardis at Monash University

Maintaining MyTardis

Enforcing Coding Standard and error detection

- https://www.pylint.org/

```
(venv) C:\Users\Manish Kumar\Documents\GitHub\MK\mytardis>pylint tardis.apps.social_auth
Using config file C:\Users\Manish Kumar\Documents\GitHub\MK\mytardis\.pylintrc


------------------------------------
Your code has been rated at 10.00/10
```

Developing MyTardis at Monash University

Maintaining MyTardis

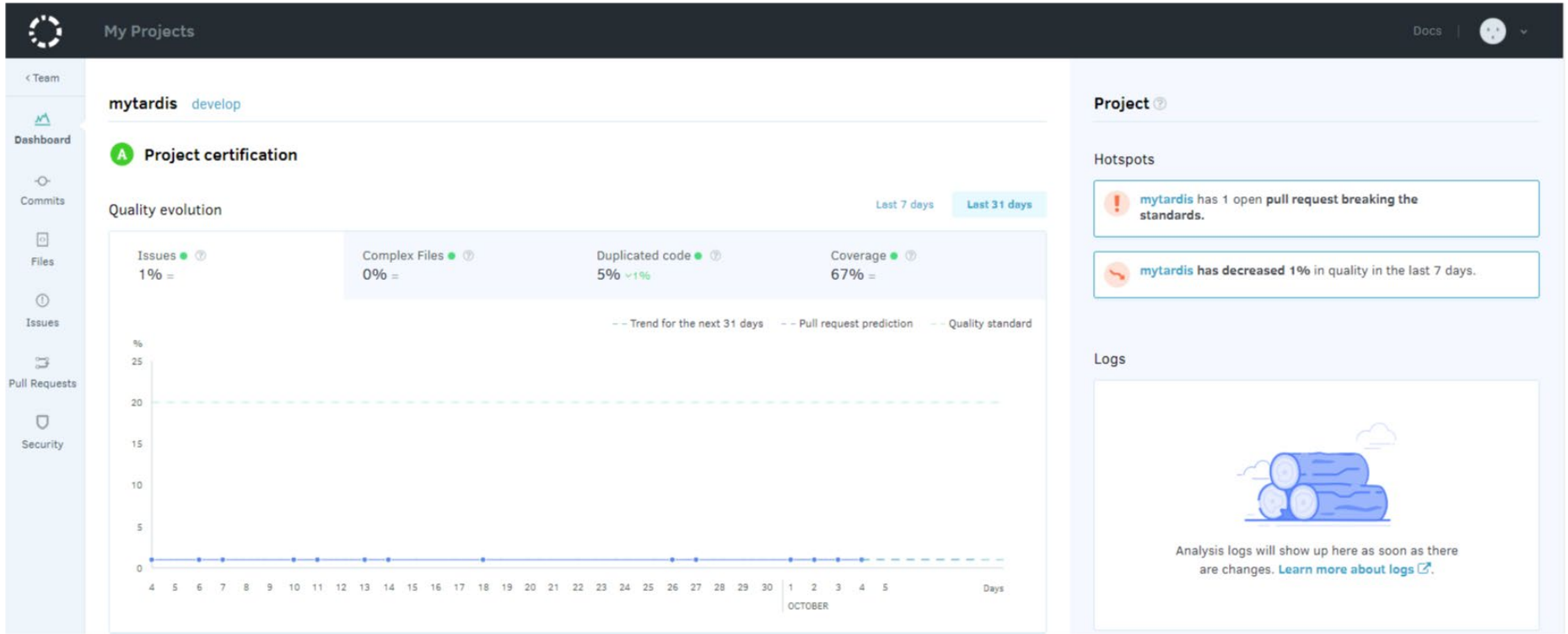
Code review and code quality monitoring

- **Codacy** https://www.codacy.com/

Developing MyTardis at Monash University

Maintaining MyTardis

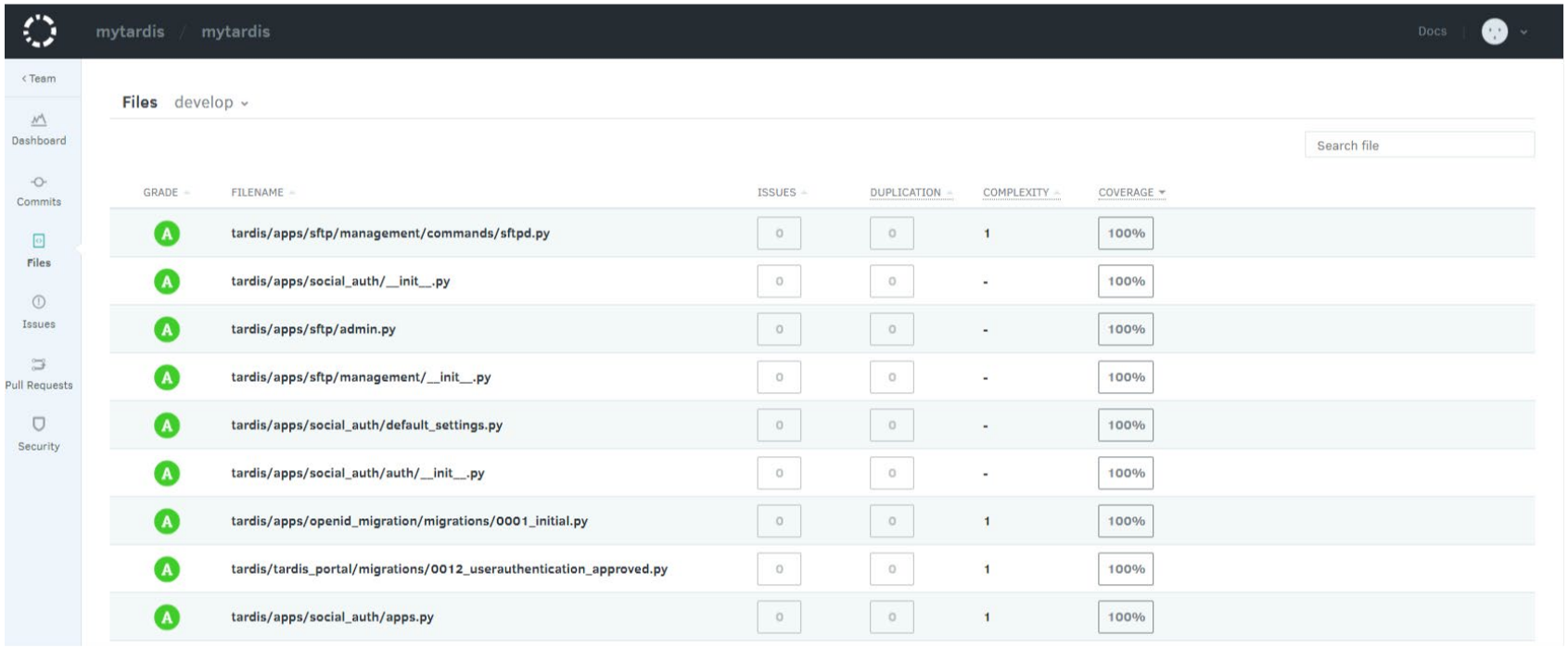
Code review and code quality monitoring



Developing MyTardis at Monash University

Maintaining MyTardis

Code review and code quality monitoring



Developing MyTardis at Monash University

Maintaining MyTardis

Automated testing

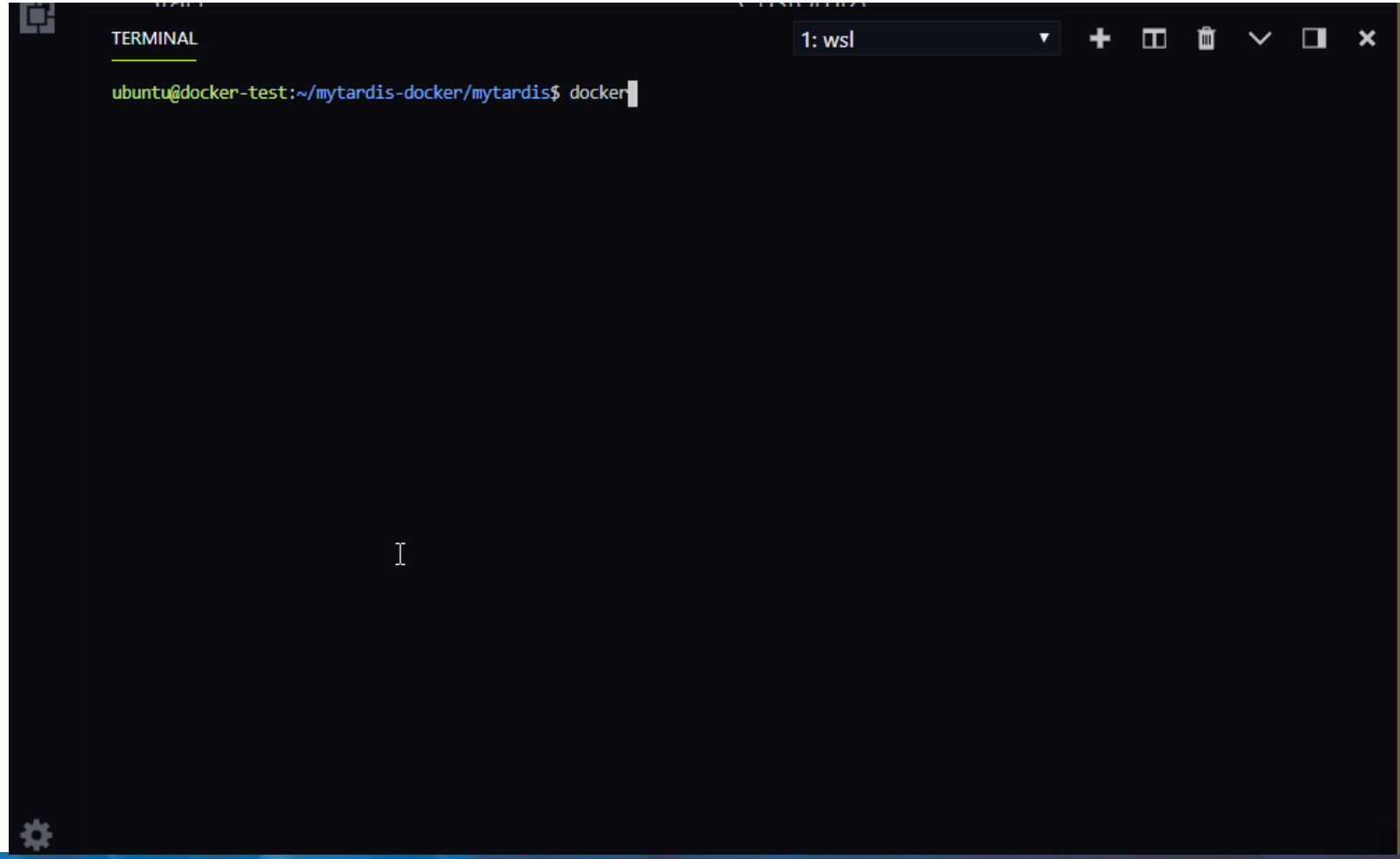
- Unit Tests
  - ./test.py

Developing MyTardis at Monash University

Maintaining MyTardis

Automated testing

- Unit Tests
  - ./test.py



Developing MyTardis at Monash University

Maintaining MyTardis

Automated testing

- **BDD**
  - **./test.py behave**

Developing MyTardis at Monash University

Maintaining MyTardis

Automated testing

- **BDD**
  - **./test.py behave**

TERMINAL

```
ubuntu@docker-test:~/mytardis-docker/mytardis$
```

Developing MyTardis at Monash University

Maintaining MyTardis

Automated testing

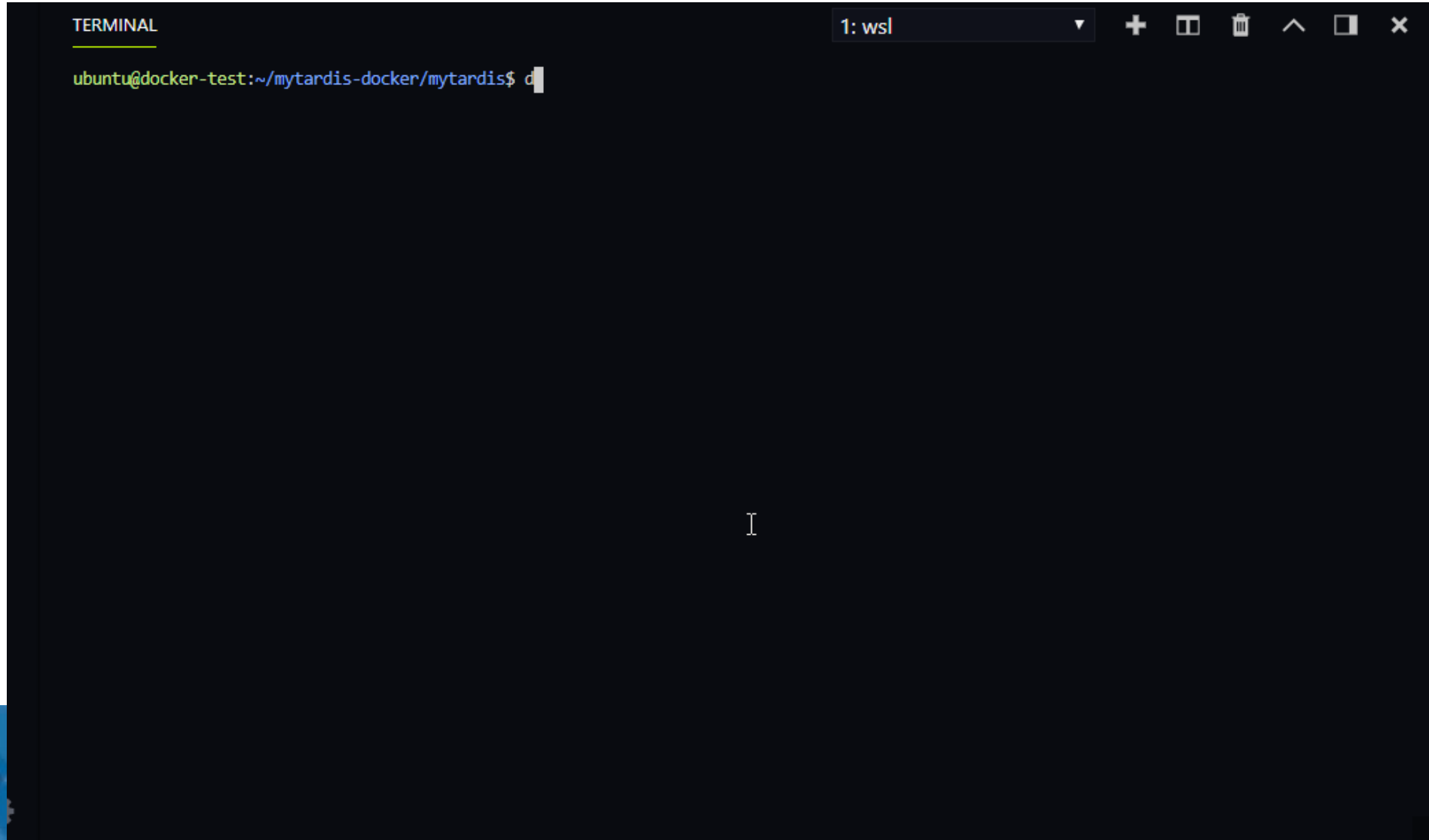
- **Qunit tests**
  - **Npm test**

Developing MyTardis at Monash University

Maintaining MyTardis

Automated testing

- **Qunit tests**
  - **Npm test**

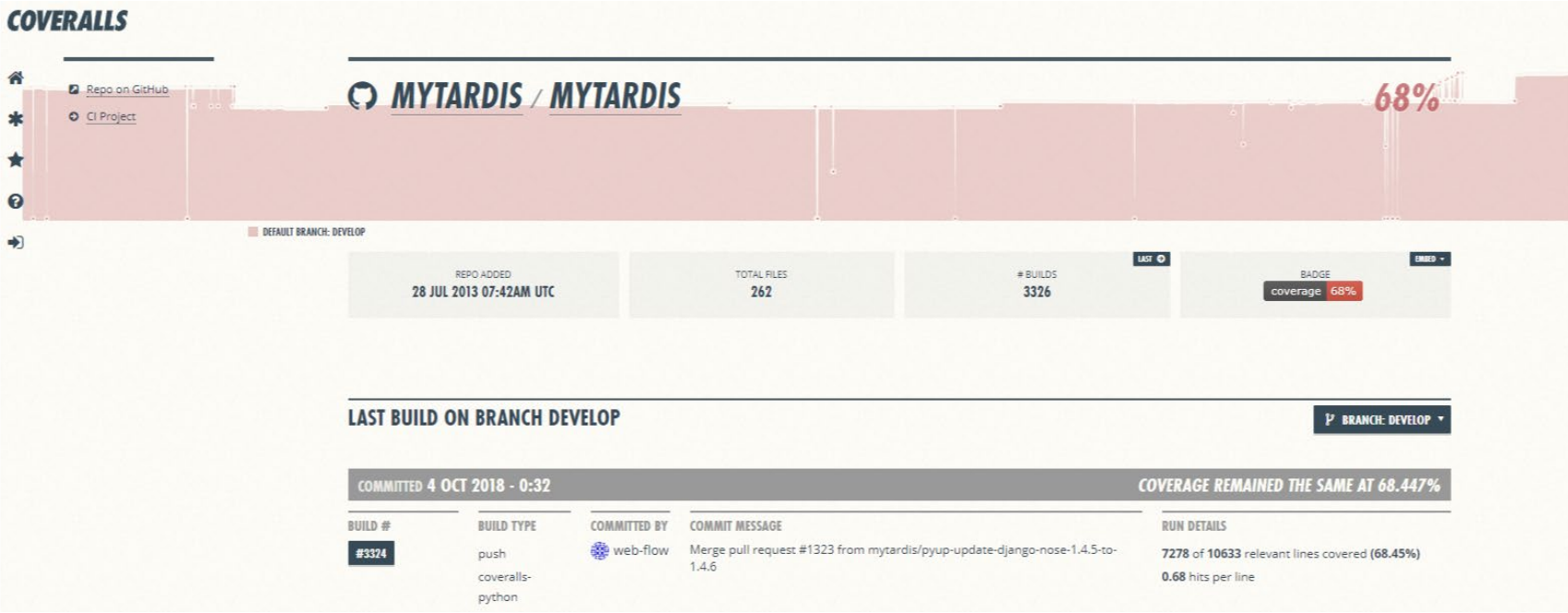


Developing MyTardis at Monash University

Maintaining MyTardis

Test coverage

- https://coveralls.io/



Developing MyTardis at Monash University

Continuous Integration

Developing MyTardis at Monash University

Continuous Integration

- How we do it
  - Developers check out code into their private workspaces
  - When done, commit the changes to the repository
  - The CI server monitors the repository and checks out changes when they occur
  - The CI server builds the system and runs unit and integration tests

Developing MyTardis at Monash University

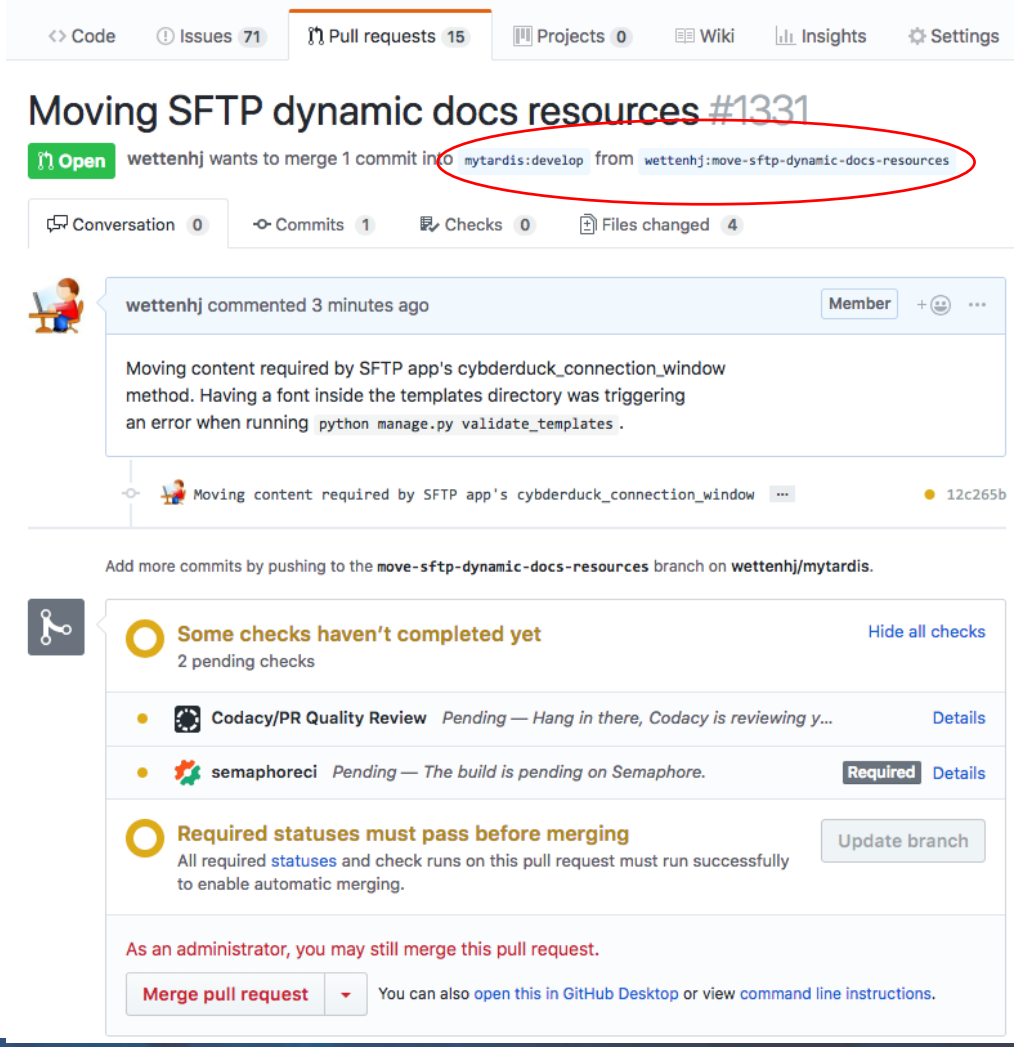
Continuous Integration

- How we do it
  - The CI server releases deployable artefacts for testing
  - The CI server assigns a build label to the version of the code it just built
  - The CI server informs the team of the successful build
  - If the build or tests fail, the CI server alerts the team
  - The team fixes the issue at the earliest opportunity
  - Continue to continually integrate and test throughout the project

Developing MyTardis at Monash University

Continuous Integration

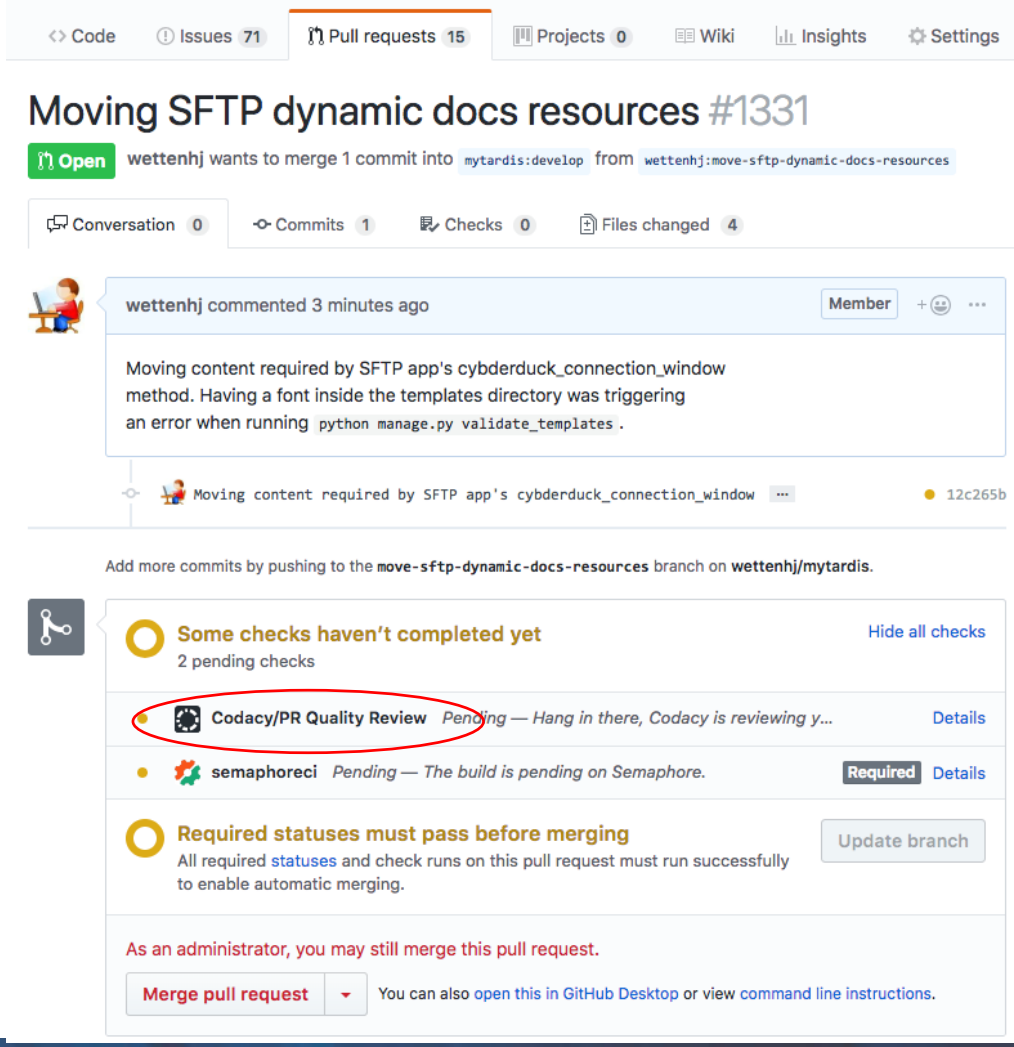
- Semaphore
  - PR submitted
  - Codacy check
  - Semaphore build
  - PR ready to be merged



Developing MyTardis at Monash University

Continuous Integration

- Semaphore
  - PR submitted
  - Codacy check
  - Semaphore build
  - PR ready to be merged



Developing MyTardis at Monash University

Continuous Integration

- Semaphore
  - PR submitted
  - Codacy check
  - Semaphore build
  - PR ready to be merged

mytardis / mytardis PUBLIC

Insights Project settings

Branch: pull-request-1331

View pull request

Moving content required by SFTP app's cybderduck_connection_window

method. Having a font inside the templates directory was triggering

an error when running "python manage.py validate_templates".

wettenhj created a build based on revision 12c265b 12 minutes ago

Pull Request #1331

PASSED build 1 11:29

all tests 11:29

1 more setup command +

Launch SSH

```
docker-compose -f docker-test.yml run -e TEST_TYPE=memory -e CODACY_PROJECT_TOKEN=$CODACY_PROJECT_TOKEN -e COVERALLS_REPO_TOKEN=$COVERALLS_REPO_TOKEN
-e CI_NAME=SemaphoreCI -e CI_BUILD_NUMBER=$SEMAPHORE_BUILD_NUMBER -e CI_BRANCH=$BRANCH_NAME -e PULL_REQUEST_NUMBER=$PULL_REQUEST_NUMBER test
```

02:51

```
docker-compose -f docker-test.yml run -e TEST_TYPE=pg test
```

00:43

```
docker-compose -f docker-test.yml run -e TEST_TYPE=behave test
```

00:34

```
docker-compose -f docker-test.yml run -e TEST_TYPE=mysql test
```

01:01

```
docker-compose -f docker-test.yml run -e TEST_TYPE=pylint test
```

02:57

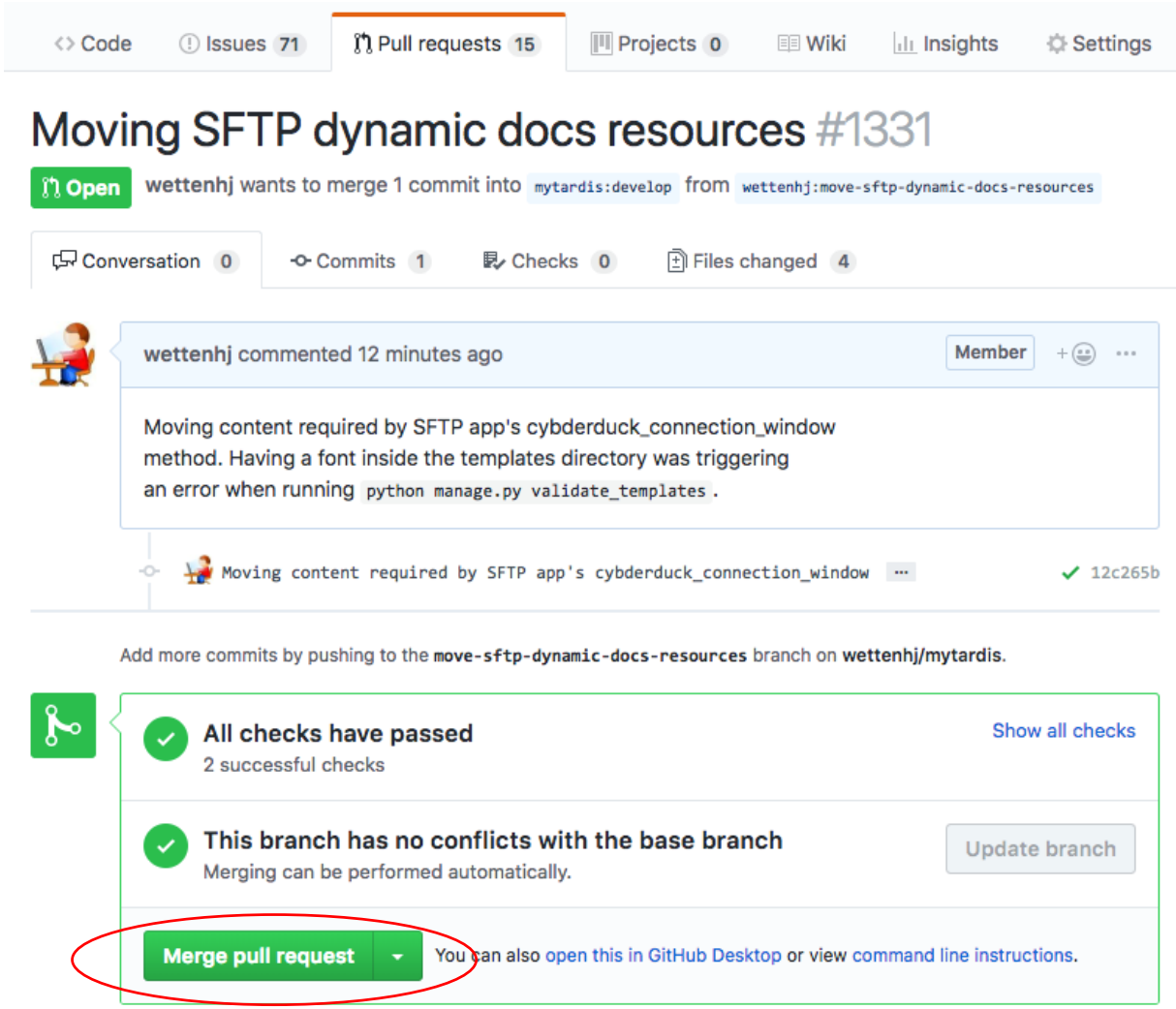
```
for image in mytardis/mytardis-base mytardis/mytardis-run mytardis/mytardis-test; do if [ ! -v PULL_REQUEST_NUMBER ]; then if [ ! "$(git tag --
points-at $BRANCH_NAME)" == "" ]; then tag=$(git tag --points-at $BRANCH_NAME); docker tag $image $image:$tag; docker push $image:$tag; elif [
"$BRANCH_NAME" == "master" ]; then docker push $image; elif [ "$BRANCH_NAME" == "develop" ]; then tag=develop; docker tag $image $image:$tag; docker
push $image:$tag; fi; fi; done
```

00:00

Developing MyTardis at Monash University

Continuous Integration

- Semaphore
  - PR submitted
  - Codacy check
  - Semaphore build
  - PR ready to be merged



Developing MyTardis at Monash University

Continuous Integration

- Semaphore

PUBLIC REPOSITORY

mytardis/mytardis-run

Last pushed: 4 hours ago

Repo Info | Tags | Collaborators | Webhooks | Settings

Tag Name	Compressed Size	Last Updated
develop	434 MB	4 hours ago
v3.9.0	341 MB	2 days ago
latest	341 MB	2 days ago
v4.0.0-RC9	433 MB	10 days ago
v4.0.0-RC8	432 MB	24 days ago
v4.0.0-RC6	432 MB	a month ago
v4.0.0-RC5	432 MB	a month ago
v4.0.0-RC3	429 MB	a month ago
v4.0.0-RC1	430 MB	a month ago

Developing MyTardis at Monash University

MyTardis Release process

- Minor release(Monthly)
- Major release(as needed)
- Security and patch release(anytime within release cycle)

Day of the Month	Tasks
8th	Feature Freeze day: •Announce creation of series-x.y branch and feature freeze on #general MyTardis Slack channel •Create meta issue to track release on http://github.com/mytardis/release issues •Merge any final Pull Requests labelled with this milestone into develop that have been approved (i.e., reviewed and have approval status). •Create series-x.y branch from develop •Tag and release X.Y.0-rc1
11th	Update Store.Monash test stack with latest X.Y.0-rc release.
14-24th (as needed)	Tag and release subsequent RCs as needed. Generally should aim for at least a couple.
22nd	Release day: •Cherry-pick any final patches for features in this release that have been merged into develop. •Ensure all tests on series-x.y are passing on Semaphore •Tag X.Y.0 version from series-x.y •Check that all tests for tag X.Y.0 pass •Create github release

Objectives

Developing MyTardis

MyTardis Codebase

Contributing to Mytardis

Maintaining MyTardis

Continuous Integration

Operating MyTardis

Infrastructure as code

Managing Mytardis deployment at Monash

Operating MyTardis at Monash University

Infrastructure as Code

- Nectar cloud(Based on OpenStack)
- Heat
- Salt

Operating MyTardis at Monash University

Infrastructure as Code

- Nectar cloud(Based on OpenStack)
  - provides
    - flexible scalable computing power
    - computing infrastructure
    - software and services

- Heat
- Salt

Operating MyTardis at Monash University

Infrastructure as Code

- Nectar cloud(Based on OpenStack)
- Heat
  - Heat template
  - Heat create stack
  - Heat update stack
- Salt

```yaml
heat_template_version: 2013-05-23

description: Simple template to deploy a single compute instance

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      image: cirros-0.3.3-x86_64
      flavor: m1.small
      key_name: my_key
      networks:
        - network: private-net
```

Operating MyTardis at Monash University

Infrastructure as Code

- Nectar cloud(Based on OpenStack)
- Heat
  - Heat template
  - Heat create stack
  - Heat update stack

```
python deploy_all.py --deployment qat --salt_master_ip=118.138.233.109 --ssh_key_name=cloud-nodes --
security_group=open-to-all --celery_n_servers 2 --celeryd_filters_n_servers 2 > qat-store-monash.json
```

```
heat stack-create qat-store-monash --template-file qat-store-monash.json
```

Operating MyTardis at Monash University

Infrastructure as Code

- Nectar cloud(Based on OpenStack)
- Heat
  - Heat template
  - Heat create stack
  - Heat update stack
- Salt

```
        "Web1": {
            "Properties": {
                "availability_zone": {
                    "Ref": "WebAvailabilityZone1"
                },
                "flavor": {
                    "Ref": "WebVMFlavour"
                },
                "image": {
                    "Fn::FindInMap": [
                        "OSNameToImageID",
                        {
                            "Ref": "WebOSName"
                        },
                        "ID"
                    ]
                },
                "key_name": {
                    "Ref": "WebKeyPair"
                },
                "networks": [],
                "security_groups": [
                    "open-to-all"
                ],
                "user_data": {
                    "Fn::Base64": "#!/bin/bash\napt-get update\napt-get re
 qat\nroles:\n- www\n\nEOF\nchmod 644 /etc/salt/grains\n\n\nSALT_MINION_
/salt/minion_id\necho '118.138.233.170 salt' >> /etc/hosts\nreboot\n"
                }
            },
            "Type": "OS::Nova::Server"
        }
```

Operating MyTardis at Monash University

Infrastructure as Code

- Nectar cloud(Based on OpenStack)
- Heat
- Salt
  - Configuration Management
    - Run commands on managed systems
    - Define re-usable configurations
    - Apply commands and configurations to specific systems

Operating MyTardis at Monash University

Infrastructure as Code

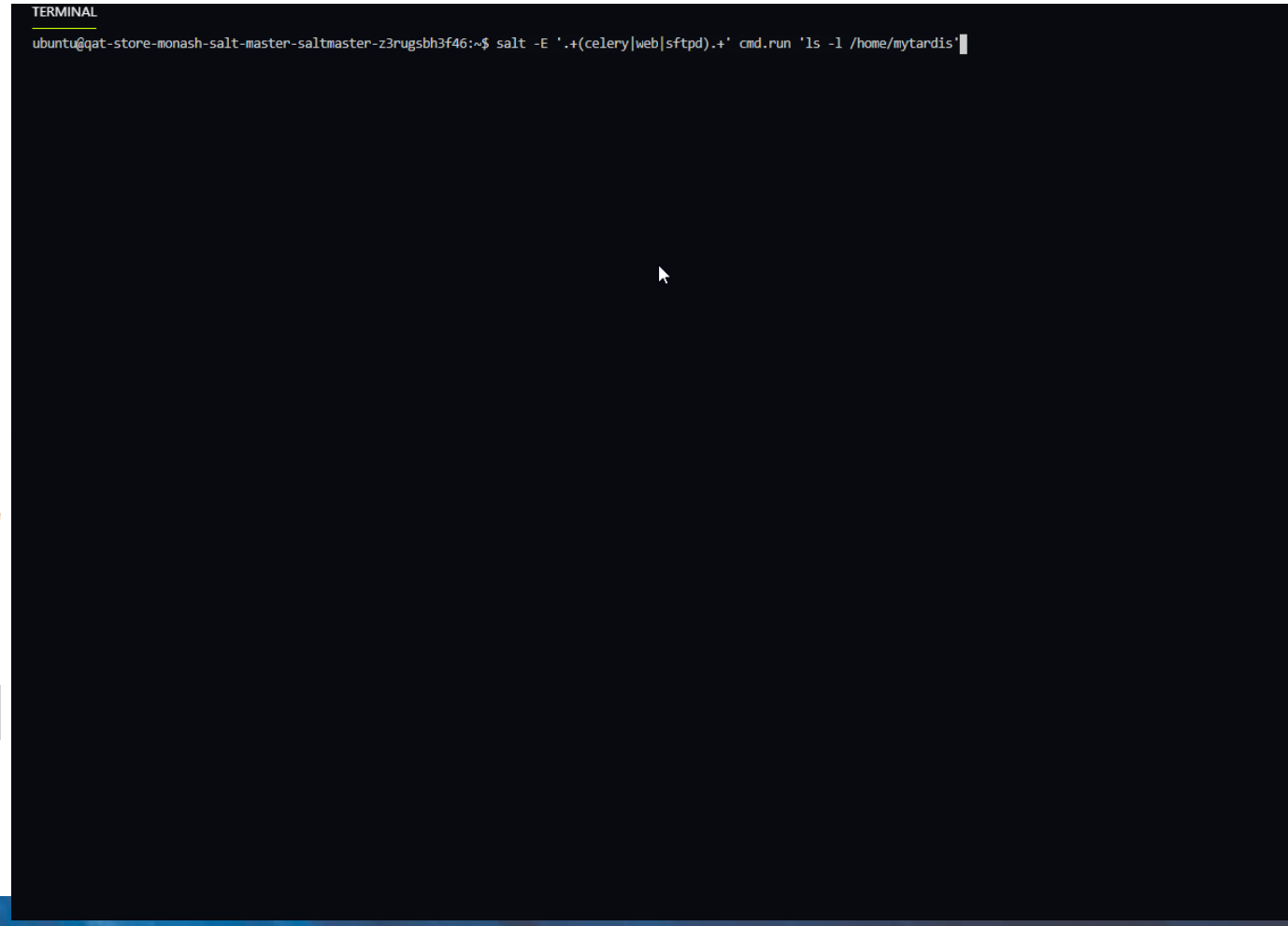
- Salt
  - Configuration Management
    - Run commands on managed systems
    - Define re-usable configurations
    - Apply commands and configurations to

```
salt -E '.+(celery|web|sftpd).+' cmd.run 'ls -l /home/mytardis'
```

target

module.function

arguments



Operating MyTardis at Monash University

Infrastructure as Code

- Salt
  - Configuration Management
    - Run commands on managed systems
    - Define re-usable configurations
      - Salt state
    - Apply commands and configurations to specific systems

Salt '*web*' state.apply

storedotstar-salt-states / networking / firewall / web.sls

```
http:
  iptables.append:
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - proto: tcp
    - save: True
    - require:
      - iptables: clean_iptables_ipv4

https:
  iptables.append:
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 443
    - proto: tcp
    - save: True
    - require:
      - iptables: clean_iptables_ipv4
```

Operating MyTardis at Monash University

Infrastructure as Code

- Nectar cloud(Based on OpenStack)
- Heat
- Salt
  - Configuration Management
    - Run commands on managed systems
    - Define re-usable configurations
    - Apply commands and configurations to specific systems

storedotstar-salt-states / top.sls

```
base:
  '*':
    - common.pkgs
    - common.timezone
    - ssh
    - storage
    - networking.hosts
    - networking.firewall.default
    - logging.rsyslog
    - logging.munin_node
    - admin.authorized_keys
  'G@roles:saltmaster':
    - logging.munin_master
  'G@roles:elasticsearch_loadbalancer':
    - haproxy
  'G@roles:elasticsearch':
    - database.elasticsearch
  'G@roles:postgresql*':
    - database.postgres_client
  'G@roles:postgresql_server_master':
    - database.postgres_server
  'G@roles:postgresql_server_slave':
    - database.postgres_server
  'G@roles:postgresql_pgpool':
    - database.postgres_pgpool
  'G@roles:haproxy':
    - haproxy
  'G@roles:www':
    - tardis.web
```

http://www.mytardis.org/

https://github.com/mytardis/mytardis

https://mytardis.readthedocs.io

store.star.help@monash.edu

MyTardis has been supported by:

MONASH University

nectar

UNSW SYDNEY

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

THE UNIVERSITY OF SYDNEY

THE UNIVERSITY OF WESTERN AUSTRALIA

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

THE UNIVERSITY of ADELAIDE

RDS www.rds.edu.au

ands AUSTRALIAN NATIONAL DATA SERVICE

ansto

National Imaging Facility

Australian Synchrotron