redhat.

OVIRT STDCI V2

Introduction for oVirt developers

Barak Korren
RHV DevOps – CI team lead
bkorren@redhat.com

WHAT IS STDCI

The premise of STDCI (CI separation of concerns)

DEVELOPERS CONFIGURE	INFRA TEAM TAKES CARE OF
How to build source code	Jenkins
How to test source code	Jobs
Target operating systems	Slaves
Target architectures	Plugins
Target oVirt versions	Servers
	Networks

BUT...

STDCI V1 didn't quite get this right

DEVELOPERS CONFIGURE

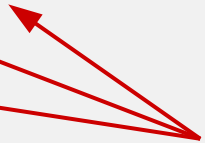
How to build source code

How to test source code

Target operating systems

Target architectures

Target oVirt versions



Configured in the 'jenkins' repo

INFRA TEAM TAKES CARE OF

Jenkins

Jobs

Slaves

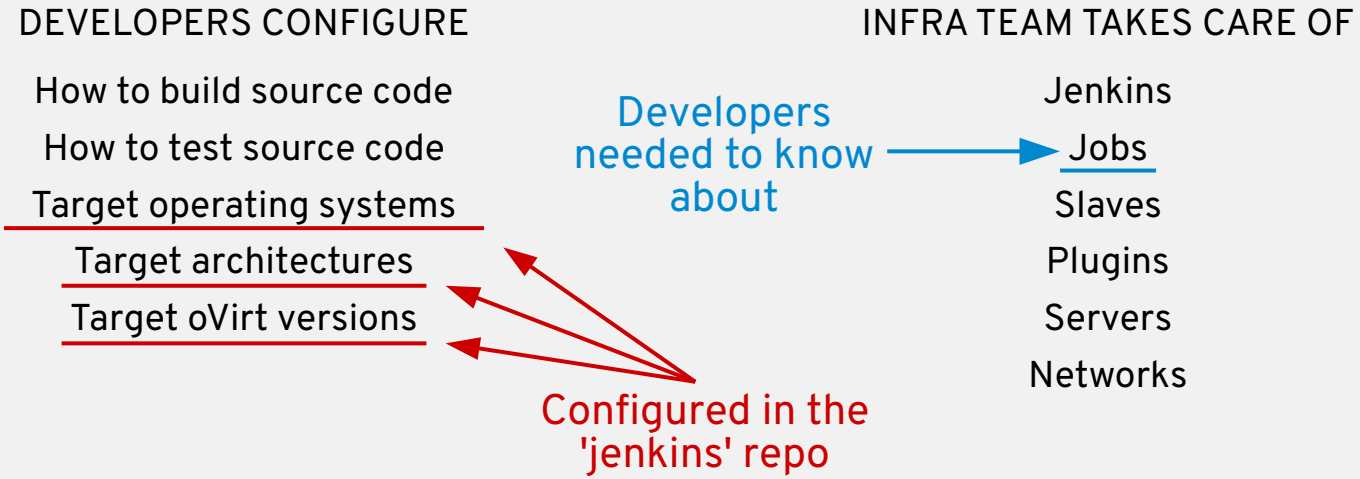
Plugins

Servers

Networks

BUT...

STDCI V1 didn't quite get this right



STDCI V2

The 'big' change

A new YAML-based DSL file in the project's source repo that defines:

- The project's target distributions
- The project's target architectures
- The oVirt versions particular branches target

All configuration is optional – with no file projects will be built on el7/x86_64 and not sent to the change/release queues.

```
|-- automation
|   |-- build-artifacts.packages
|   |-- build-artifacts.sh
|   |-- check-patch.packages
|   |-- check-patch.sh
|   `-- README.md
|-- automation.yaml
|-- build.sh
|-- examples
|   |-- cluster_upgrade.yml
|   |-- manageiq.yml
|   |-- ovirt_image_template.yml
|   |-- ovirt_infra_vars.yml
|   |-- ovirt_infra.yml
|   |-- ovirt_vm_infra.yml
|   `-- passwords.yml
|-- LICENSE
|-- meta
|   `-- main.yml
|-- ovirt-ansible-roles.spec.in
```

STDCI V2

But wait, there is more!

The new DSL file lets us make a few things configurable:

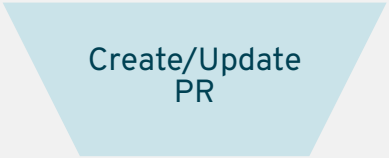
- The names and locations of script files
- How many script files to run in parallel (AKA sub-stages – the other big change)
- Conditional execution of scripts

If we don't configure these, we get V1 style behavior – predefined script names determine what happens when.

```
|-- automation
|   |-- build-artifacts.packages
|   |-- build-artifacts.sh
|   |-- check-patch.packages
|   |-- check-patch.sh
|   `-- README.md
|-- automation.yaml
|-- build.sh
|-- examples
|   |-- cluster_upgrade.yml
|   |-- manageiq.yml
|   |-- ovirt_image_template.yml
|   |-- ovirt_infra_vars.yml
|   |-- ovirt_infra.yml
|   |-- ovirt_vm_infra.yml
|   `-- passwords.yml
|-- LICENSE
|-- meta
|   `-- main.yml
|-- ovirt-ansible-roles.spec.in
```

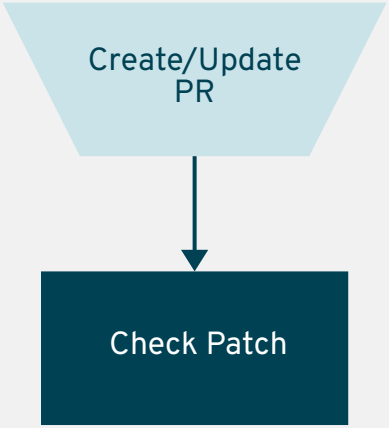
STDCI STAGES

Code life-cycle - when do things run



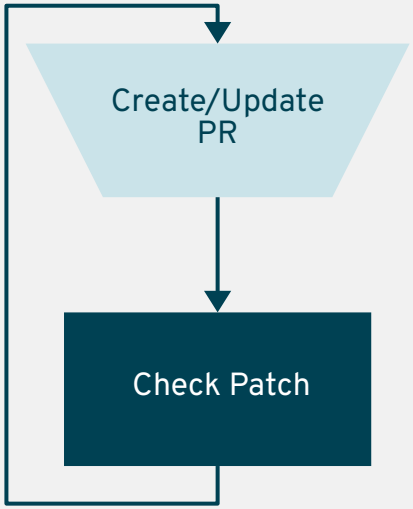
STDCI STAGES

Code life-cycle - when do things run



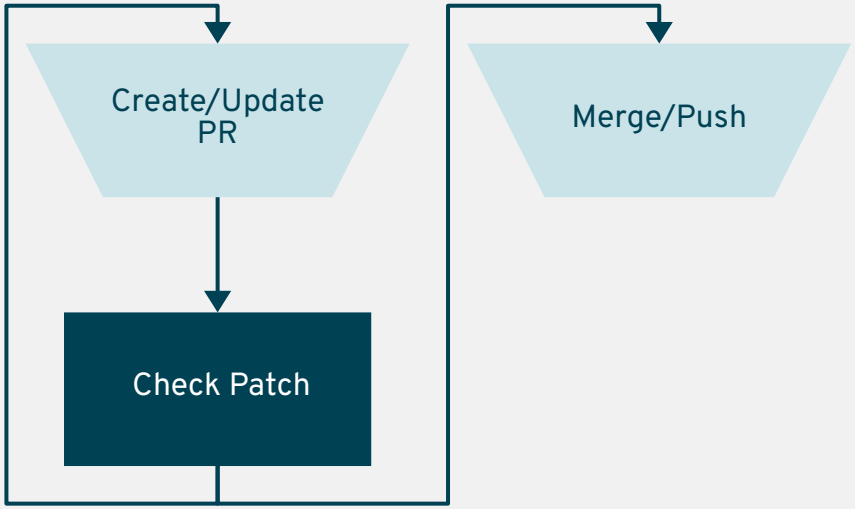
STDCI STAGES

Code life-cycle - when do things run



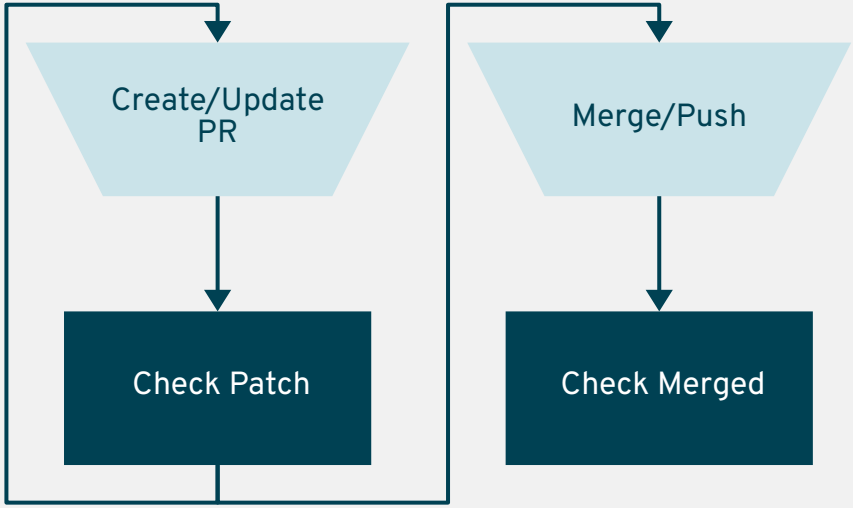
STDCI STAGES

Code life-cycle - when do things run



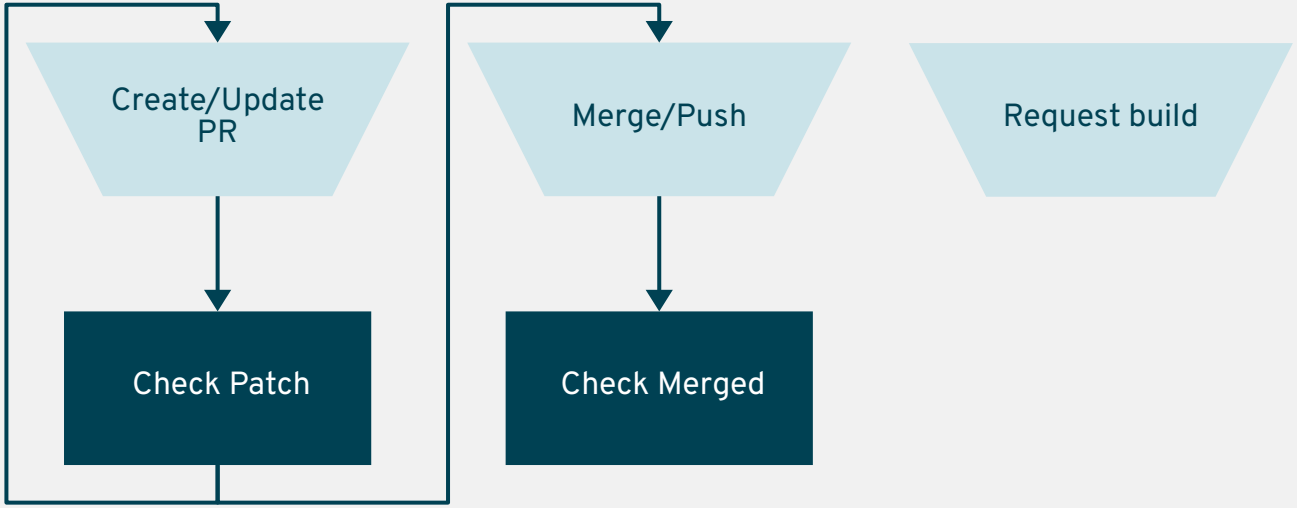
STDCI STAGES

Code life-cycle - when do things run



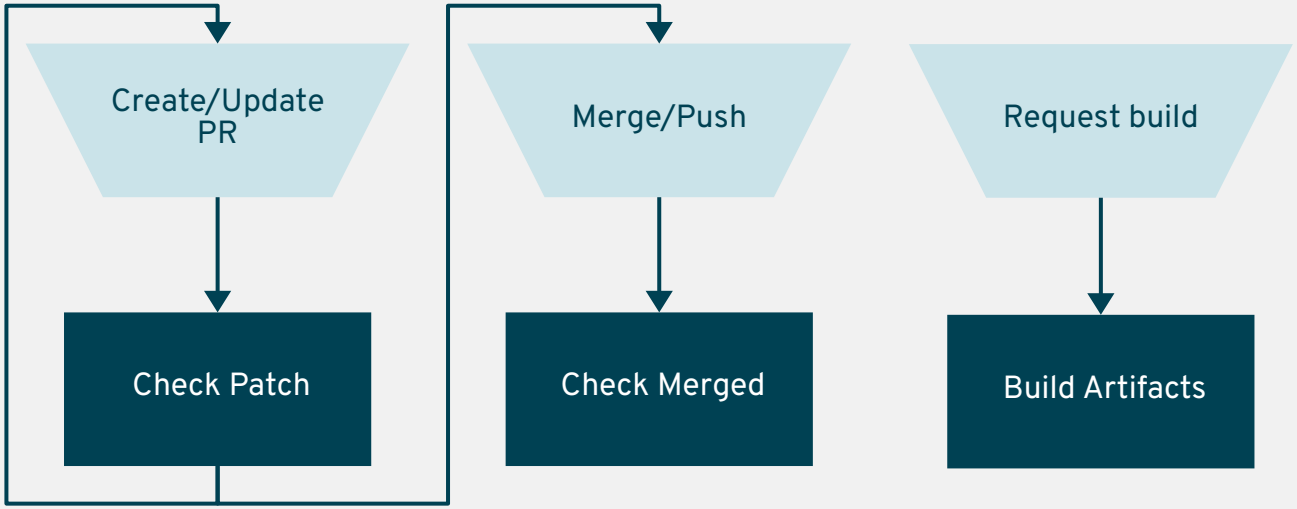
STDCI STAGES

Code life-cycle - when do things run



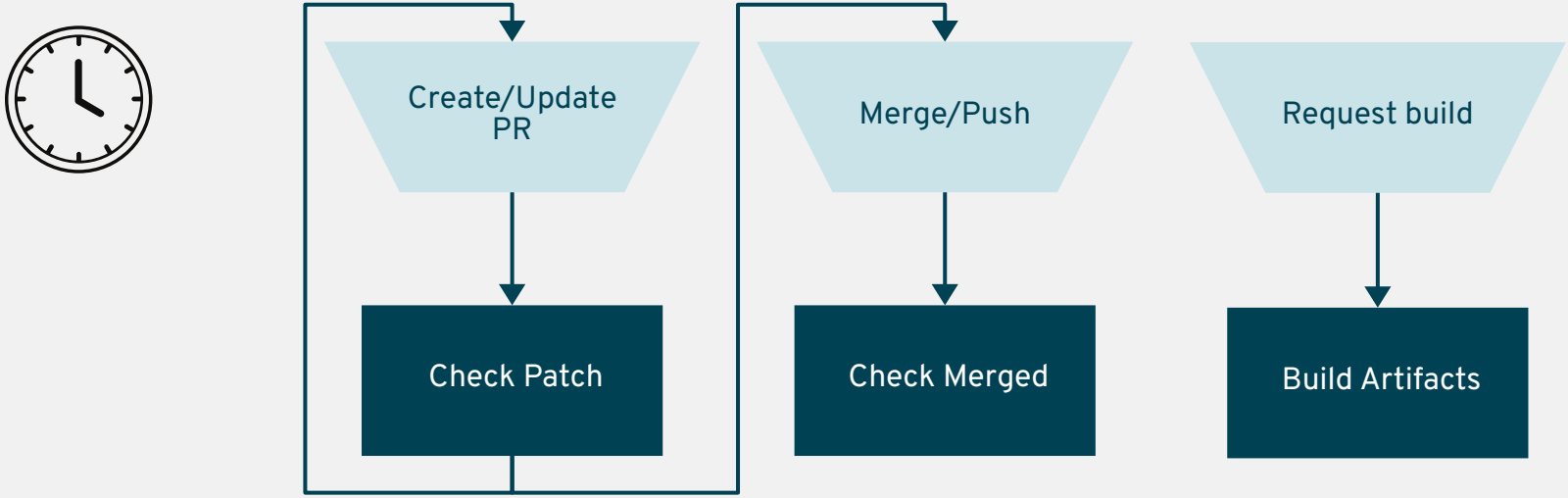
STDCI STAGES

Code life-cycle - when do things run



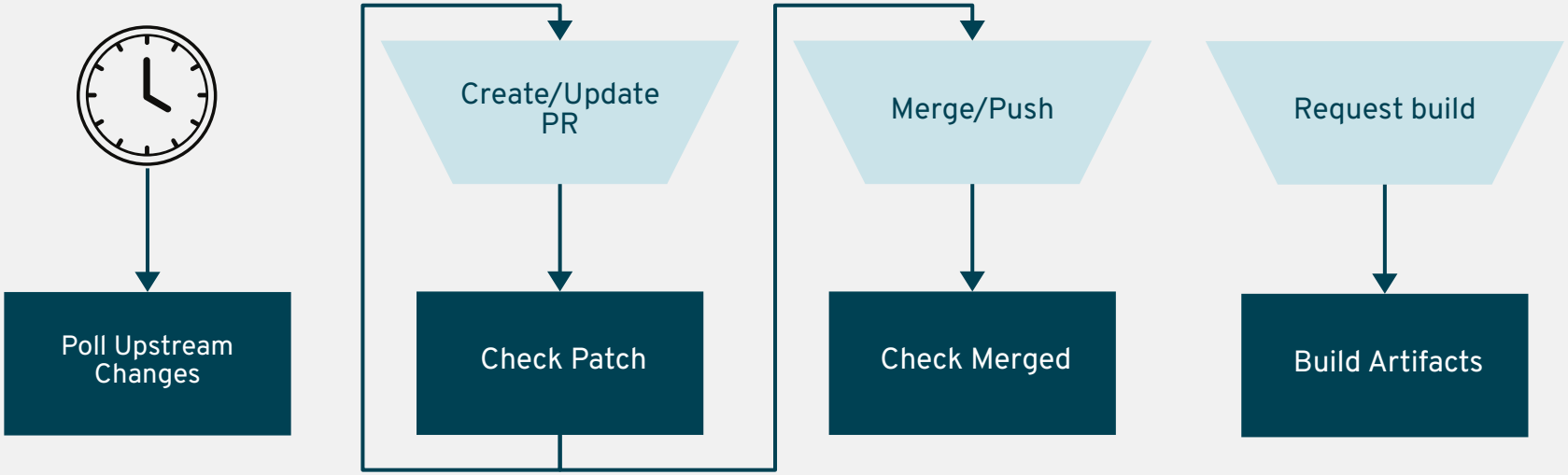
STDCI STAGES

Code life-cycle - when do things run



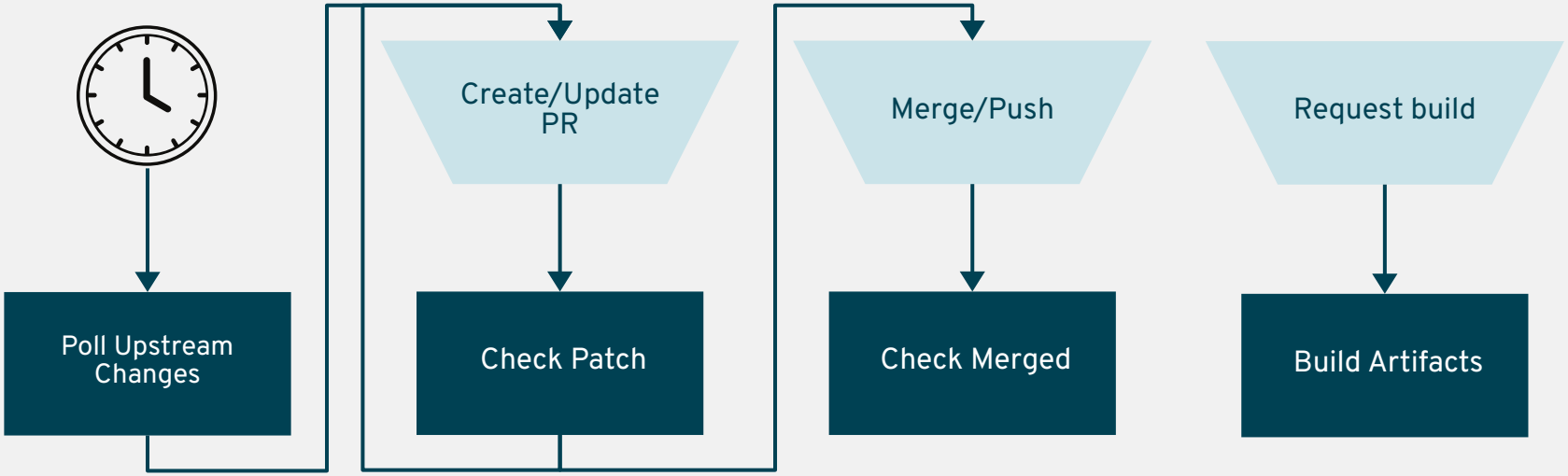
STDCI STAGES

Code life-cycle - when do things run



STDCI STAGES

Code life-cycle - when do things run

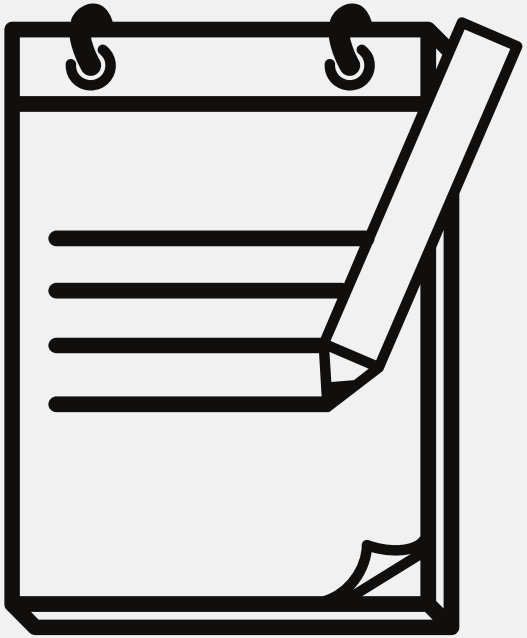


STDCI DSL

Making it easy to use

File format is YAML-based and:

- Key names are case-insensitive
- ...And space, hyphen and underscore insensitive
- ...And plural/singular form insensitive
- ...And sane defaults are there for everything
- ...And some inline Jinja2 is supported where it makes sense.



STDCI DSL

Setting architectures and distributions

```
Architectures:
  - x86_64
  - ppc64le
Distributions:
  - el7
  - fc27
```

STDCI DSL

Setting different distributions for different architectures

```
Architectures:
  - x86_64:
      Distributions:
        - el7
        - fc27
  - ppc64le:
      Distribution: el7
```

STDCI DSL

Setting different distributions for different architectures (JSON shorthand)

```
Architectures:
  - x86_64:
      Distributions: [ el7, fc27 ]
  - ppc64le:
      Distribution: el7
```

STDCI DSL

Setting different distributions for different architectures (From general to specific)

```
Architectures:
  - x86_64:
  - ppc64le:
      Distribution: el7
  - s390x
Distributions: [ el7, fc27 ]
```

STDCI DSL

Setting which scripts run

```
Script:
  From file: test.sh
```

STDCI DSL

Only run on specific stage

```
Script:
  From file: test.sh
Stage: check-patch
```

STDCI DSL

Different scripts for different stages

```
Stage:
  - check-patch:
      Script:
        From file: test.sh
  - build-artifacts:
      Script:
        From file: build.sh
```

STDCI DSL

Running on multiple platforms

```
Stage:
  - check-patch:
      Script:
        From file: test.sh
  - build-artifacts:
      Script:
        From file: build.sh
Distributions:
  - el7
  - fc27
```

STDCI DSL

Different platforms for different stages

```
Stage:
  - check-patch:
      Script:
        From file: test.sh
  - build-artifacts:
      Script:
        From file: build.sh
      Distros: [“el7”, “fc27”]
Distro: el7
```

STDCI DSL

Different scripts for different platforms

```
Stage:
  - build-artifacts:
      Distros:
        - el7:
            Script:
              From file: build.el7.sh
        - fc27:
            Script:
              From file: build.fc27.sh
```

STDCI DSL

Different scripts for different platforms with Jinja templates

```
Stage:
  - build-artifacts:
      Distros: [“el7”, “fc27”]
      Script:
        From file: “build.{{ distro }}.sh"
```

STDCI DSL

Multiple parallel scripts for the same stage

```
stage: check-patch
sub-stages:
  - unit-test:
      script:
        from-file: "test/unit-test.sh"
  - linter-test
      script:
        from-file: "test/linter-test.sh"
```

STDCI DSL

Multiple parallel scripts for the same stage with Jinja

```
stage: check-patch
sub-stages:
  - unit-test
  - linter-test
script:
  from-file: "test/{{ substage }}.sh
```

STDCI DSL

Multiple parallel scripts for the same stage with default script names

```
stage: check-patch
sub-stages:
  - unit-test
  - linter-test
```

STDCI DSL

Require a bare-metal host

```
substages:
  - build-docs
  - system-test:
      runtime-requirements:
        support-nesting-level: 2
```

STDCI DSL

Send builds to the change queue.

```
Release branches:
  - master: [ ‘ovirt-master’, ‘ovirt-4.2' ]
  - 1.0: ovirt-4.1
```

STDCI SCRIPT REQUIREMENT FILES

What happened to all those ‘*.packages’ and ‘*.repos’ files

Files are still needed, names should match the name of the script rather then the name of the STDCI stage.

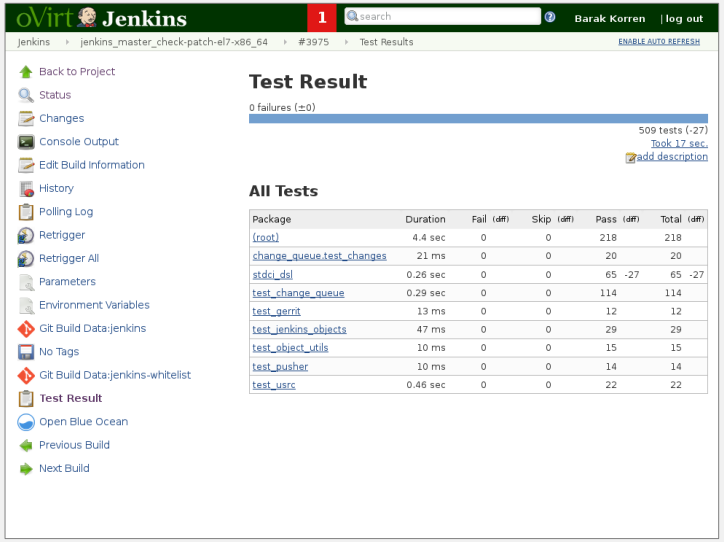
In the future we will allow setting these directly from YAML, but not yet.

STDCI V2 RESULT COLLECTION

Basically the same as it was in V1

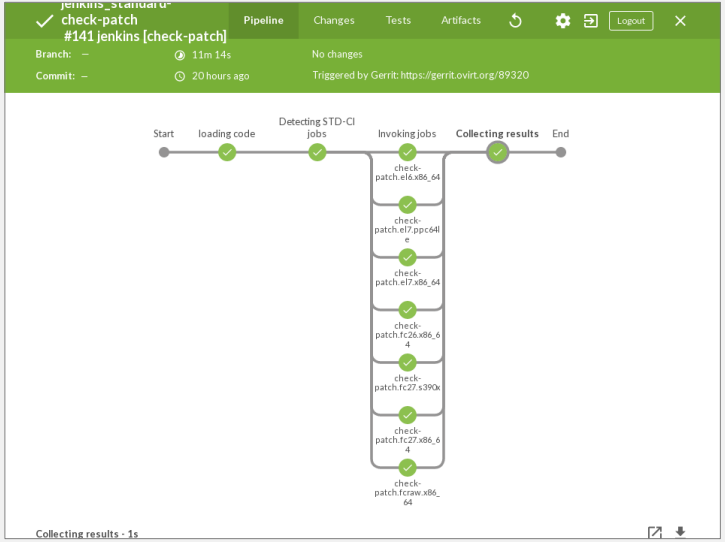
Build/Test script can drop files in 'exported-artifacts/' so that:

- RPMs become yum repositories
- Containers * get pushed to a registry
- JUNIT XML files get displayed in Jenkins results screen
- The job's artifacts are splic into directories according to the stage/sub stage/distro/arch combination that made them



STDCI IMPLEMENTATION

How we've built it

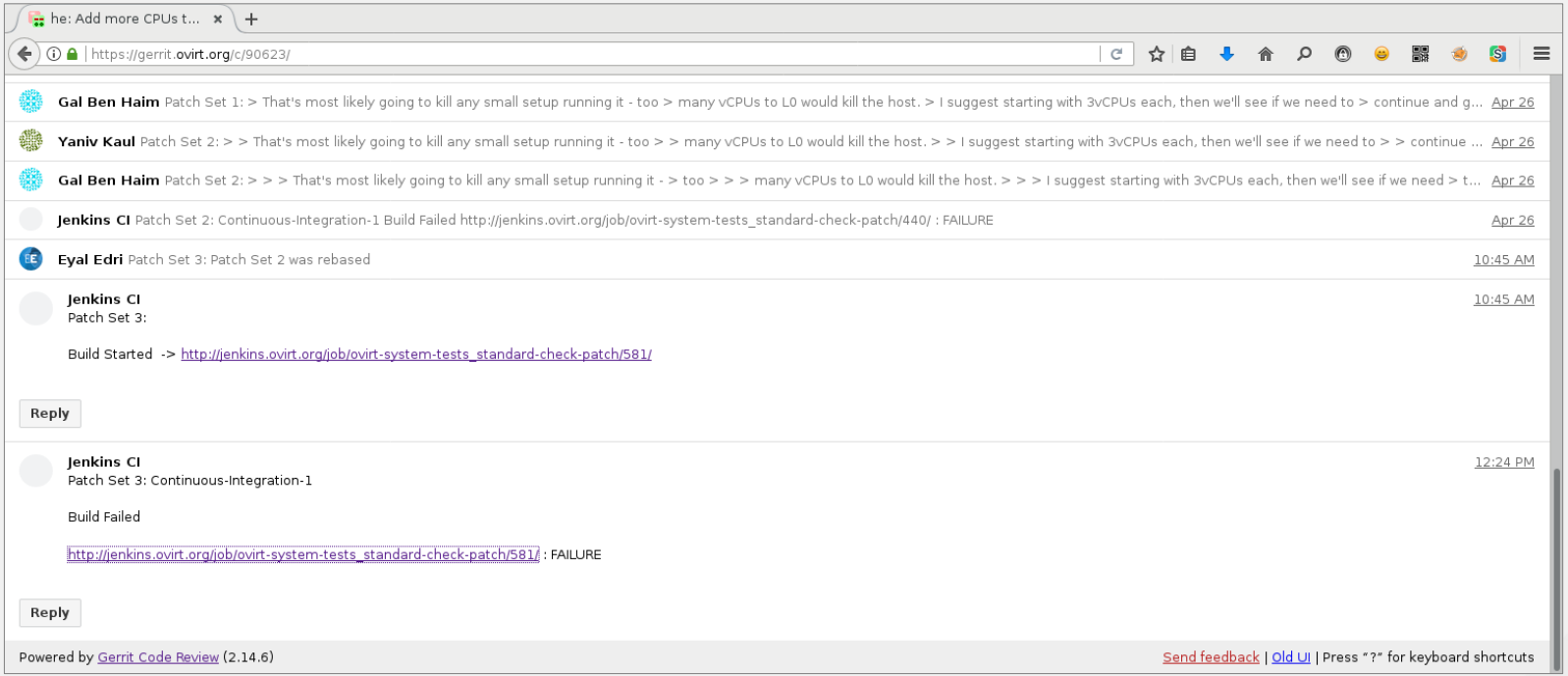


The STDCI V2 implementation is using:

- Jenkins pipeline jobs – each project gets the same two jobs instead of different configuration per project
- Python for DSL parsing
- mock_runner.sh still used as the core runtime tool

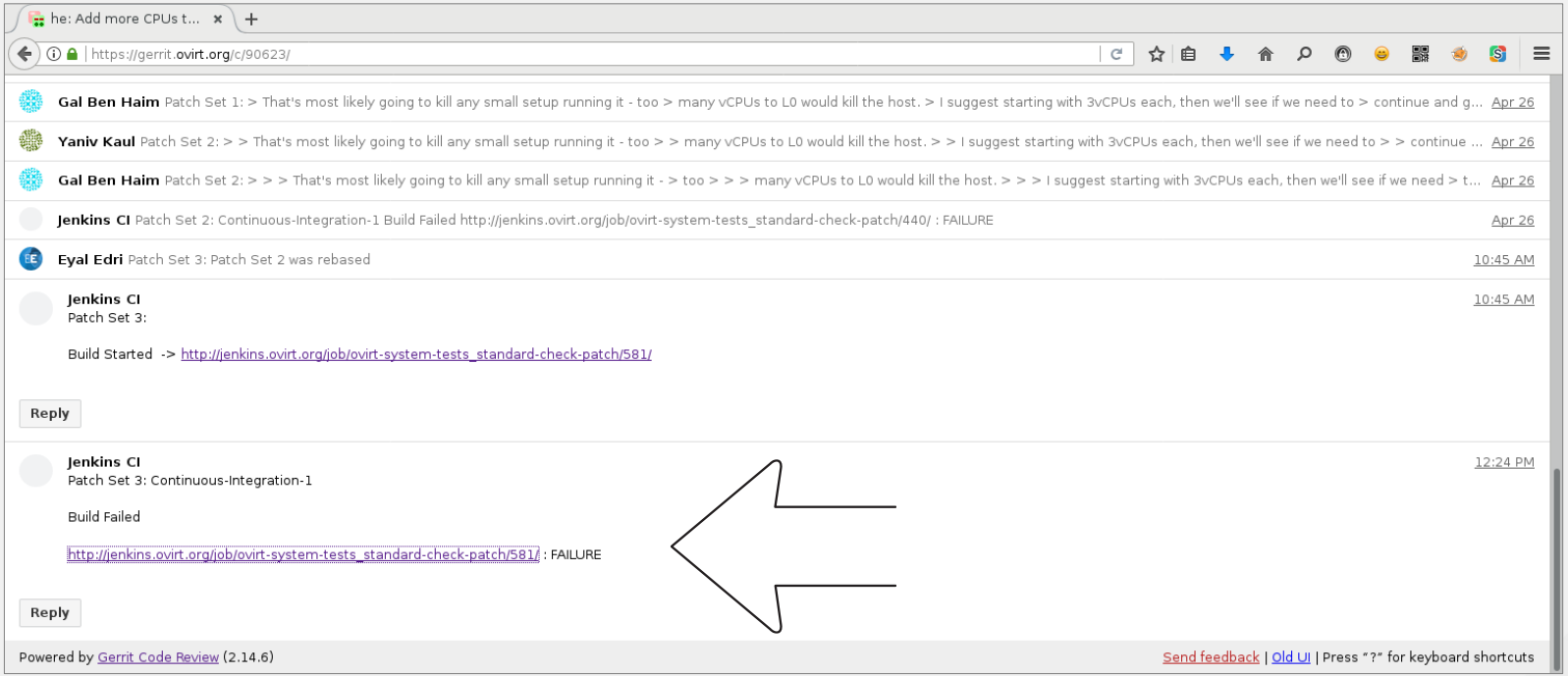
STDCI – DEBUGGING ISSUES

How to find out why is it broken



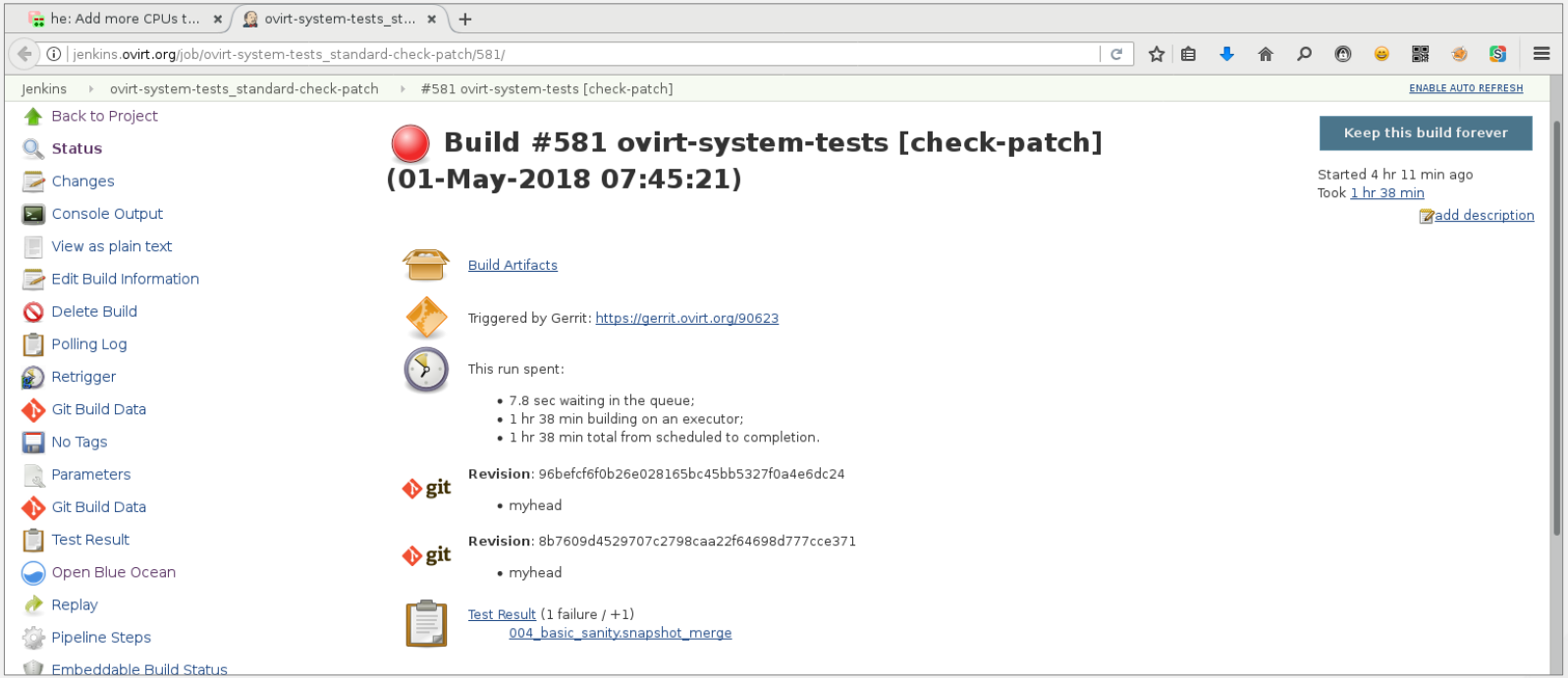
STDCI – DEBUGGING ISSUES

How to find out why is it broken



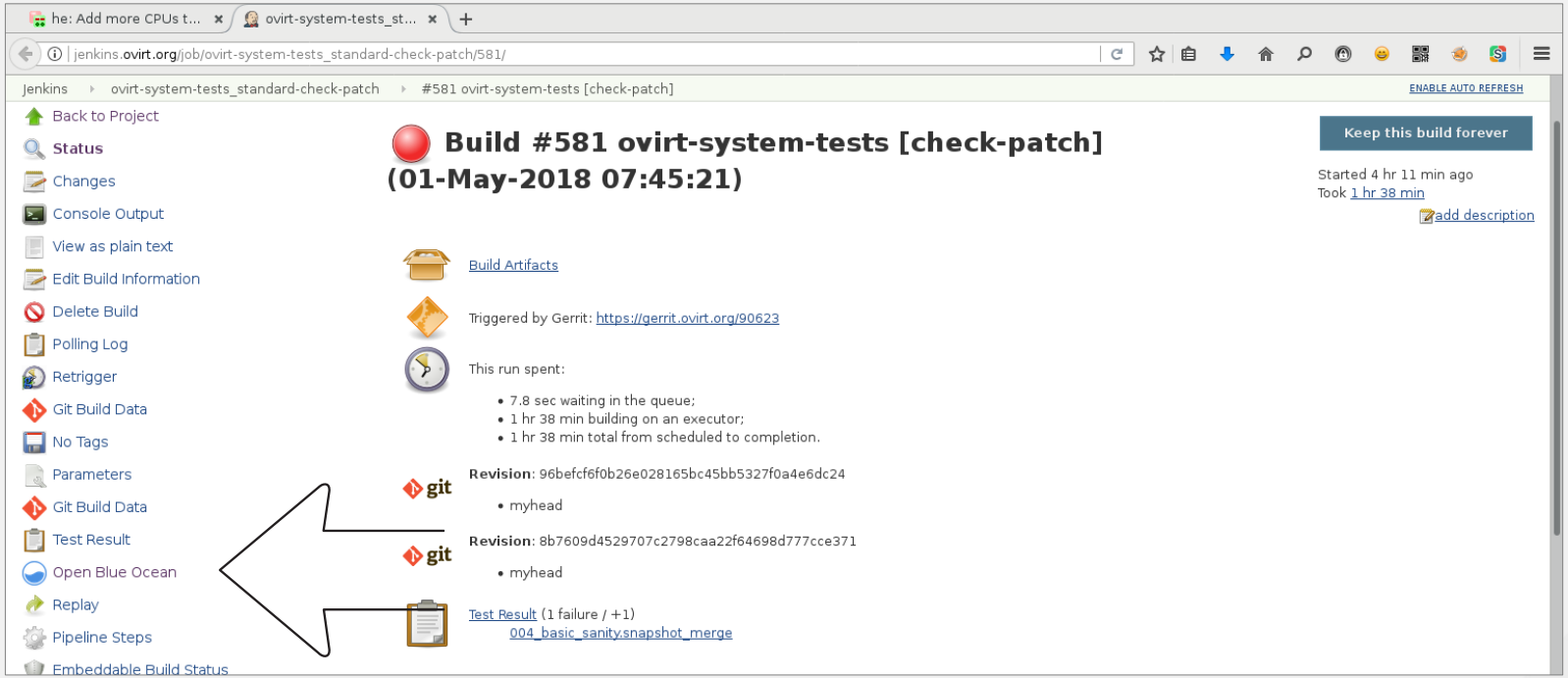
STDCI – DEBUGGING ISSUES

How to find out why is it broken



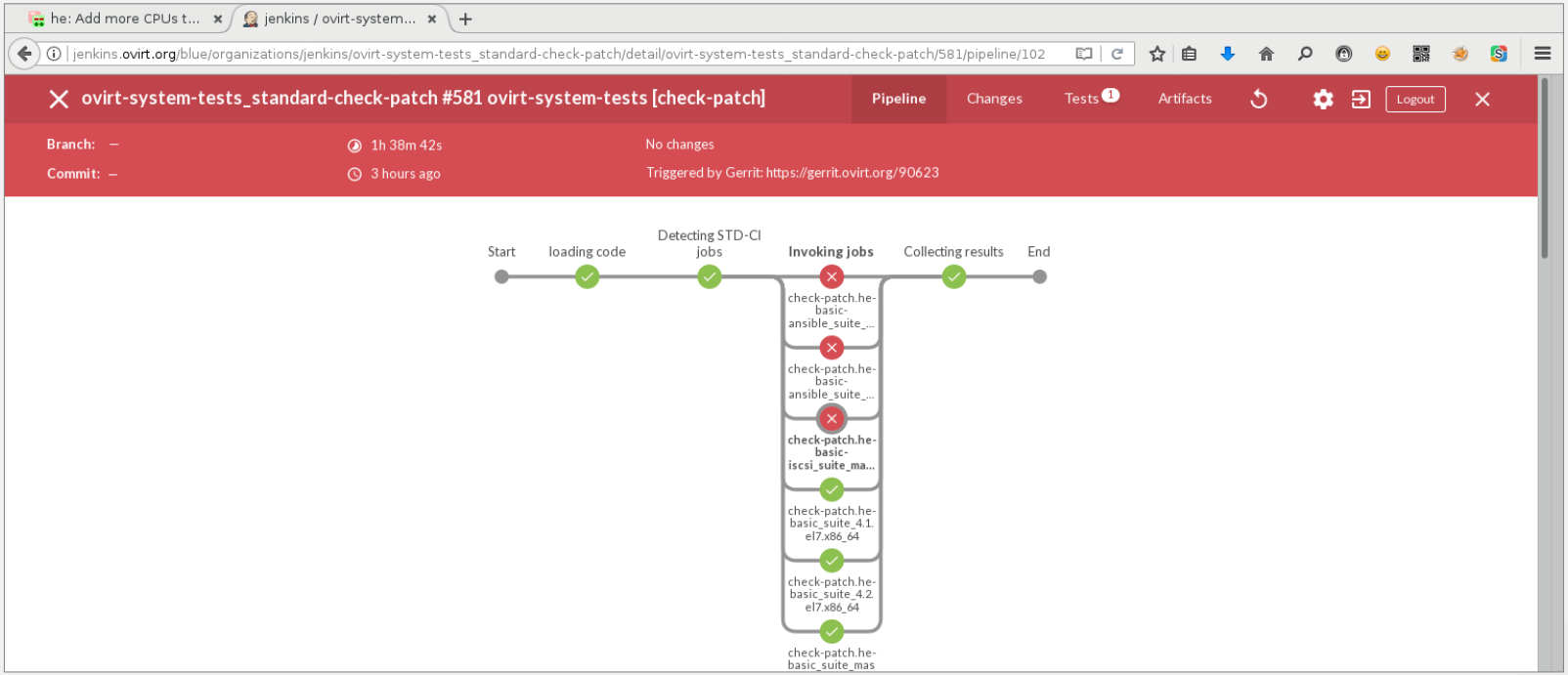
STDCI – DEBUGGING ISSUES

How to find out why is it broken



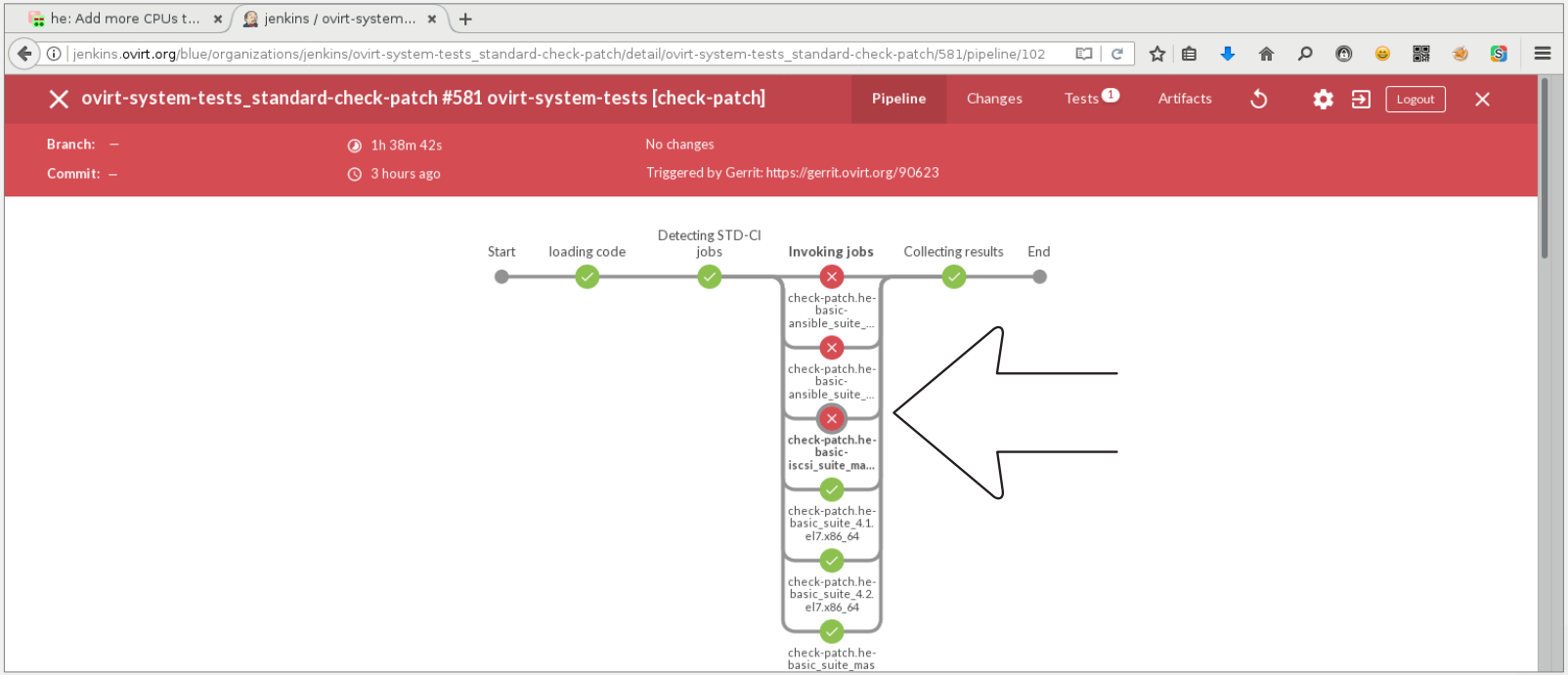
STDCI – DEBUGGING ISSUES

How to find out why is it broken



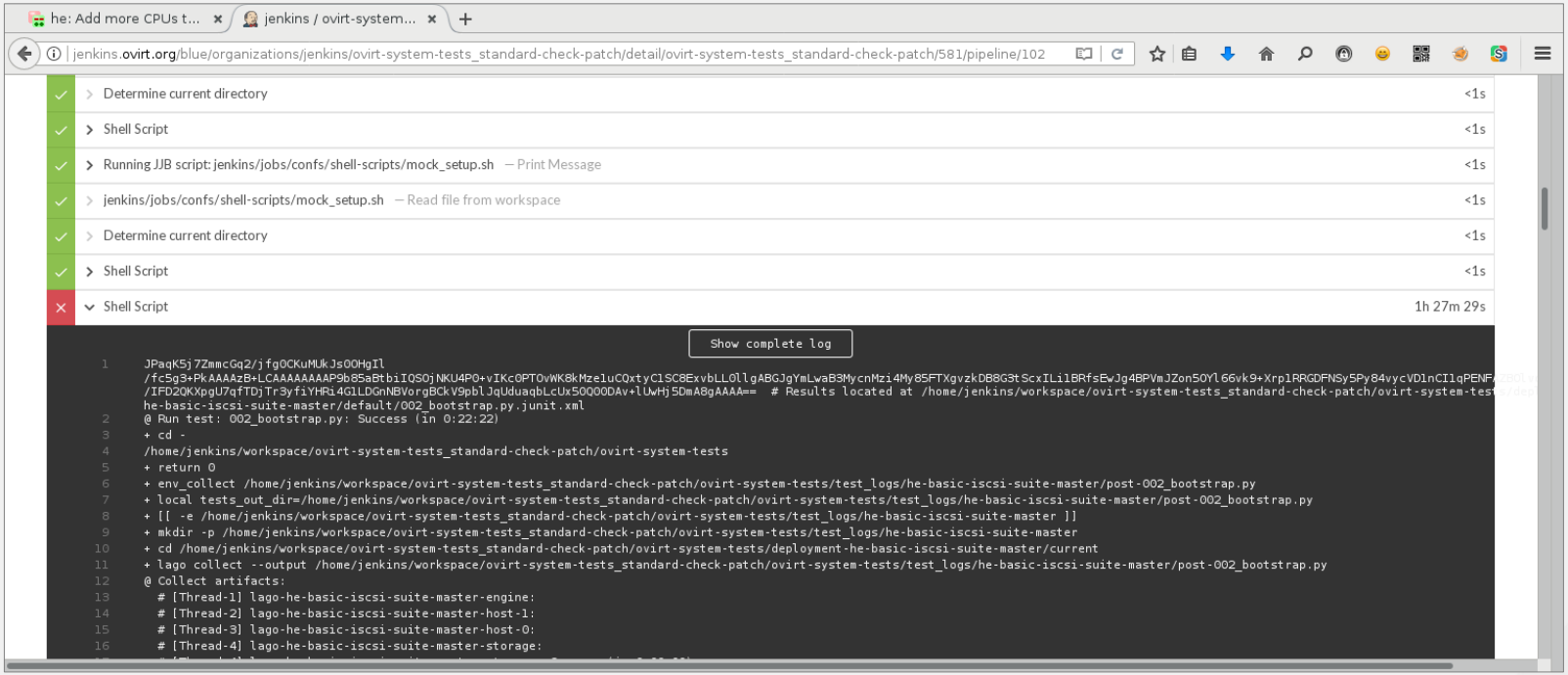
STDCI – DEBUGGING ISSUES

How to find out why is it broken



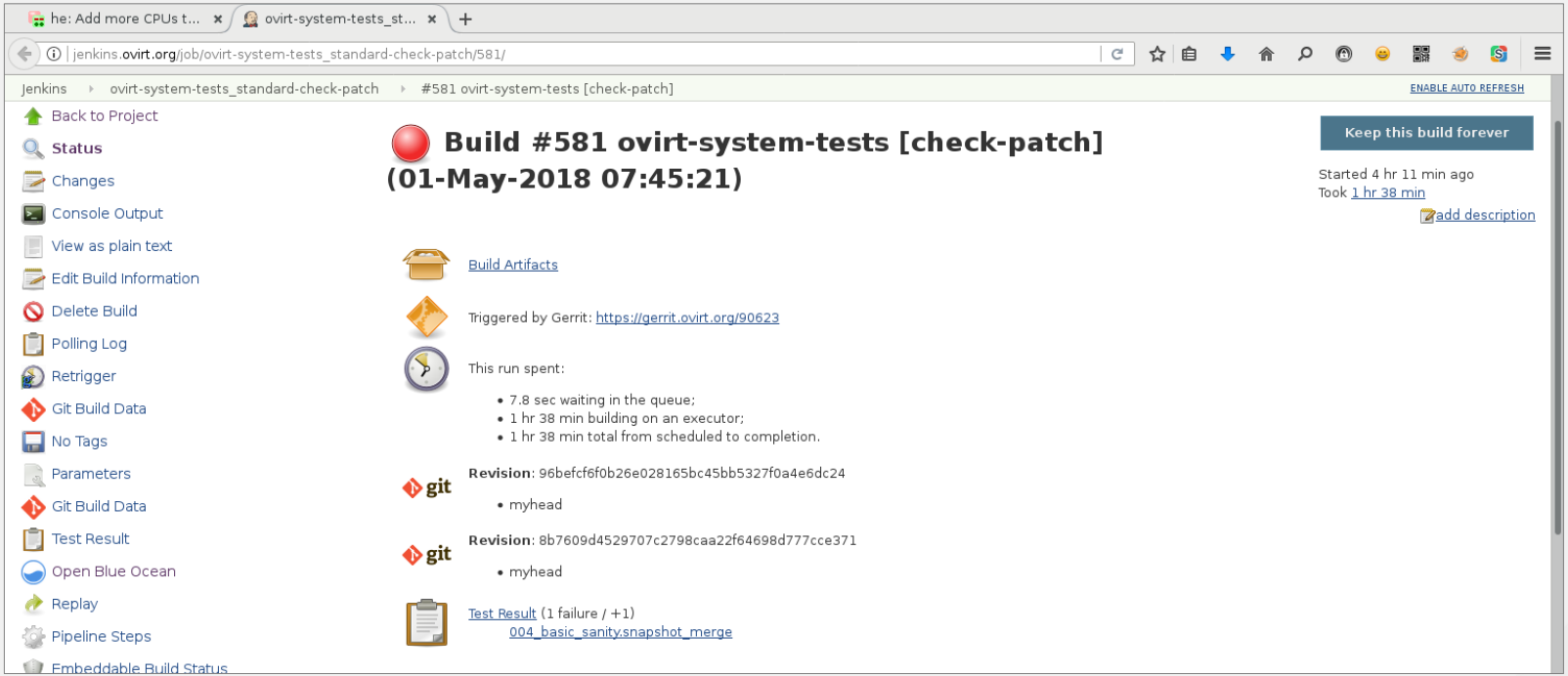
STDCI – DEBUGGING ISSUES

How to find out why is it broken



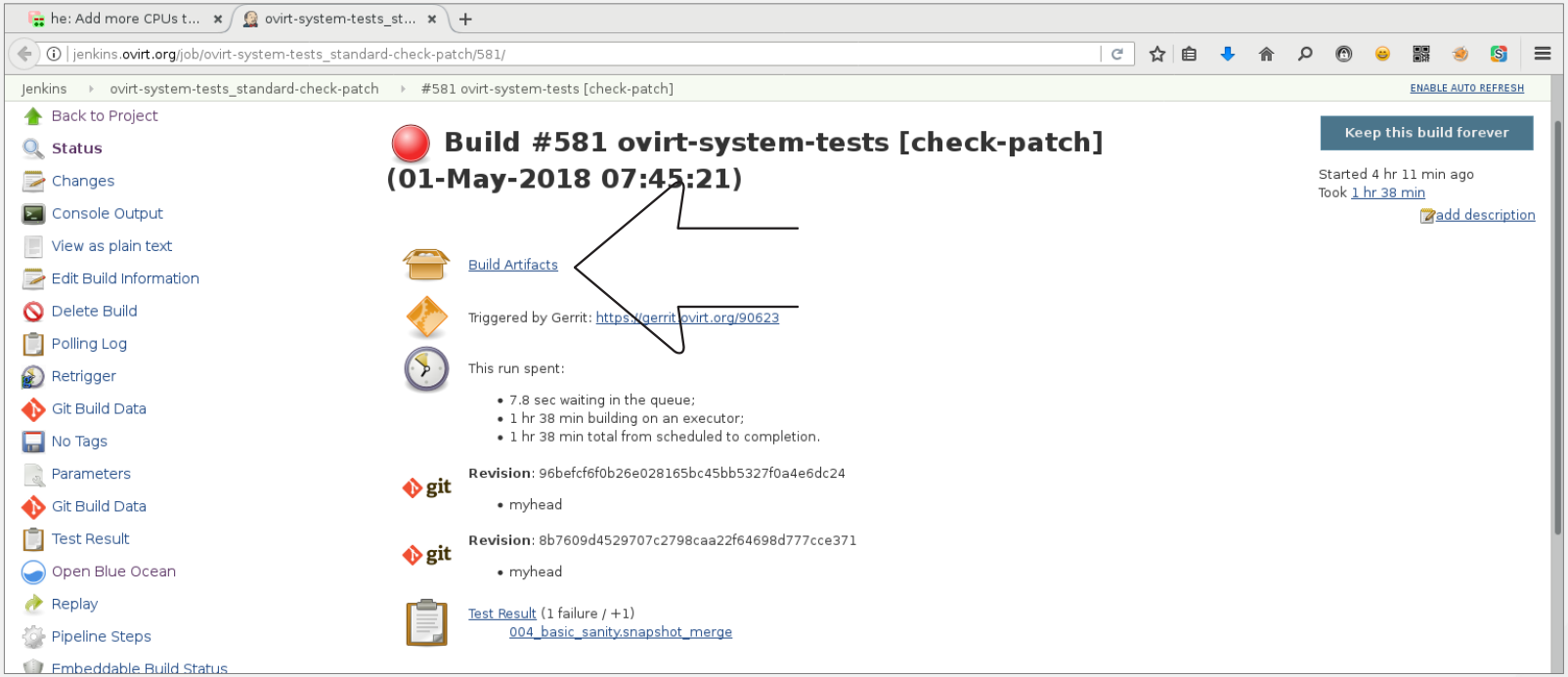
STDCI – DEBUGGING ISSUES

How to find out why is it broken



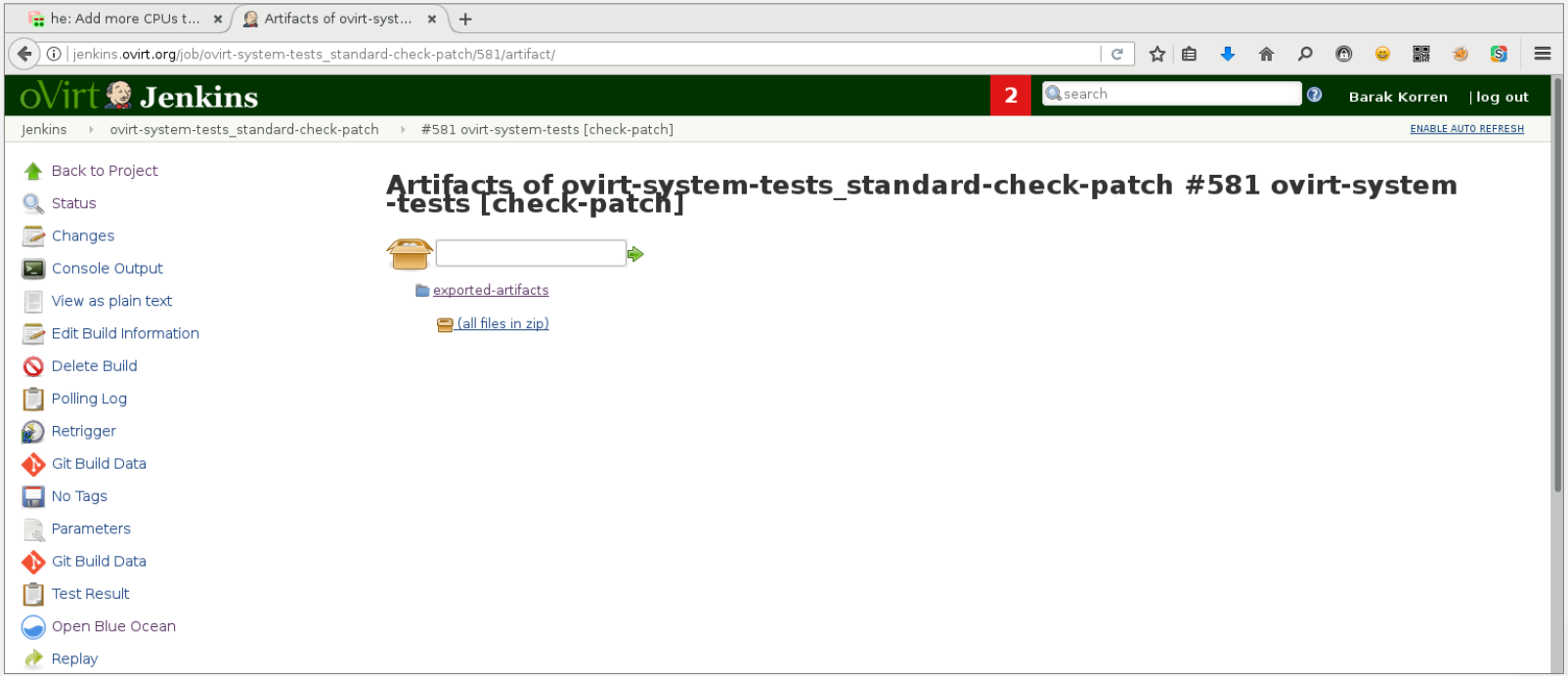
STDCI – DEBUGGING ISSUES

How to find out why is it broken



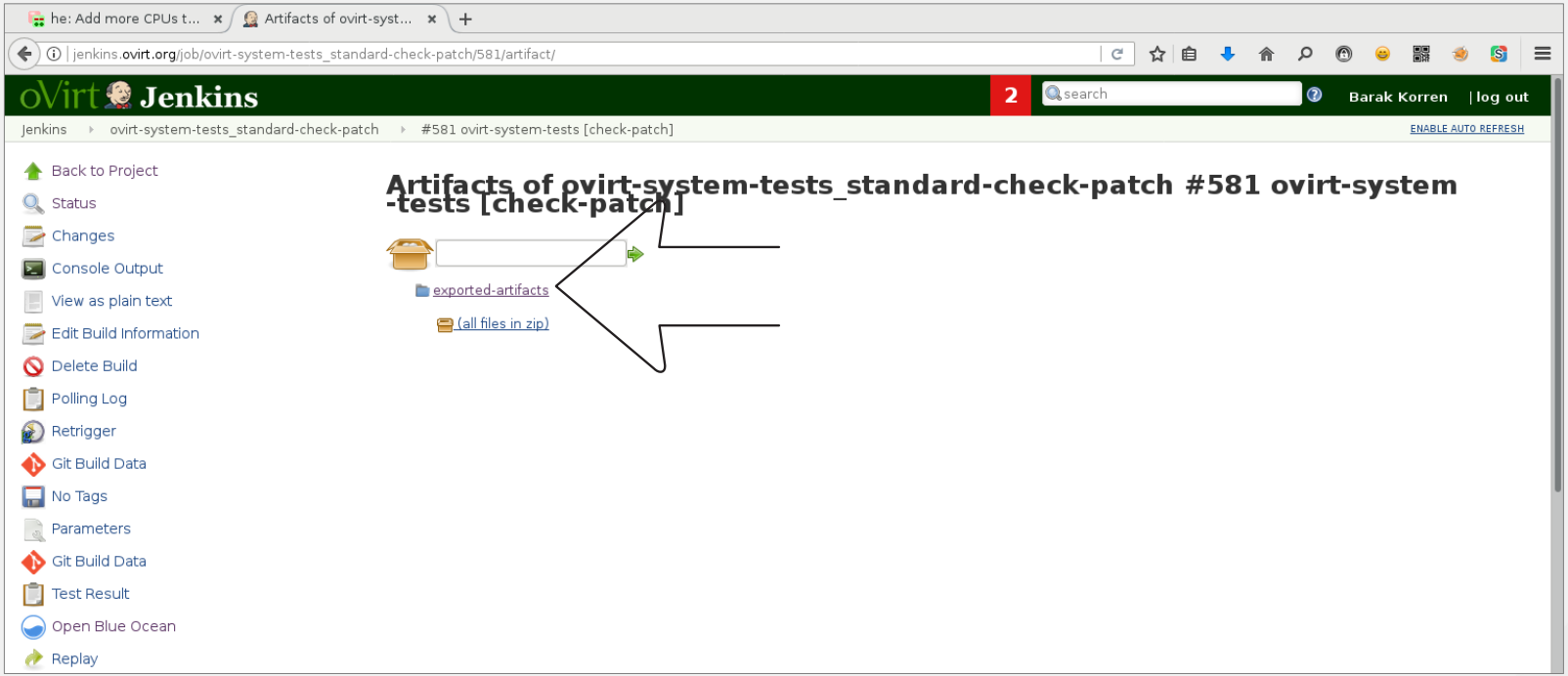
STDCI – DEBUGGING ISSUES

How to find out why is it broken



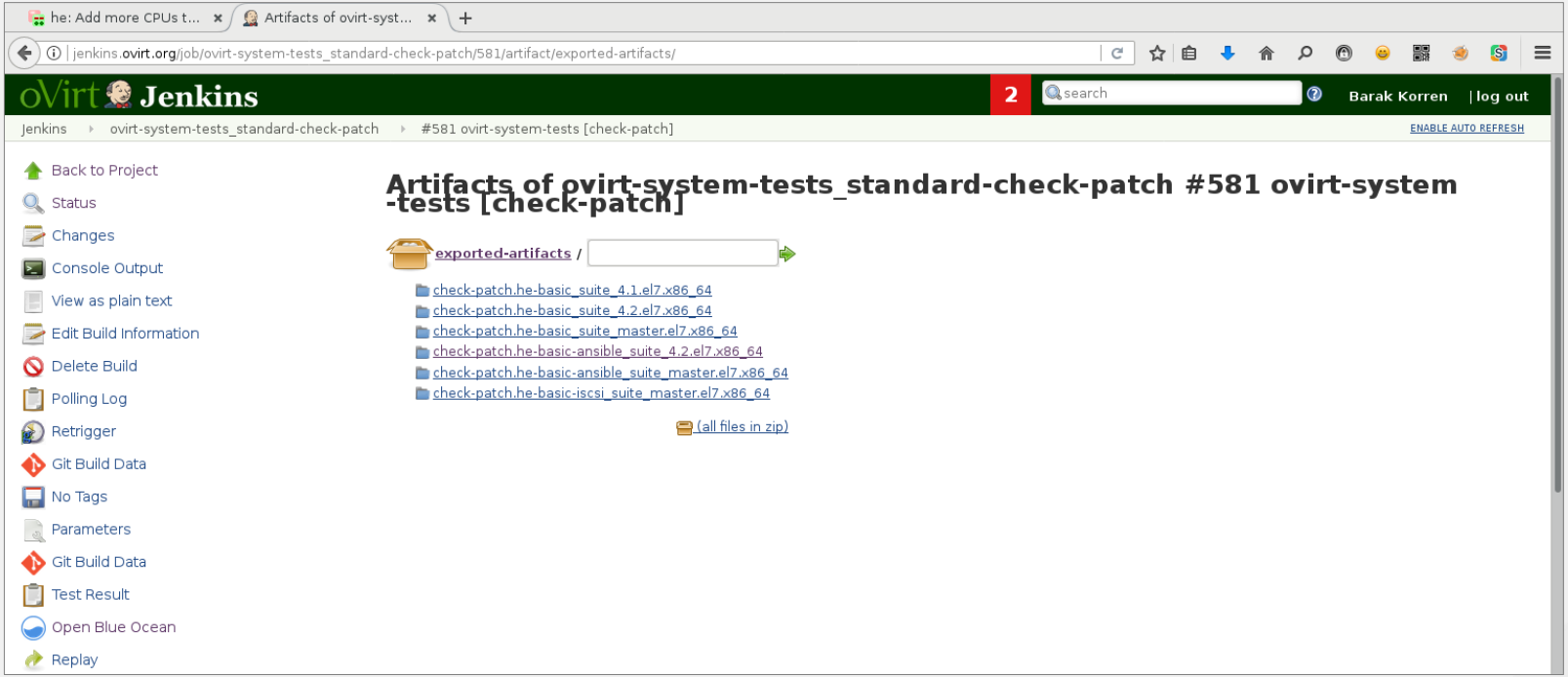
STDCI – DEBUGGING ISSUES

How to find out why is it broken



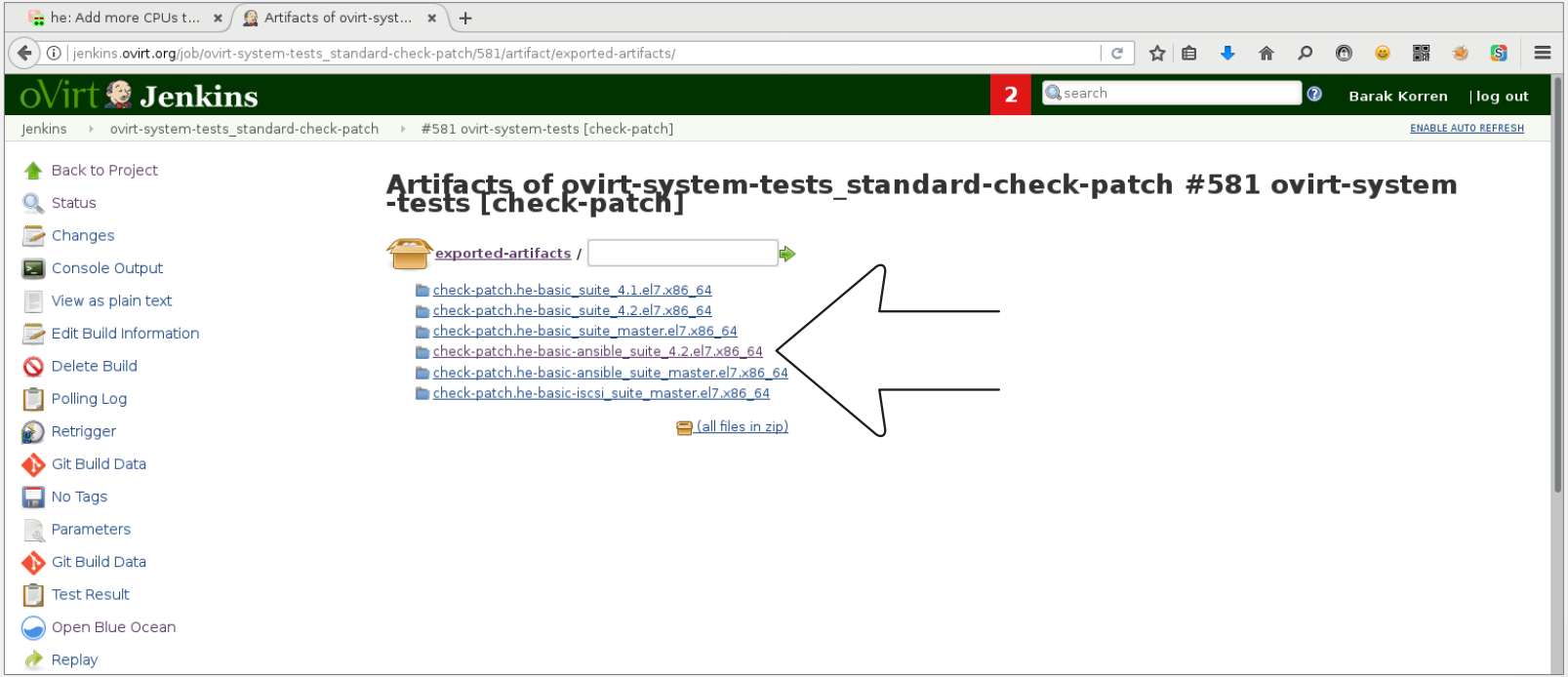
STDCI – DEBUGGING ISSUES

How to find out why is it broken



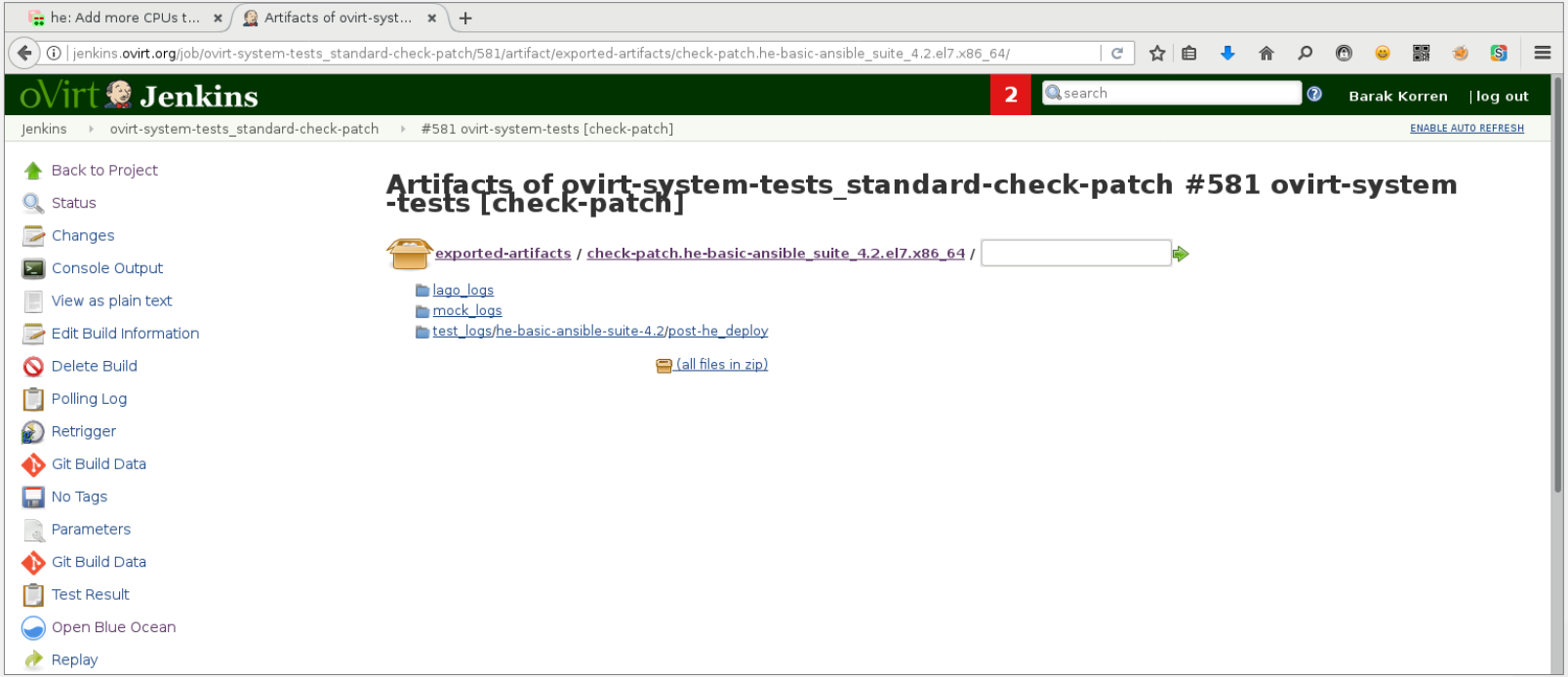
STDCI – DEBUGGING ISSUES

How to find out why is it broken



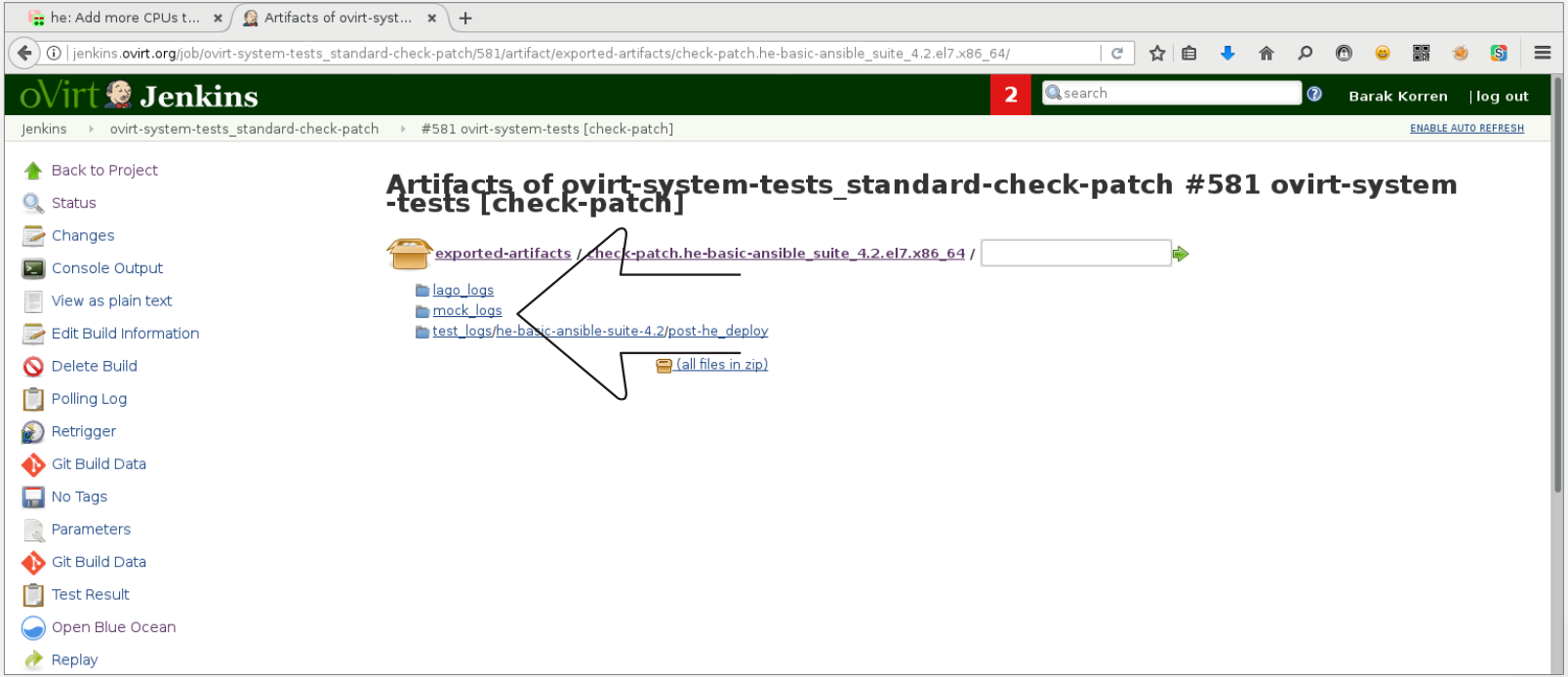
STDCI – DEBUGGING ISSUES

How to find out why is it broken



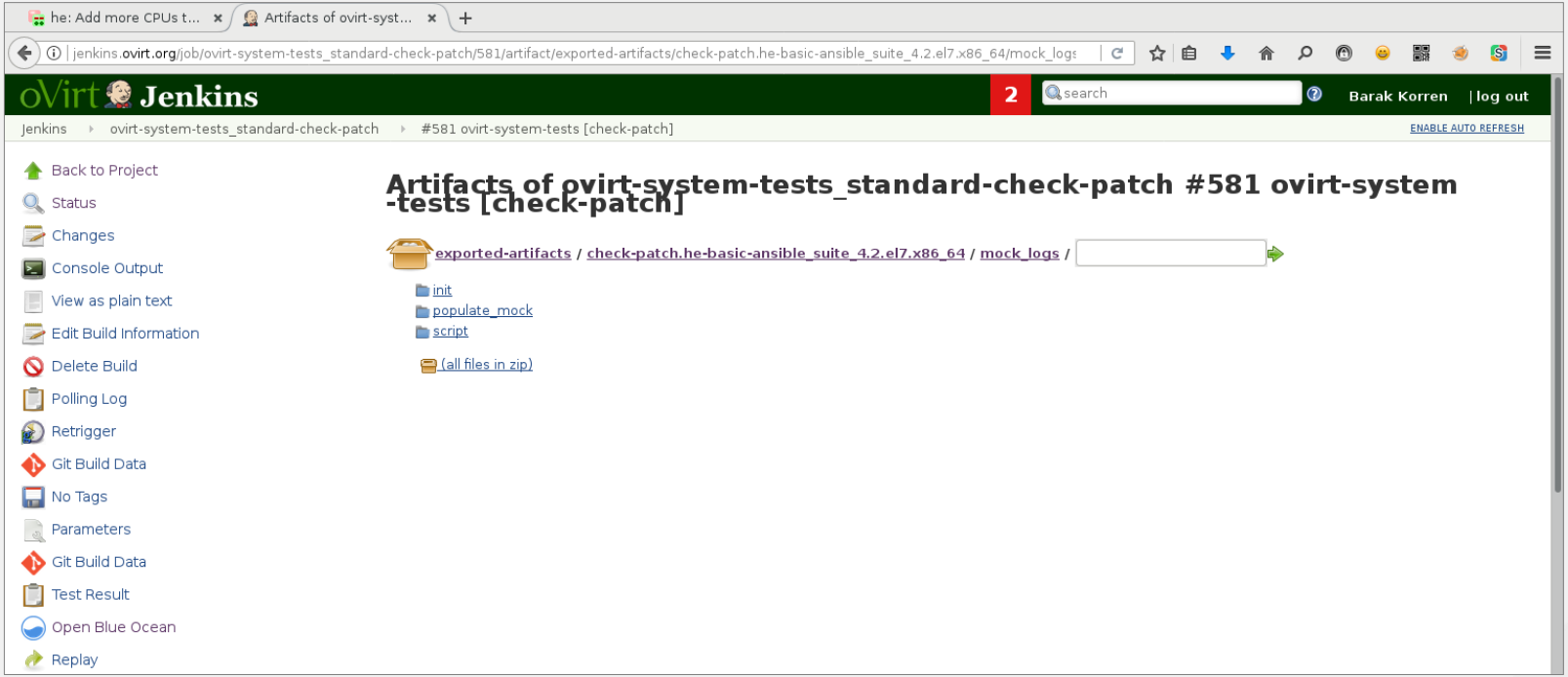
STDCI – DEBUGGING ISSUES

How to find out why is it broken



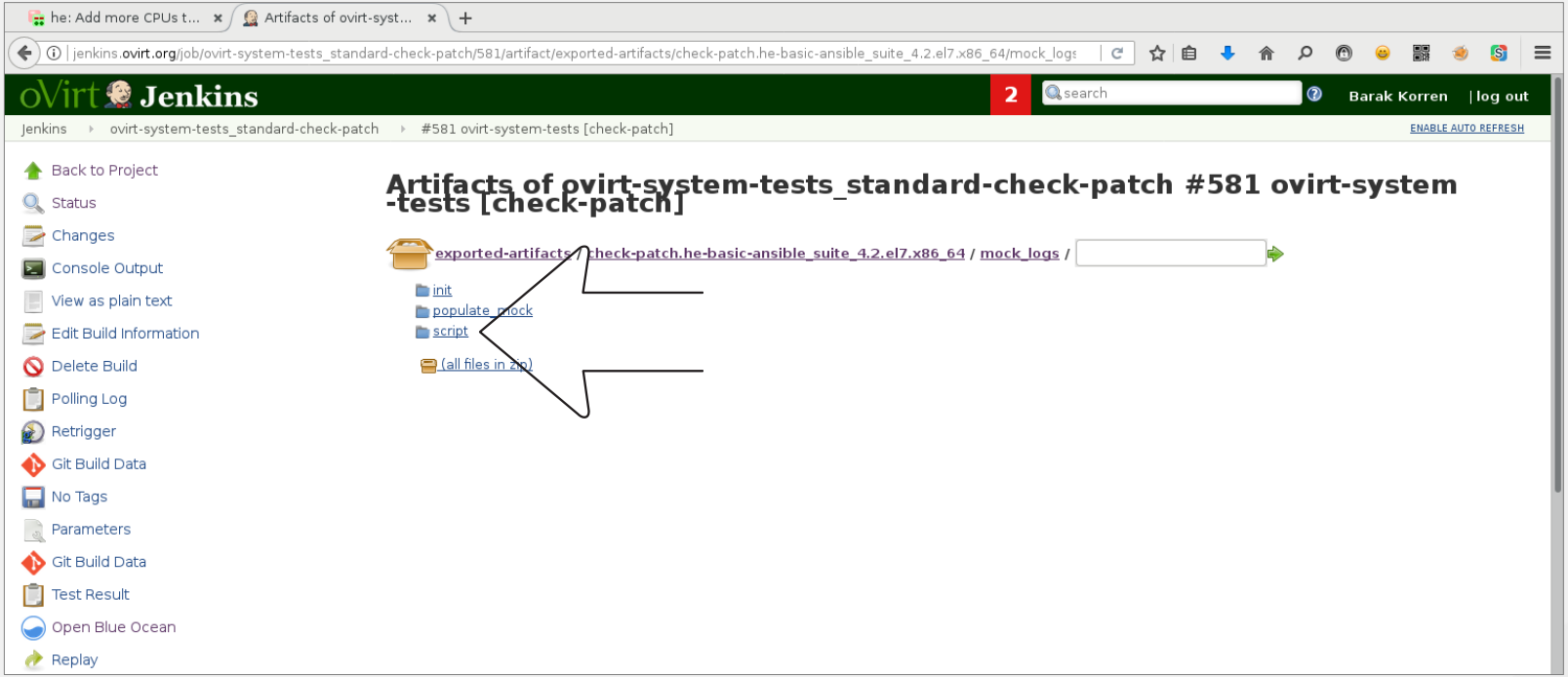
STDCI – DEBUGGING ISSUES

How to find out why is it broken



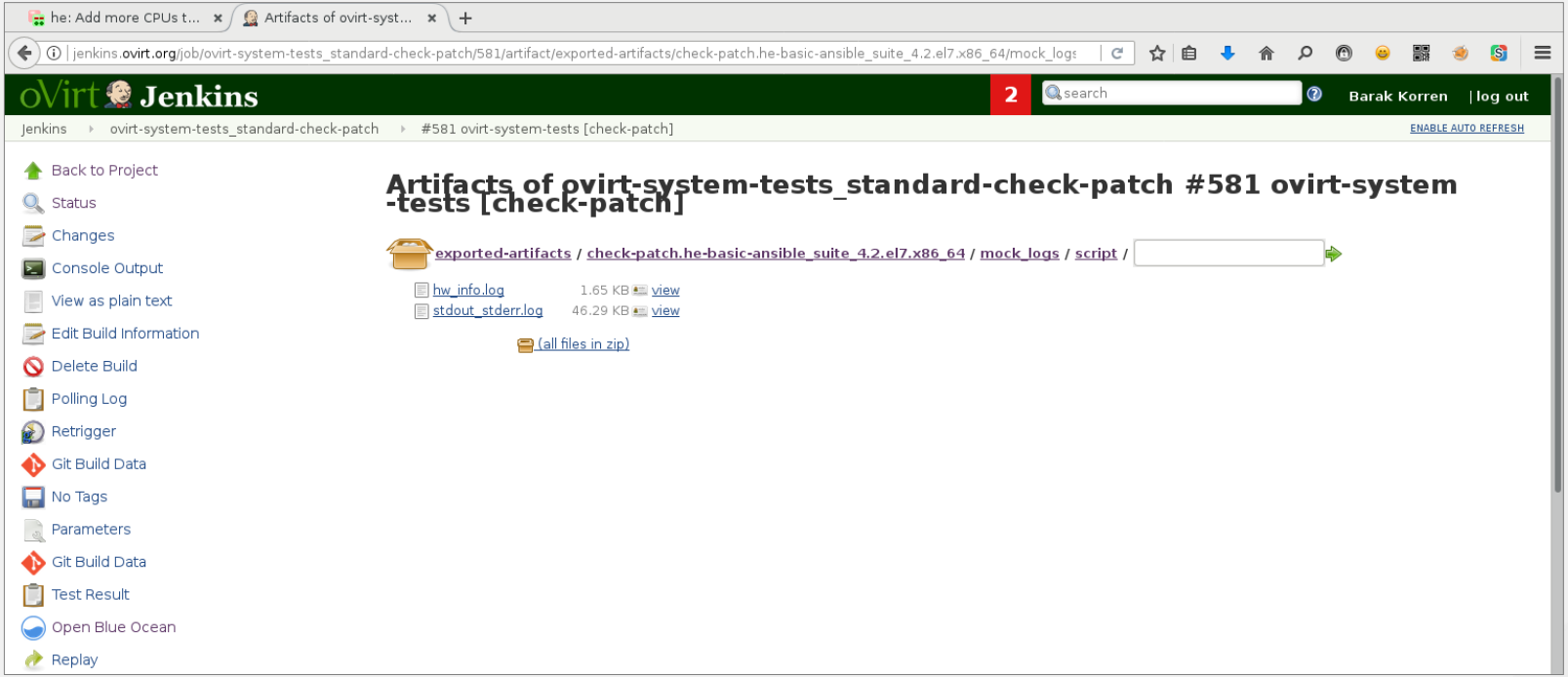
STDCI – DEBUGGING ISSUES

How to find out why is it broken



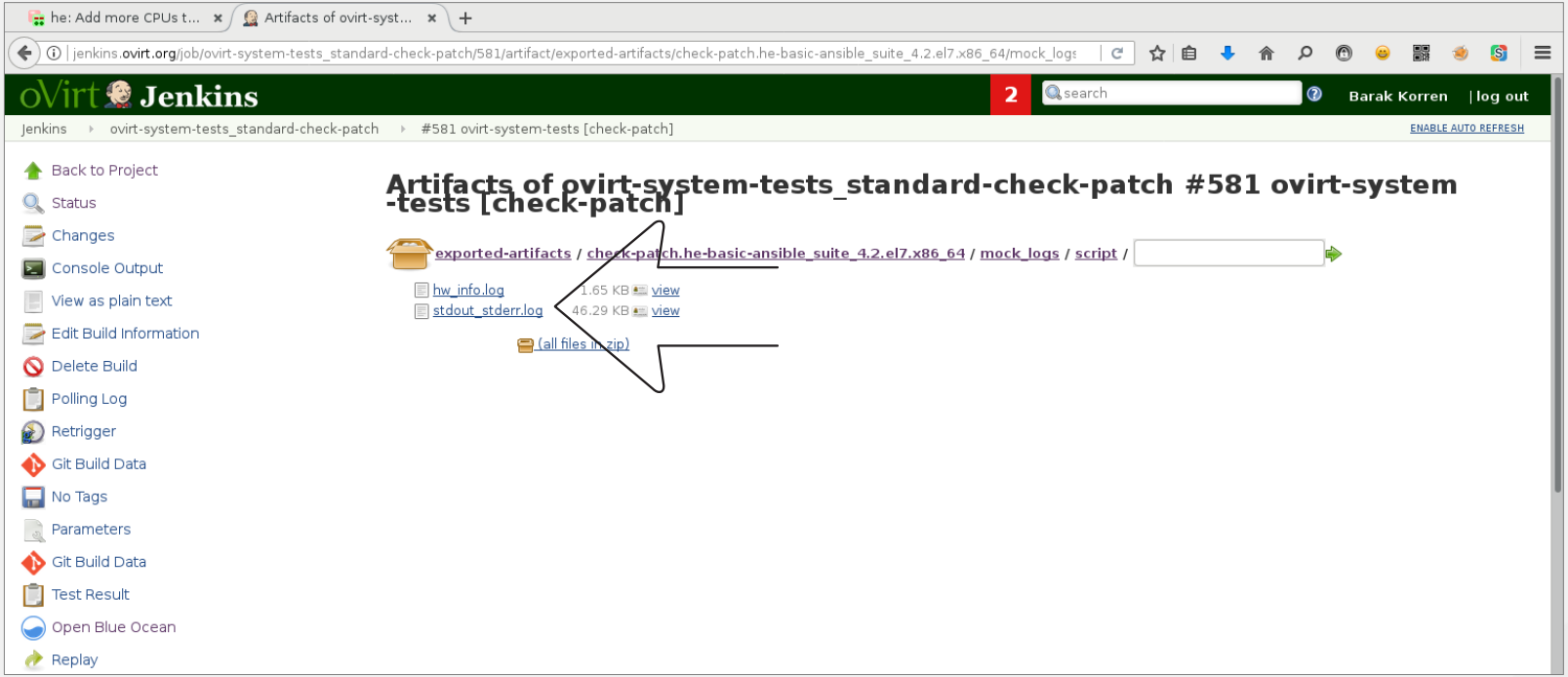
STDCI – DEBUGGING ISSUES

How to find out why is it broken



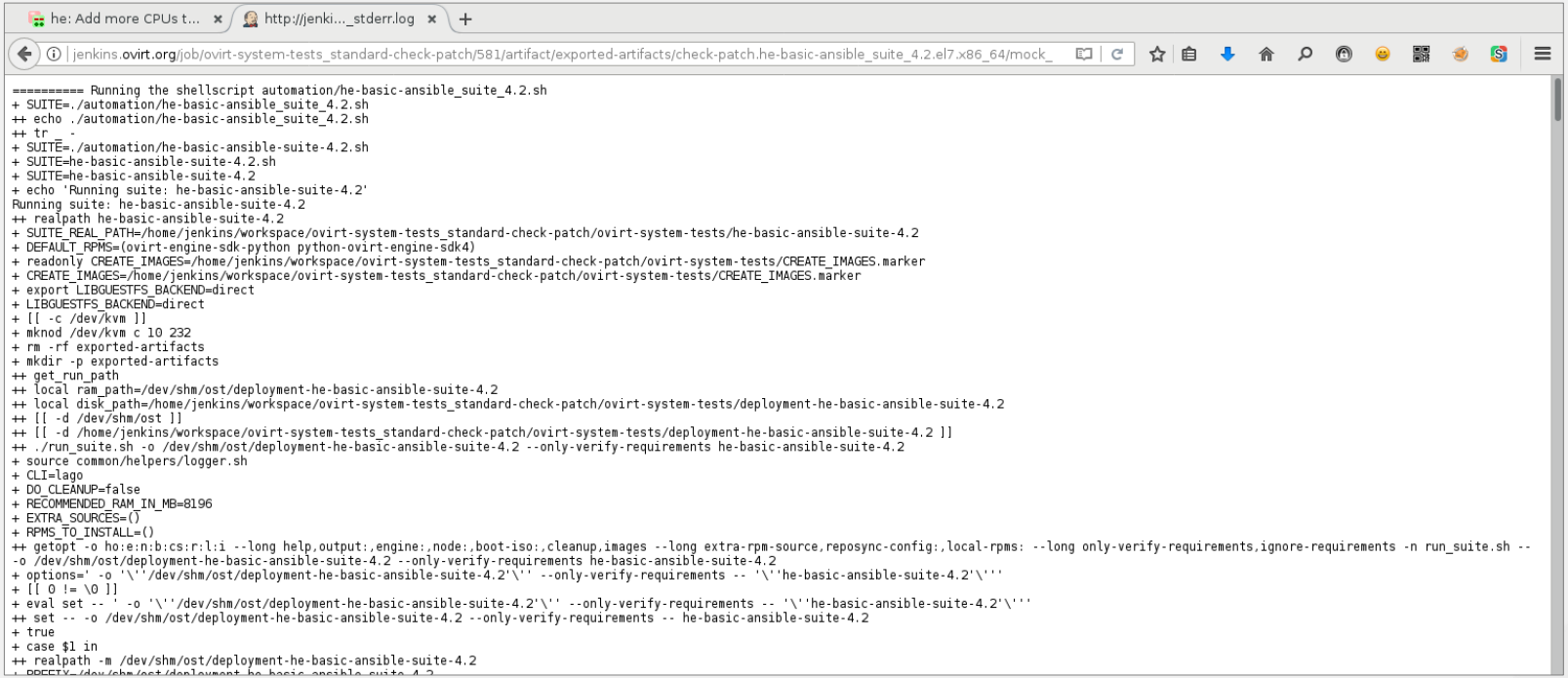
STDCI – DEBUGGING ISSUES

How to find out why is it broken



STDCI – DEBUGGING ISSUES

How to find out why is it broken



MIGRATING TO STDCI V2

Getting all the new goodies

ASK US TO DO IT:

Just email infra-support@ovirt.org

(Opens a Jira ticket)

DIY:

Send a JJB YAML patch to enable V2 for your project

Write an STDCI YAML file for Distros and Archs

Send a JJB YAML patch to remove V1 jobs

RESOURCES

Where to get more information

SOURCE CODE
https://gerrit.ovirt.org/#/admin/projects/jenkins

DOCS
http://ovirt-infra-docs.readthedocs.io/en/latest/CI/Build_and_test_standards/

DOWNSTREAM DOCS
https://mojo.redhat.com/docs/DOC-1153478

redhat.

THANK YOU

Barak Korren
bkorren@redhat.com