



# OVIRT STDCI V2

Introduction for oVirt developers

Barak Korren

RHV DevOps – CI team lead

[bkorren@redhat.com](mailto:bkorren@redhat.com)

# WHAT IS STDCI

The premise of STDCI (CI separation of concerns)

## DEVELOPERS CONFIGURE

How to build source code

How to test source code

Target operating systems

Target architectures

Target oVirt versions

## INFRA TEAM TAKES CARE OF

Jenkins

Jobs

Slaves

Plugins

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

## INFRA TEAM TAKES CARE OF

Jenkins

Jobs

Slaves

Plugins

Servers

Networks

Configured in the  
'jenkins' repo



# 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

## INFRA TEAM TAKES CARE OF

Jenkins

Jobs

Slaves

Plugins

Servers

Networks

Developers  
needed to know  
about

Configured in the  
'jenkins' repo

# 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

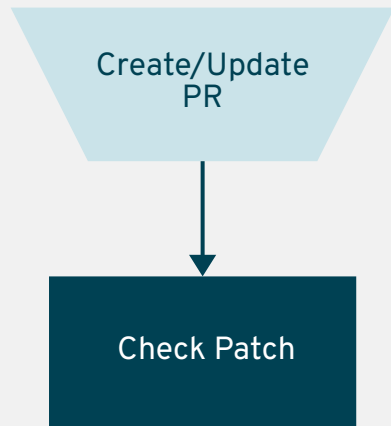
Code life-cycle - when do things run



Create/Update  
PR

# STDCI STAGES

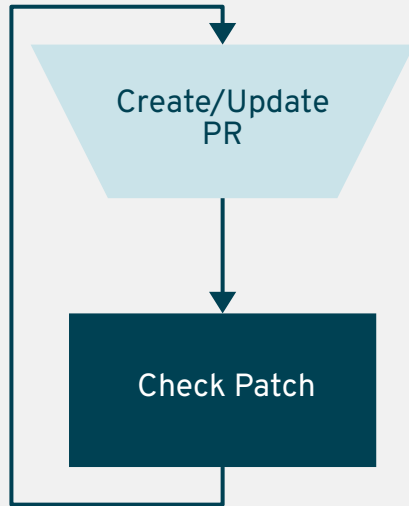
Code life-cycle - when do things run





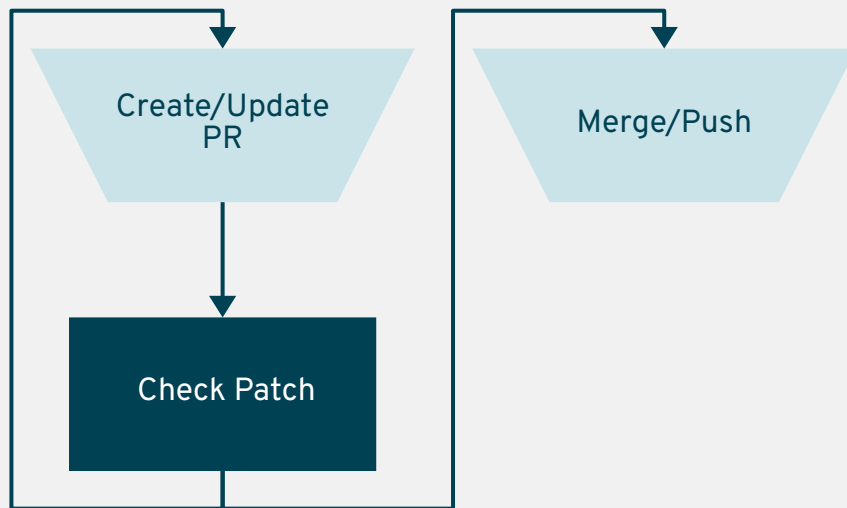
# STDCI STAGES

Code life-cycle - when do things run



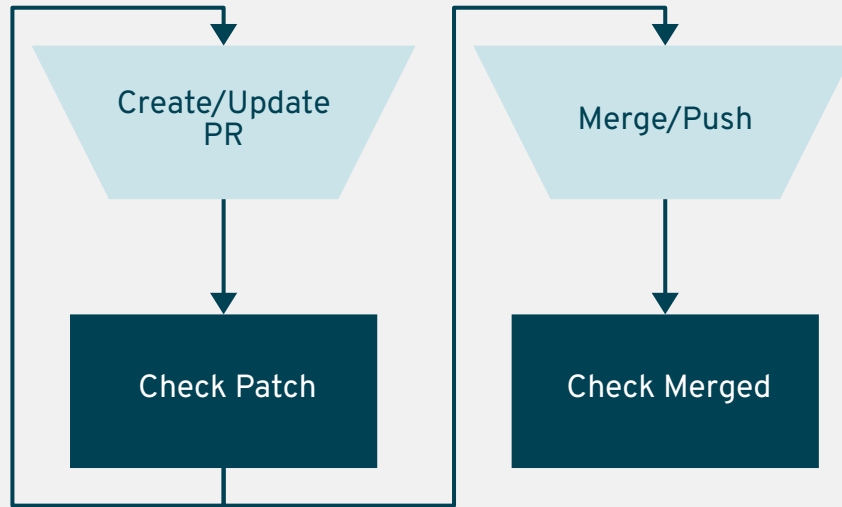
# STDCI STAGES

Code life-cycle - when do things run



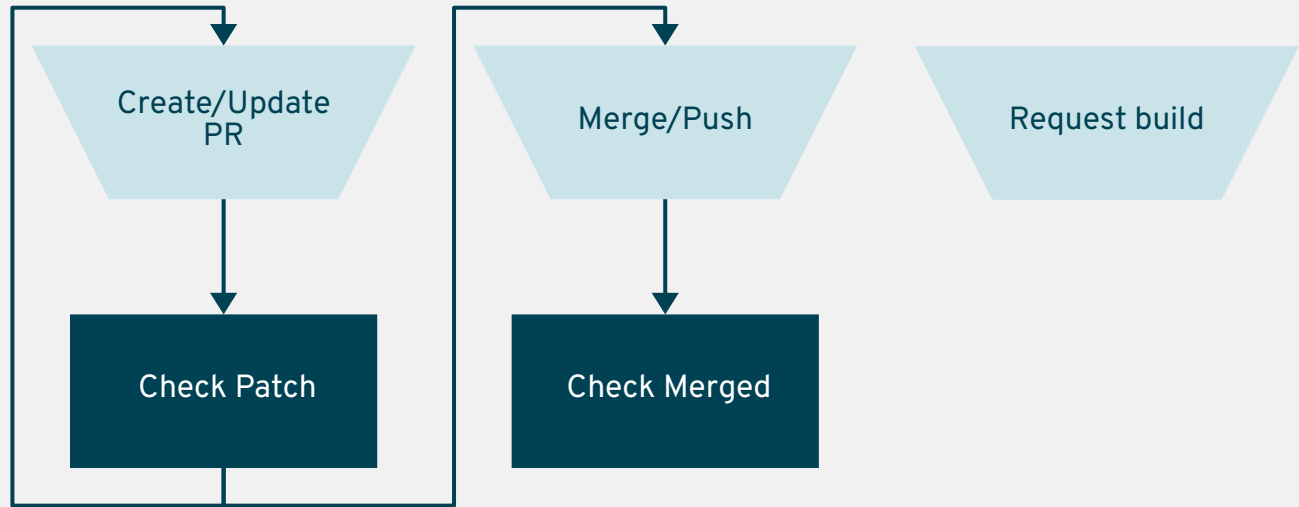
# STDCI STAGES

Code life-cycle - when do things run



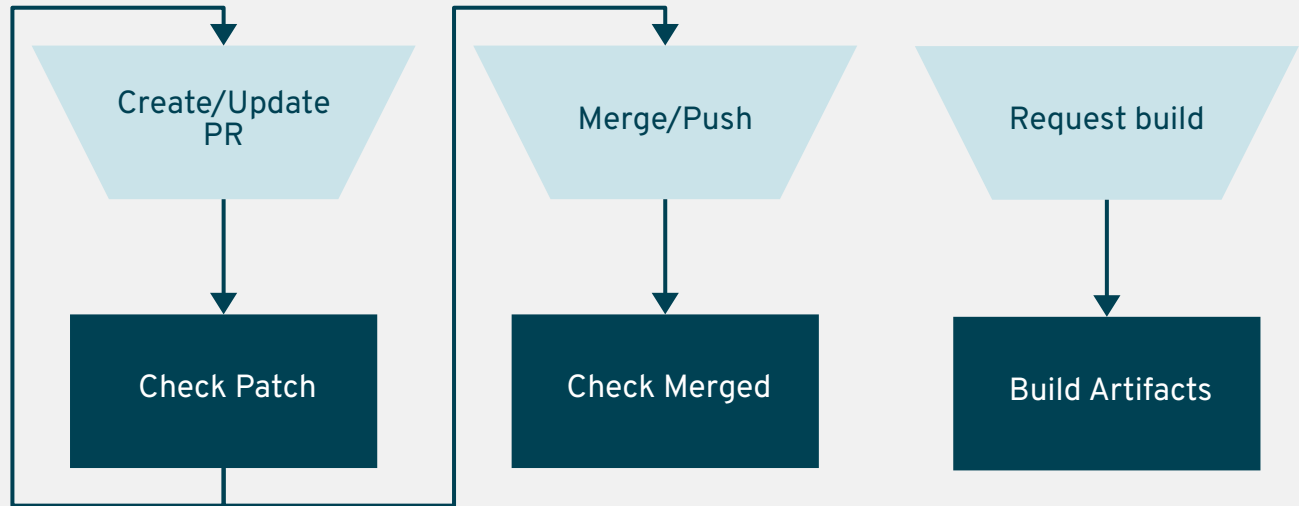
# STDCI STAGES

Code life-cycle - when do things run



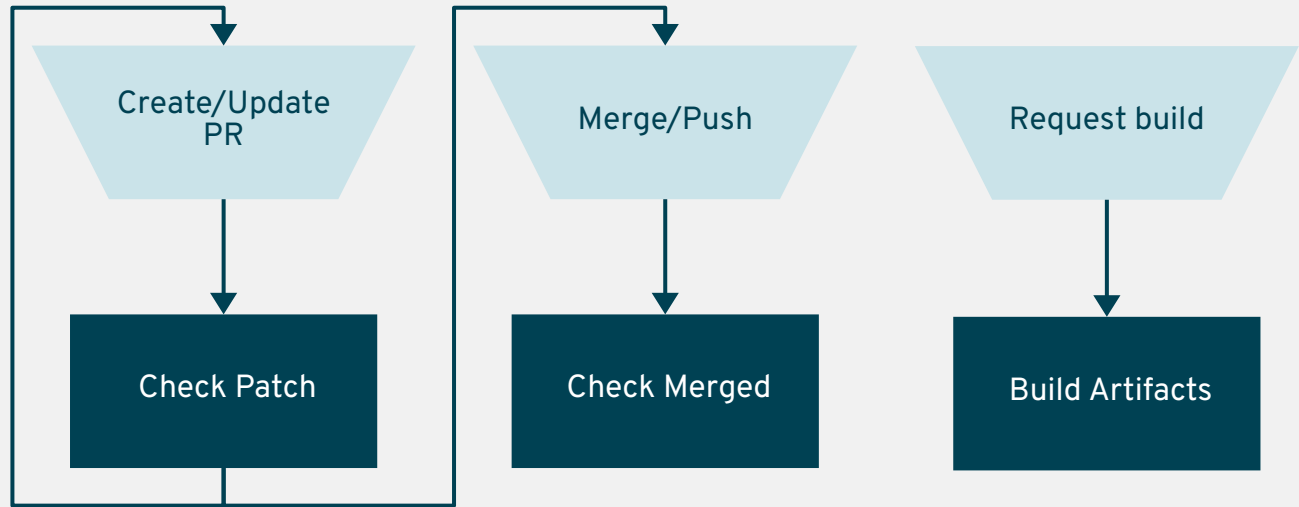
# STDCI STAGES

Code life-cycle - when do things run



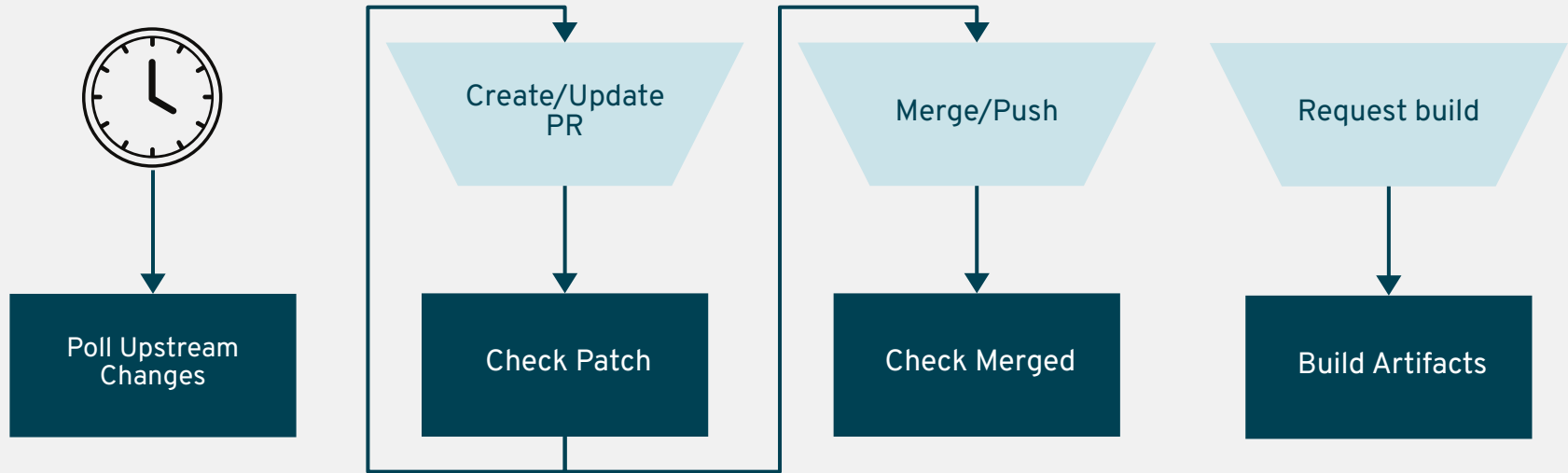
# STDCI STAGES

Code life-cycle - when do things run



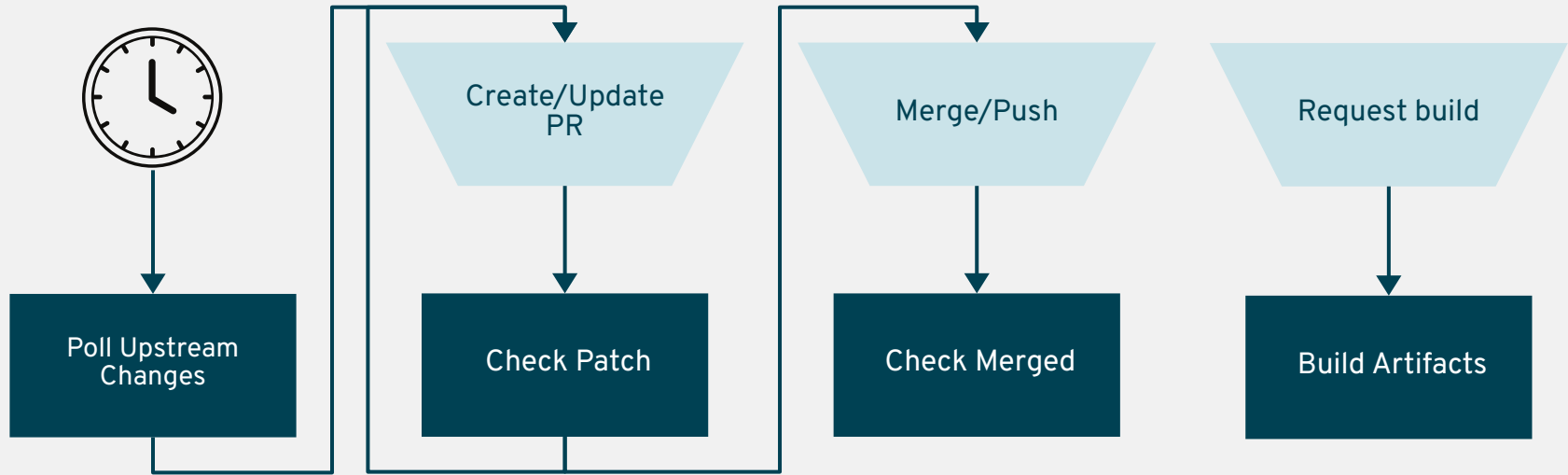
# STDCI STAGES

Code life-cycle - when do things run



# STDCI STAGES

Code life-cycle - when do things run



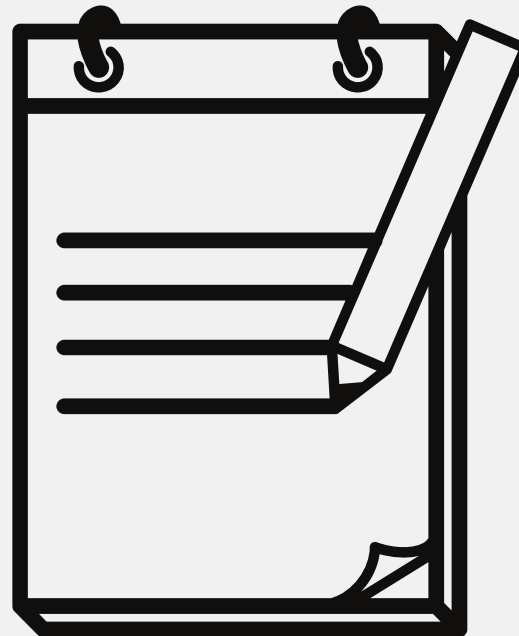


# STDCI DSL

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 than the name of the STDCI stage.

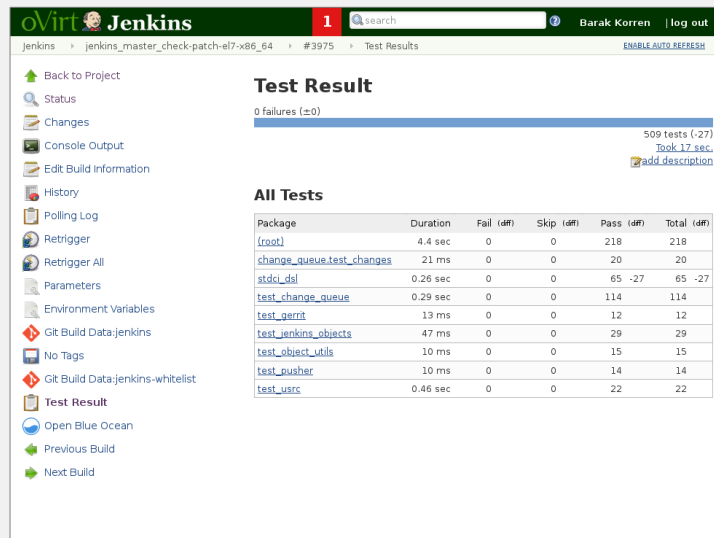
In the future we will allow setting these directly from YAML, but not yet.

# 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



The screenshot shows the Jenkins Test Result page for a build named 'jenkins\_master\_check-patch-el7-x86\_64' with build number #3975. The page displays a summary of test results and a table of individual test cases.

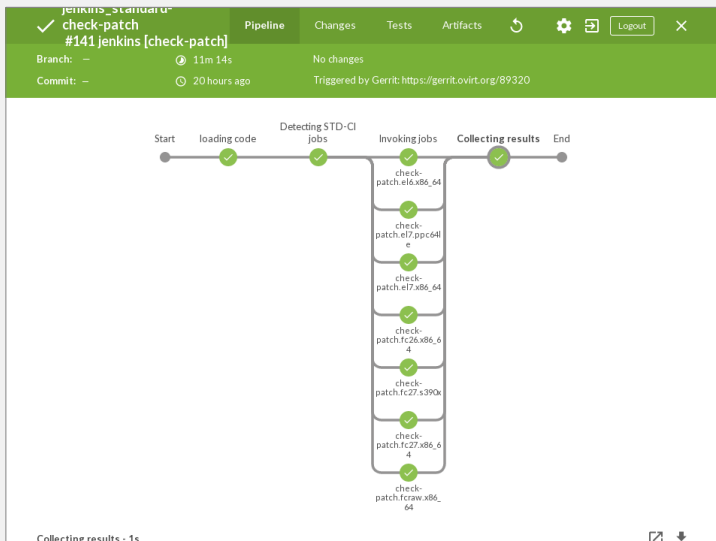
**Test Result**  
0 failures (±0)  
509 tests (-27)  
Took 17 sec  
[add description](#)

**All Tests**

| Package                                   | Duration | Fail (dfr) | Skip (dfr) | Pass (dfr) | Total (dfr) |
|---|----------|------------|------------|------------|-------------|
| <a href="#">[root]</a>                    | 4.4 sec  | 0          | 0          | 218        | 218         |
| <a href="#">change_queue.test_changes</a> | 21 ms    | 0          | 0          | 20         | 20          |
| <a href="#">stdcl_dsl</a>                 | 0.26 sec | 0          | 0          | 65         | 65 -27      |
| <a href="#">test_change_queue</a>         | 0.29 sec | 0          | 0          | 114        | 114         |
| <a href="#">test_gerrit</a>               | 13 ms    | 0          | 0          | 12         | 12          |
| <a href="#">test_jenkins_objects</a>      | 47 ms    | 0          | 0          | 29         | 29          |
| <a href="#">test_object_utils</a>         | 10 ms    | 0          | 0          | 15         | 15          |
| <a href="#">test_pusher</a>               | 10 ms    | 0          | 0          | 14         | 14          |
| <a href="#">test_usrc</a>                 | 0.46 sec | 0          | 0          | 22         | 22          |

# STDCI IMPLEMENTATION

How we've built it

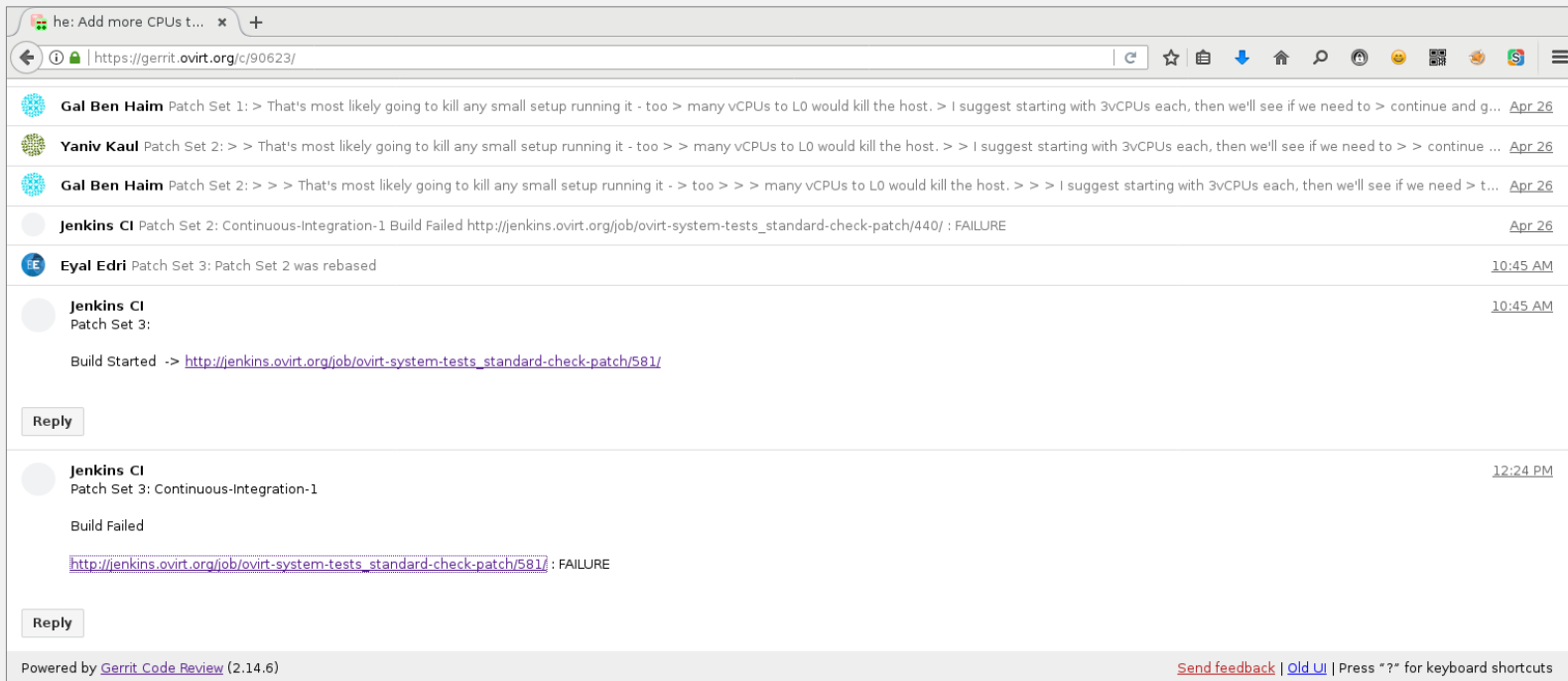


The STDCI V2 implementation is using:

- 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



The screenshot shows a web browser window displaying a Gerrit code review page. The address bar shows the URL <https://gerrit.ovirt.org/c/90623/>. The page content includes several comments from users Gal Ben Haim, Yaniv Kaul, and Jenkins CI, discussing the issue of too many vCPUs being allocated to L0, which could kill the host. The comments mention patch sets and build failures. A 'Reply' button is visible below the last comment. At the bottom of the page, there is a footer with the text 'Powered by [Gerrit Code Review](#) (2.14.6)' and a link to 'Send feedback | [Old UI](#) | Press "?" for keyboard shortcuts'.

he: Add more CPUs t... x +

<https://gerrit.ovirt.org/c/90623/>

**Gal Ben Haim** Patch Set 1: > That's most likely going to kill any small setup running it - too > many vCPUs to L0 would kill the host. > I suggest starting with 3vCPUs each, then we'll see if we need to > continue and g... [Apr 26](#)

**Yaniv Kaul** Patch Set 2: > > That's most likely going to kill any small setup running it - too > > many vCPUs to L0 would kill the host. > > I suggest starting with 3vCPUs each, then we'll see if we need to > > continue ... [Apr 26](#)

**Gal Ben Haim** Patch Set 2: > > > That's most likely going to kill any small setup running it - > too > > > many vCPUs to L0 would kill the host. > > > I suggest starting with 3vCPUs each, then we'll see if we need > t... [Apr 26](#)

**Jenkins CI** Patch Set 2: Continuous-Integration-1 Build Failed [http://jenkins.ovirt.org/job/ovirt-system-tests\\_standard-check-patch/440/](http://jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/440/) : FAILURE [Apr 26](#)

**Eyal Edri** Patch Set 3: Patch Set 2 was rebased [10:45 AM](#)

**Jenkins CI** Patch Set 3: [10:45 AM](#)

Build Started -> [http://jenkins.ovirt.org/job/ovirt-system-tests\\_standard-check-patch/581/](http://jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/)

**Reply**

**Jenkins CI** Patch Set 3: Continuous-Integration-1 [12:24 PM](#)

Build Failed

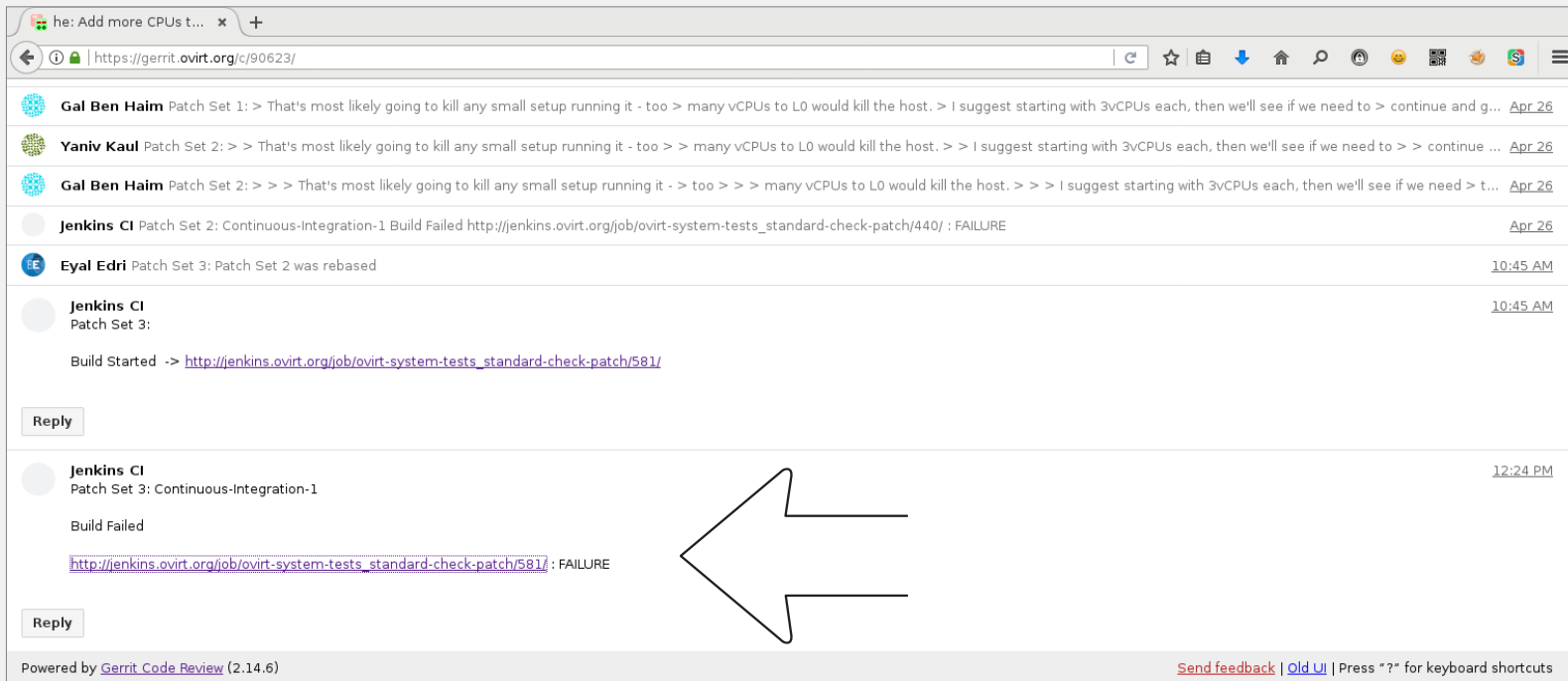
[http://jenkins.ovirt.org/job/ovirt-system-tests\\_standard-check-patch/581/](http://jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/) : FAILURE

**Reply**

Powered by [Gerrit Code Review](#) (2.14.6) [Send feedback](#) | [Old UI](#) | Press "?" for keyboard shortcuts

# STDCI – DEBUGGING ISSUES

How to find out why is it broken



The screenshot shows a web browser window displaying a Gerrit code review page. The address bar shows the URL <https://gerrit.ovirt.org/c/90623/>. The page content includes several comments from users: Gal Ben Haim, Yaniv Kaul, and Eyal Edri. A Jenkins CI comment indicates a build failure for patch set 3, with a link to [http://jenkins.ovirt.org/job/ovirt-system-tests\\_standard-check-patch/581/](http://jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/). A large white arrow is drawn on the page, pointing from the right towards the failure message.

Gal Ben Haim Patch Set 1: > That's most likely going to kill any small setup running it - too > many vCPUs to L0 would kill the host. > I suggest starting with 3vCPUs each, then we'll see if we need to > continue and g... [Apr 26](#)

Yaniv Kaul Patch Set 2: >> That's most likely going to kill any small setup running it - too >> many vCPUs to L0 would kill the host. >> I suggest starting with 3vCPUs each, then we'll see if we need to >> continue ... [Apr 26](#)

Gal Ben Haim Patch Set 2: >>> That's most likely going to kill any small setup running it - > too >>> many vCPUs to L0 would kill the host. >>> I suggest starting with 3vCPUs each, then we'll see if we need > t... [Apr 26](#)

Jenkins CI Patch Set 2: Continuous-Integration-1 Build Failed [http://jenkins.ovirt.org/job/ovirt-system-tests\\_standard-check-patch/440/](http://jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/440/) : FAILURE [Apr 26](#)

Eyal Edri Patch Set 3: Patch Set 2 was rebased [10:45 AM](#)

Jenkins CI Patch Set 3: [10:45 AM](#)

Build Started -> [http://jenkins.ovirt.org/job/ovirt-system-tests\\_standard-check-patch/581/](http://jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/)

Reply

Jenkins CI Patch Set 3: Continuous-Integration-1 [12:24 PM](#)

Build Failed

[http://jenkins.ovirt.org/job/ovirt-system-tests\\_standard-check-patch/581/](http://jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/) : FAILURE

Reply

Powered by [Gerrit Code Review](#) (2.14.6) [Send feedback](#) | [Old UI](#) | Press "?" for keyboard shortcuts

# STDCI – DEBUGGING ISSUES

How to find out why is it broken

The screenshot shows a Jenkins web interface for a build named "#581 ovirt-system-tests [check-patch]". The browser address bar shows the URL "jenkins.ovirt.org/job/ovirt-system-tests\_standard-check-patch/581/". The left sidebar contains navigation options: Back to Project, Status, Changes, Console Output, View as plain text, Edit Build Information, Delete Build, Polling Log, Retrigger, Git Build Data, No Tags, Parameters, Git Build Data, Test Result, Open Blue Ocean, Replay, Pipeline Steps, and Embeddable Build Status. The main content area displays the build title "Build #581 ovirt-system-tests [check-patch] (01-May-2018 07:45:21)" with a red status icon. A "Keep this build forever" button is visible. The build started 4 hr 11 min ago and took 1 hr 38 min. A "add description" link is present. Below the build title, there are sections for "Build Artifacts", "Triggered by Gerrit: https://gerrit.ovirt.org/90623", and "This run spent:" with a list of times: 7.8 sec waiting in the queue, 1 hr 38 min building on an executor, and 1 hr 38 min total from scheduled to completion. There are two "git" sections, each with a revision ID and a "myhead" branch. The first revision is 96befcf6f0b26e028165bc45bb5327f0a4e6dc24. The second revision is 8b7609d4529707c2798caa22f64698d777cce371. At the bottom, there is a "Test Result" section showing "1 failure / +1" and a link to "004\_basic\_sanitary.snapshot\_merge".

# STDCI – DEBUGGING ISSUES

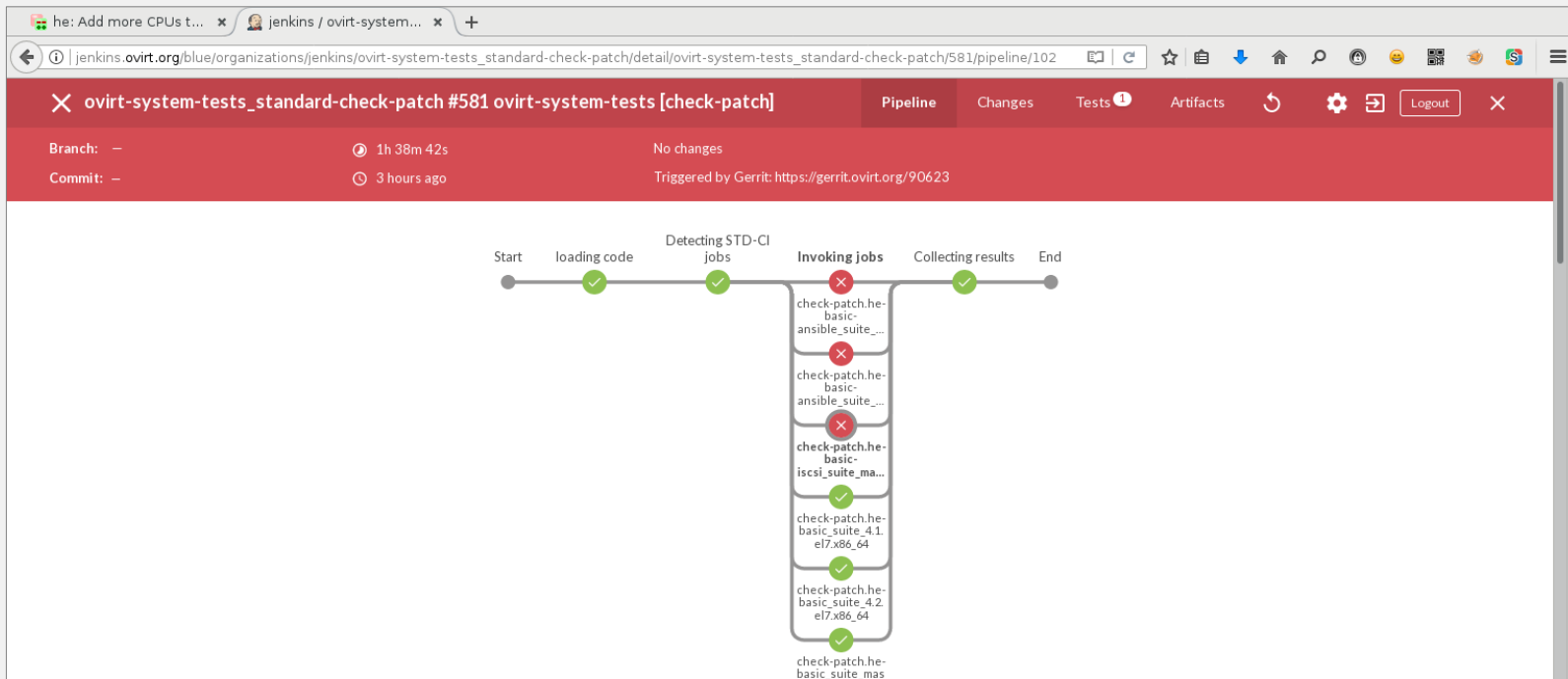
How to find out why is it broken

The screenshot shows a Jenkins build page for job #581, 'ovirt-system-tests [check-patch]'. The build is in a failed state, indicated by a red circle icon. The main heading is 'Build #581 ovirt-system-tests [check-patch] (01-May-2018 07:45:21)'. On the right, a blue button says 'Keep this build forever'. Below this, it shows 'Started 4 hr 11 min ago' and 'Took 1 hr 38 min'. There is an 'add description' link. The left sidebar contains various navigation options: Back to Project, Status, Changes, Console Output, View as plain text, Edit Build Information, Delete Build, Polling Log, Retrigger, Git Build Data, No Tags, Parameters, Git Build Data, Test Result, Open Blue Ocean, Replay, Pipeline Steps, and Embeddable Build Status. The main content area shows 'Build Artifacts', 'Triggered by Gerrit: https://gerrit.ovirt.org/90623', and 'This run spent: 7.8 sec waiting in the queue; 1 hr 38 min building on an executor; 1 hr 38 min total from scheduled to completion.' Below this, two 'git' revisions are listed: 'Revision: 96befcf6f0b26e028165bc45bb5327f0a4e6dc24' with 'myhead' and 'Revision: 8b7609d4529707c2798caa22f64698d777cce371' with 'myhead'. A 'Test Result' section shows '(1 failure / +1)' and a link to '004\_basic\_sanitary.snapshot\_merge'. A large white arrow points from the 'Test Result' section to the 'Test Result' link in the sidebar.



# STDCI – DEBUGGING ISSUES

How to find out why is it broken



# STDCI – DEBUGGING ISSUES

How to find out why is it broken

The screenshot shows a Jenkins pipeline view for 'ovirt-system-tests [check-patch]'. The pipeline is currently running, with a duration of 1h 38m 42s. The commit is triggered by Gerrit. The pipeline consists of several stages: 'Start', 'loading code', 'Detecting STD-CI jobs', 'Invoking jobs', 'Collecting results', and 'End'. The 'Invoking jobs' stage is highlighted in red, indicating a failure. A large white arrow points to the failed jobs in this stage.

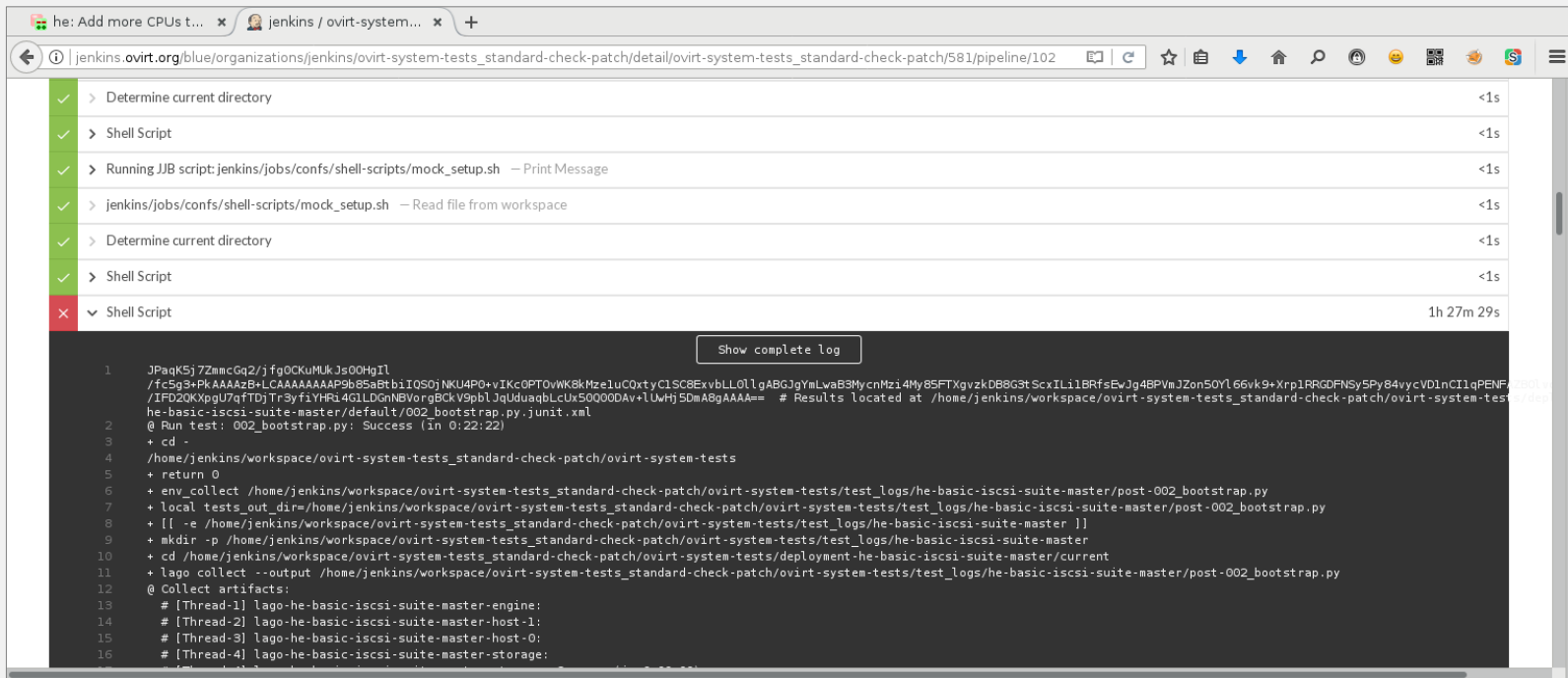
| Stage                 | Status  |
|-----------------------|---------|
| Start                 | Success |
| loading code          | Success |
| Detecting STD-CI jobs | Success |
| Invoking jobs         | Failure |
| Collecting results    | Success |
| End                   | Success |

The 'Invoking jobs' stage contains the following jobs:

- check-patch.he-basic-ansible\_suite\_... (Failure)
- check-patch.he-basic-ansible\_suite\_... (Failure)
- check-patch.he-basic-iscsi\_suite\_ma... (Failure)
- check-patch.he-basic\_suite\_4.1.e17.x86\_64 (Success)
- check-patch.he-basic\_suite\_4.2.e17.x86\_64 (Success)
- check-patch.he-basic\_suite\_mas (Success)

# STDCI – DEBUGGING ISSUES

How to find out why is it broken



The screenshot shows a Jenkins pipeline execution page for a job named 'ovirt-system-tests\_standard-check-patch'. The pipeline consists of several steps, most of which are successful (indicated by green checkmarks). The final step, 'Shell Script', has failed (indicated by a red 'X').

The failed step's log output is as follows:

```
1 JPaqK5j 7ZmcGq2/jfg0CKuMUKJs00HgIl
2 /fc5g+Pk AAAAzB+LCAAAAAA99b85aBtbiI0SDjNKU4P0+vTKCOPTovWK8kMze1uCOxytC1SC8ExvLL0LlgABGJgYmLwAB3MyCnMzi4My85FTXgvzkDBBGStScxILi1BRfsEvJg4BPVnJZon5OYl66vk9+Xrp1RRGFNSy5Py84vycVD1nCI1qPENF...
3 /IFD20KXpgU7fDjTr3yfiYHR14GLLDGnNBVorgBCKV9pblJqUduaqLcUx50000DAV+1UwHj50mABgAAAA= # Results located at /home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests_deploy
4 @ Run test: 002_bootstrap.py: Success (in 0:22:22)
5 + cd -
6 + /home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests
7 + return 0
8 + env_collect /home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/test_logs/he-basic-iscsi-suite-master/post-002_bootstrap.py
9 + local tests_out_dir=/home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/test_logs/he-basic-iscsi-suite-master/post-002_bootstrap.py
10 + [[ -e /home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/test_logs/he-basic-iscsi-suite-master ]]
11 + mkdir -p /home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/test_logs/he-basic-iscsi-suite-master
12 + cd /home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/deployment-he-basic-iscsi-suite-master/current
13 + lagoon collect --output /home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/test_logs/he-basic-iscsi-suite-master/post-002_bootstrap.py
14 @ Collect artifacts:
15 # [Thread-1] lagoon-he-basic-iscsi-suite-master-engine:
16 # [Thread-2] lagoon-he-basic-iscsi-suite-master-host-1:
17 # [Thread-3] lagoon-he-basic-iscsi-suite-master-host-0:
18 # [Thread-4] lagoon-he-basic-iscsi-suite-master-storage:
```

# STDCI – DEBUGGING ISSUES

How to find out why is it broken

The screenshot shows a Jenkins web interface for a build job named 'ovirt-system-tests [check-patch]'. The build number is 581, and it was completed on May 01, 2018, at 07:45:21. The build status is 'Failed' (indicated by a red circle with a white exclamation mark). The left sidebar contains navigation options such as 'Back to Project', 'Status', 'Changes', 'Console Output', 'View as plain text', 'Edit Build Information', 'Delete Build', 'Polling Log', 'Retrigger', 'Git Build Data', 'No Tags', 'Parameters', 'Test Result', 'Open Blue Ocean', 'Replay', 'Pipeline Steps', and 'Embeddable Build Status'. The main content area displays the build title, a 'Keep this build forever' button, and the start and end times. Below this, there are links for 'Build Artifacts', 'Triggered by Gerrit: https://gerrit.ovirt.org/90623', and 'This run spent:'. The 'This run spent' section lists the following times: 7.8 sec waiting in the queue; 1 hr 38 min building on an executor; and 1 hr 38 min total from scheduled to completion. There are two 'git' revision entries, both pointing to 'myhead'. The first revision is 96b6fc6f0b26e028165bc45bb5327f0a4e6dc24. The second revision is 8b7609d4529707c2798caa22f64698d777cce371. At the bottom, there is a 'Test Result' section showing '1 failure / +1' and a link to '004\_basic\_sanitary.snapshot\_merge'. The browser's address bar shows the URL 'jenkins.ovirt.org/job/ovirt-system-tests\_standard-check-patch/581/'.

# STDCI – DEBUGGING ISSUES

How to find out why is it broken

jenkins.ovirt.org/job/ovirt-system-tests\_standard-check-patch/581/

jenkins > ovirt-system-tests\_standard-check-patch > #581 ovirt-system-tests [check-patch] [ENABLE AUTO REFRESH](#)

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[View as plain text](#)

[Edit Build Information](#)

[Delete Build](#)

[Polling Log](#)

[Retrigger](#)

[Git Build Data](#)

[No Tags](#)

[Parameters](#)

[Git Build Data](#)

[Test Result](#)

[Open Blue Ocean](#)

[Replay](#)

[Pipeline Steps](#)

[Embeddable Build Status](#)

**Build #581 ovirt-system-tests [check-patch]**  
**(01-May-2018 07:45:21)**

[Keep this build forever](#)

Started 4 hr 11 min ago  
Took [1 hr 38 min](#)  
[add description](#)

[Build Artifacts](#)

Triggered by Gerrit: <https://gerrit.ovirt.org/90623>

This run spent:

- 7.8 sec waiting in the queue;
- 1 hr 38 min building on an executor;
- 1 hr 38 min total from scheduled to completion.

**git** Revision: 96b6fcf60b26e028165bc45bb5327f0a4e6dc24

- myhead

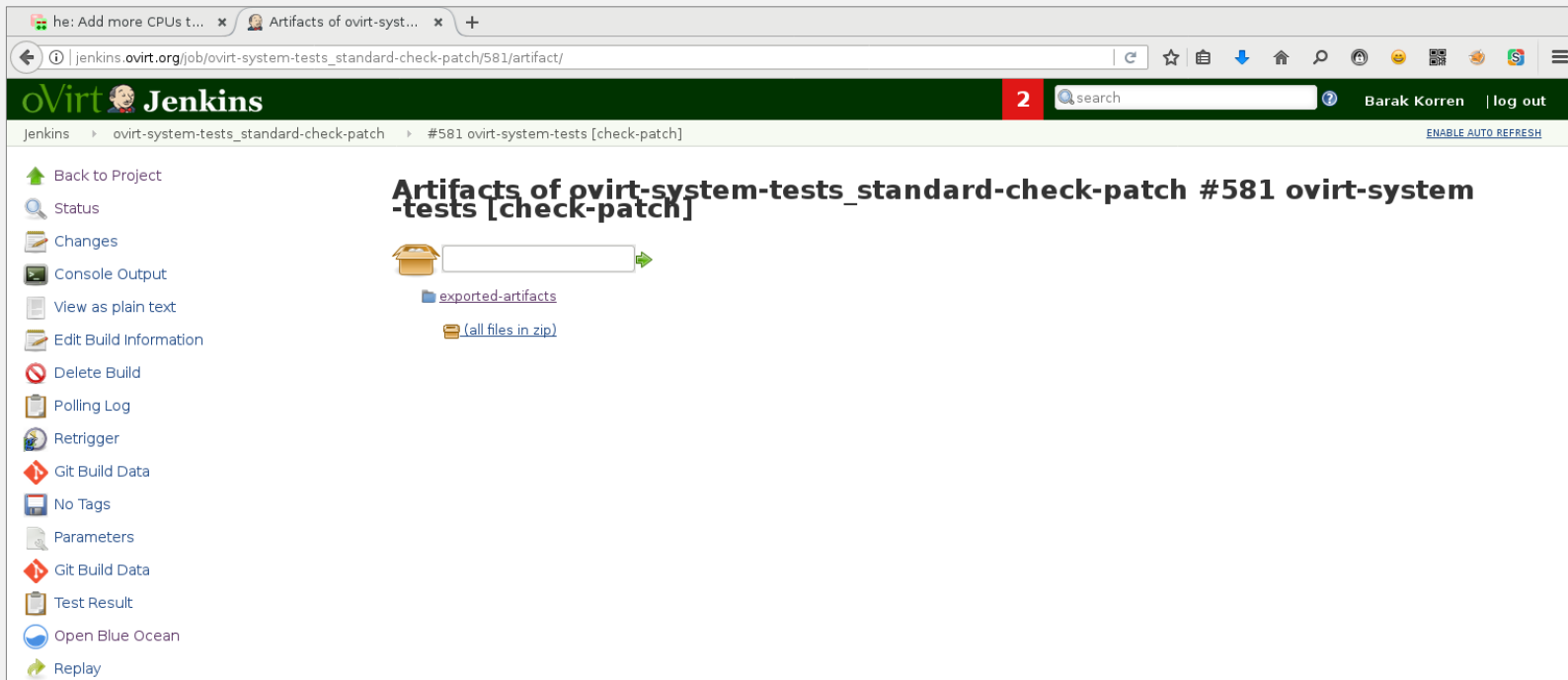
**git** Revision: 8b7609d4529707c2798caa22f64698d777cce371

- myhead

[Test Result](#) (1 failure / +1)  
[004\\_basic\\_sanitary.snapshot\\_merge](#)

# STDCI – DEBUGGING ISSUES

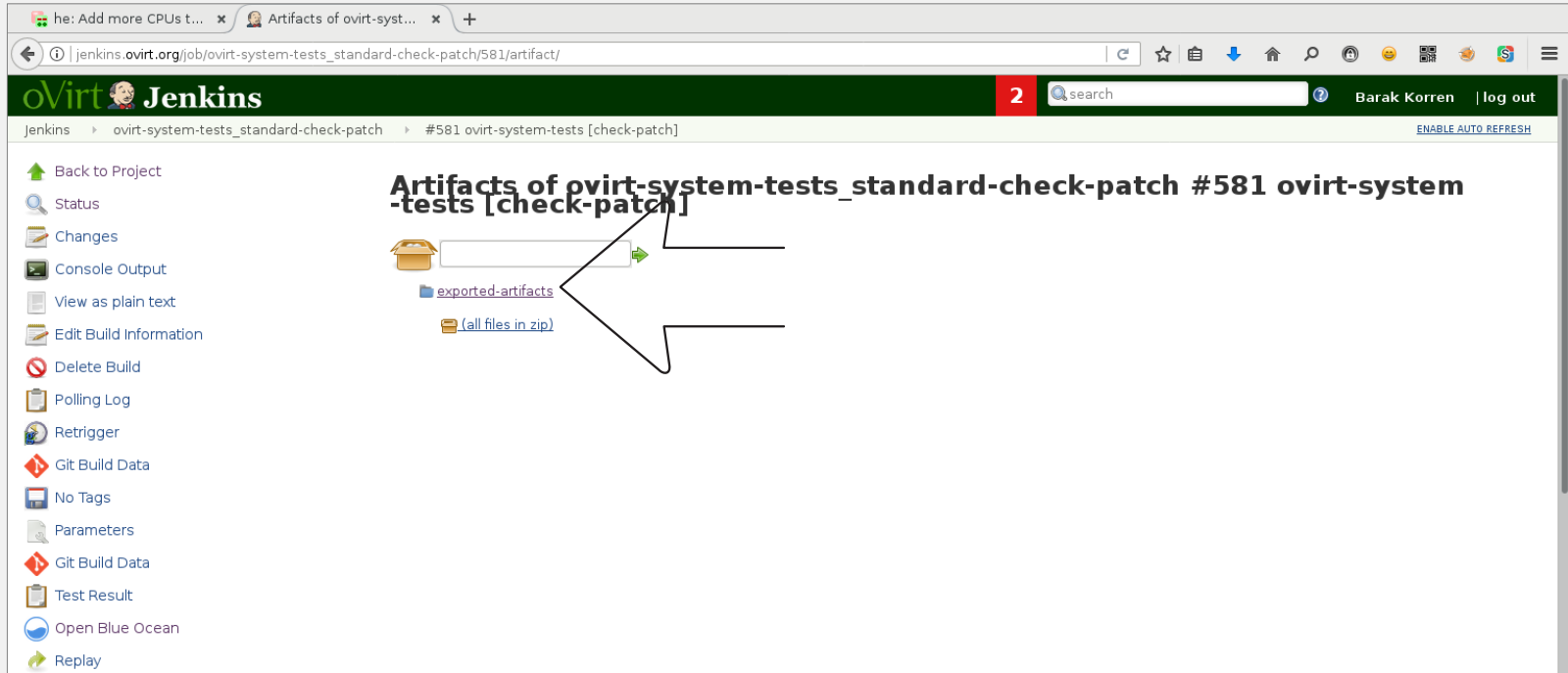
How to find out why is it broken



The screenshot shows a web browser window displaying the Jenkins interface. The address bar shows the URL: `jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/artifact/`. The page title is "Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]". The main content area displays a folder icon with a search input field, and below it, two links: `exported-artifacts` and `(all files in zip)`. The left sidebar contains a list of navigation options: Back to Project, Status, Changes, Console Output, View as plain text, Edit Build Information, Delete Build, Polling Log, Retrigger, Git Build Data, No Tags, Parameters, Git Build Data, Test Result, Open Blue Ocean, and Replay. The top navigation bar includes the Jenkins logo, a search bar, and the user name "Barak Korren" with a "log out" link.

# STDCI – DEBUGGING ISSUES

How to find out why is it broken



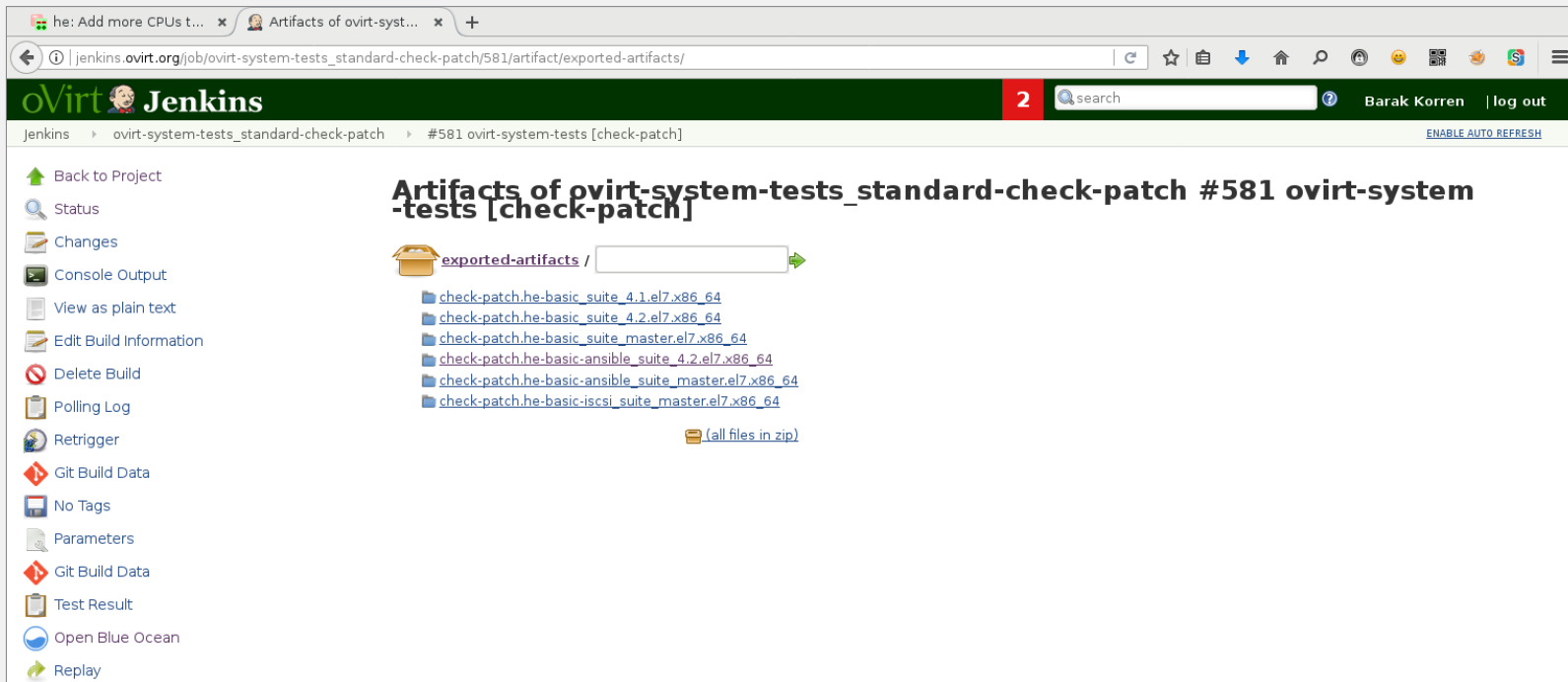
The screenshot shows the Jenkins web interface for a failed build. The browser address bar displays the URL: `jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/artifact/`. The Jenkins header shows the job name `ovirt-system-tests_standard-check-patch` and build number `#581 ovirt-system-tests [check-patch]`. A red status bar at the top right indicates a failed build with the number `2`. The main content area is titled `Artifacts of ovirt-system-tests_standard-check-patch #581 ovirt-system-tests [check-patch]`. A search bar is present above the artifact list. The artifact list contains:

- `exported-artifacts` (with a folder icon)
- `(all files in zip)` (with a zip file icon)

A large white arrow points from the text `Artifacts of ovirt-system-tests_standard-check-patch #581 ovirt-system-tests [check-patch]` to the `exported-artifacts` folder icon. A second white arrow points from the `exported-artifacts` folder icon to the `(all files in zip)` zip file icon. On the left side, a navigation menu lists various options such as `Back to Project`, `Status`, `Changes`, `Console Output`, `View as plain text`, `Edit Build Information`, `Delete Build`, `Polling Log`, `Retrigger`, `Git Build Data`, `No Tags`, `Parameters`, `Test Result`, `Open Blue Ocean`, and `Replay`.

# STDCI – DEBUGGING ISSUES

How to find out why is it broken



The screenshot shows a web browser window displaying the Jenkins interface. The address bar shows the URL: `jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/artifact/exported-artifacts/`. The page title is "Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]". The main content area shows a list of artifacts under the heading "exported-artifacts /". The artifacts listed are:

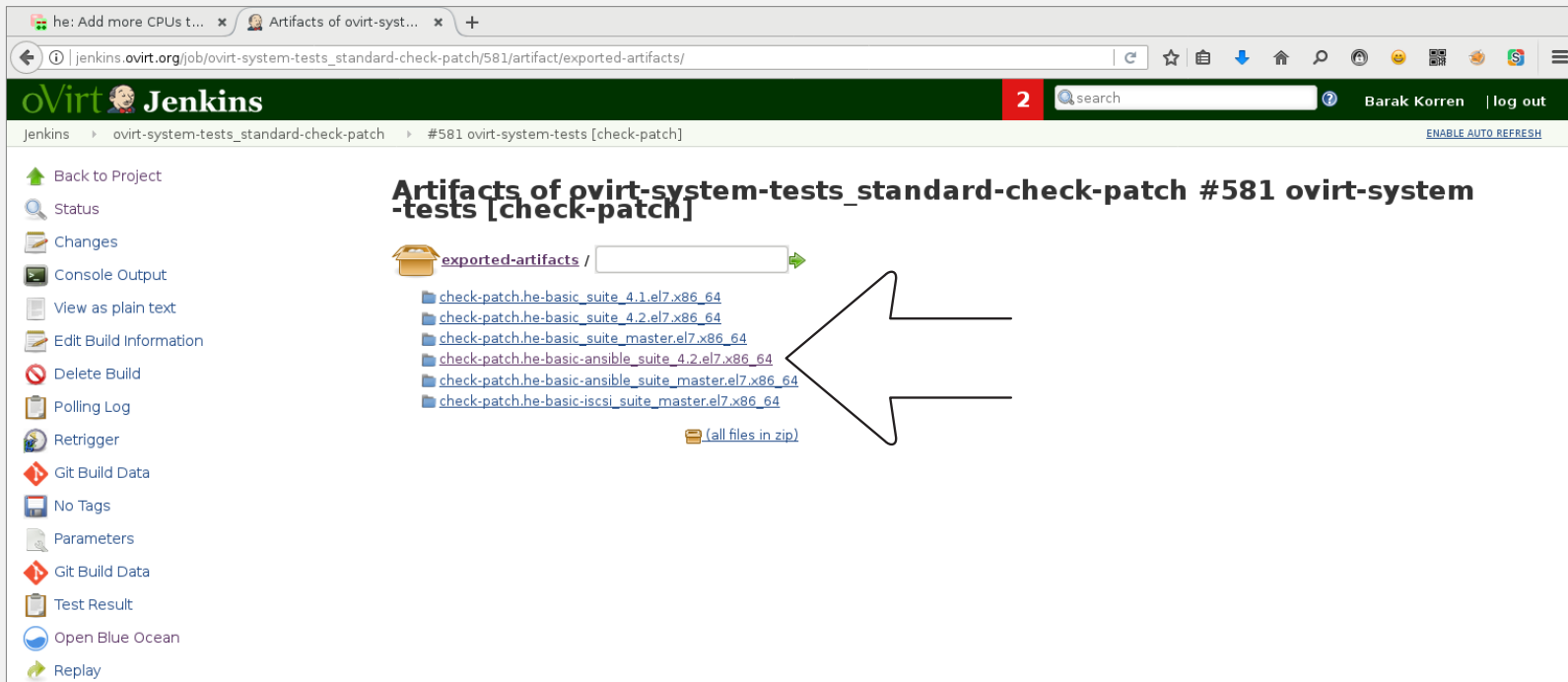
- [check-patch.he-basic\\_suite\\_4.1.el7.x86\\_64](#)
- [check-patch.he-basic\\_suite\\_4.2.el7.x86\\_64](#)
- [check-patch.he-basic\\_suite\\_master.el7.x86\\_64](#)
- [check-patch.he-basic-ansible\\_suite\\_4.2.el7.x86\\_64](#)
- [check-patch.he-basic-ansible\\_suite\\_master.el7.x86\\_64](#)
- [check-patch.he-basic-icsci\\_suite\\_master.el7.x86\\_64](#)

There is also a link for [\(all files in zip\)](#). The left sidebar contains various navigation options such as "Back to Project", "Status", "Changes", "Console Output", "View as plain text", "Edit Build Information", "Delete Build", "Polling Log", "Retrigger", "Git Build Data", "No Tags", "Parameters", "Test Result", "Open Blue Ocean", and "Replay".



# STDCI – DEBUGGING ISSUES

How to find out why is it broken



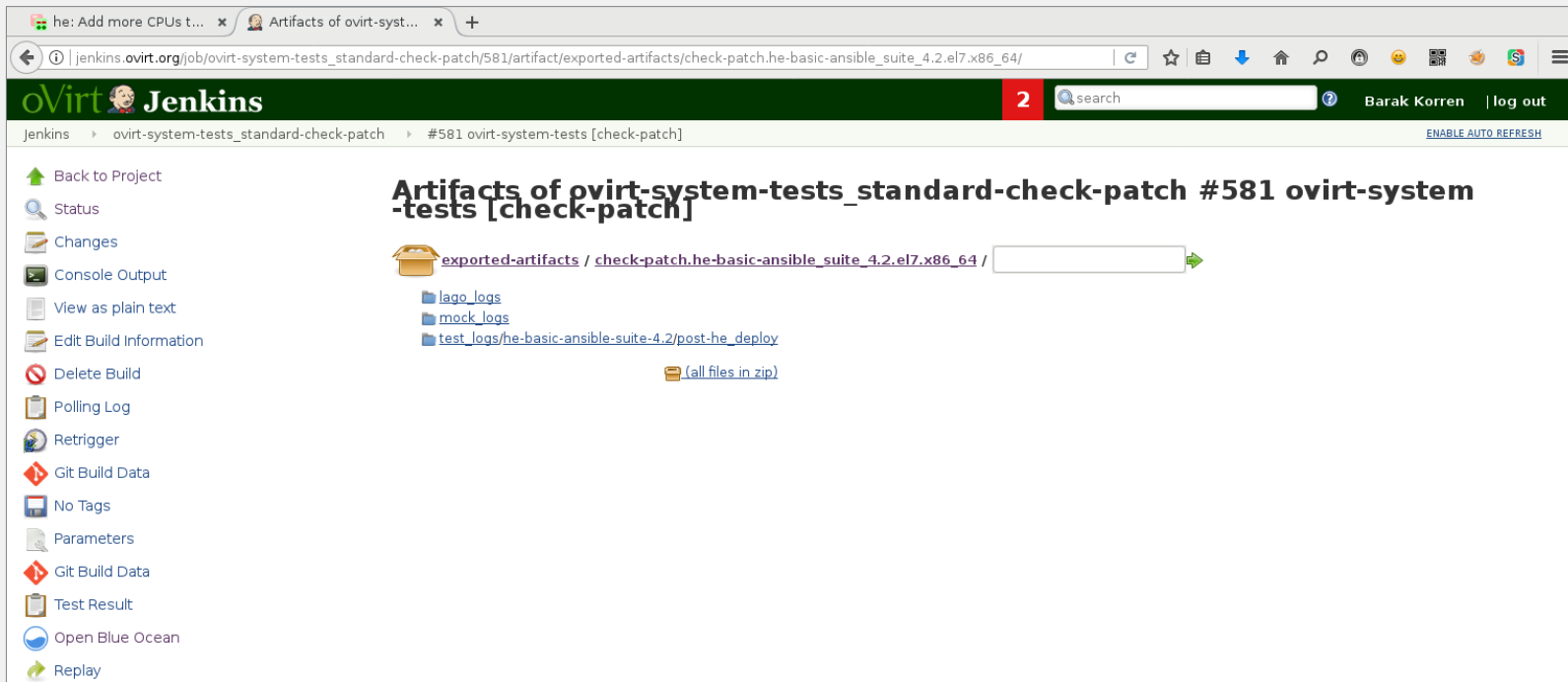
The screenshot shows the Jenkins web interface for a failed build. The browser address bar displays the URL: `jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/artifact/exported-artifacts/`. The page title is "Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]". A sidebar on the left contains navigation options such as "Back to Project", "Status", "Changes", "Console Output", "View as plain text", "Edit Build Information", "Delete Build", "Polling Log", "Retrigger", "Git Build Data", "No Tags", "Parameters", "Git Build Data", "Test Result", "Open Blue Ocean", and "Replay". The main content area shows a list of artifacts under the heading "Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]". The artifacts are:

- [check-patch.he-basic\\_suite\\_4.1.el7.x86\\_64](#)
- [check-patch.he-basic\\_suite\\_4.2.el7.x86\\_64](#)
- [check-patch.he-basic\\_suite\\_master.el7.x86\\_64](#)
- [check-patch.he-basic-ansible\\_suite\\_4.2.el7.x86\\_64](#)
- [check-patch.he-basic-ansible\\_suite\\_master.el7.x86\\_64](#)
- [check-patch.he-basic-iscsi\\_suite\\_master.el7.x86\\_64](#)

Below the list is a link for "(all files in zip)". A large white arrow points from the right side of the artifacts list towards the center of the page.

# STDCI – DEBUGGING ISSUES

How to find out why is it broken



The screenshot shows the Jenkins web interface for a failed build. The browser address bar displays the URL: `jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/artifact/exported-artifacts/check-patch.he-basic-ansible_suite_4.2.el7.x86_64/`. The Jenkins header includes the logo, the name "Jenkins", a notification badge with the number "2", a search bar, and the user name "Barak Korren" with a "log out" link. The breadcrumb navigation shows the path: `Jenkins > ovirt-system-tests_standard-check-patch > #581 ovirt-system-tests [check-patch]`. A link to "ENABLE AUTO REFRESH" is also present.

The main content area displays the title "Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]". Below the title, there is a folder icon and the path `exported-artifacts / check-patch.he-basic-ansible_suite_4.2.el7.x86_64 /` followed by a search input field and a green arrow icon. Underneath, three sub-folders are listed: `lago_logs`, `mock_logs`, and `test_logs/he-basic-ansible-suite-4.2/post-he_deploy`. At the bottom of the artifact list, there is a folder icon and the text `(all files in zip)`.

The left sidebar contains a list of navigation options: "Back to Project", "Status", "Changes", "Console Output", "View as plain text", "Edit Build Information", "Delete Build", "Polling Log", "Retrigger", "Git Build Data", "No Tags", "Parameters", "Git Build Data", "Test Result", "Open Blue Ocean", and "Replay".

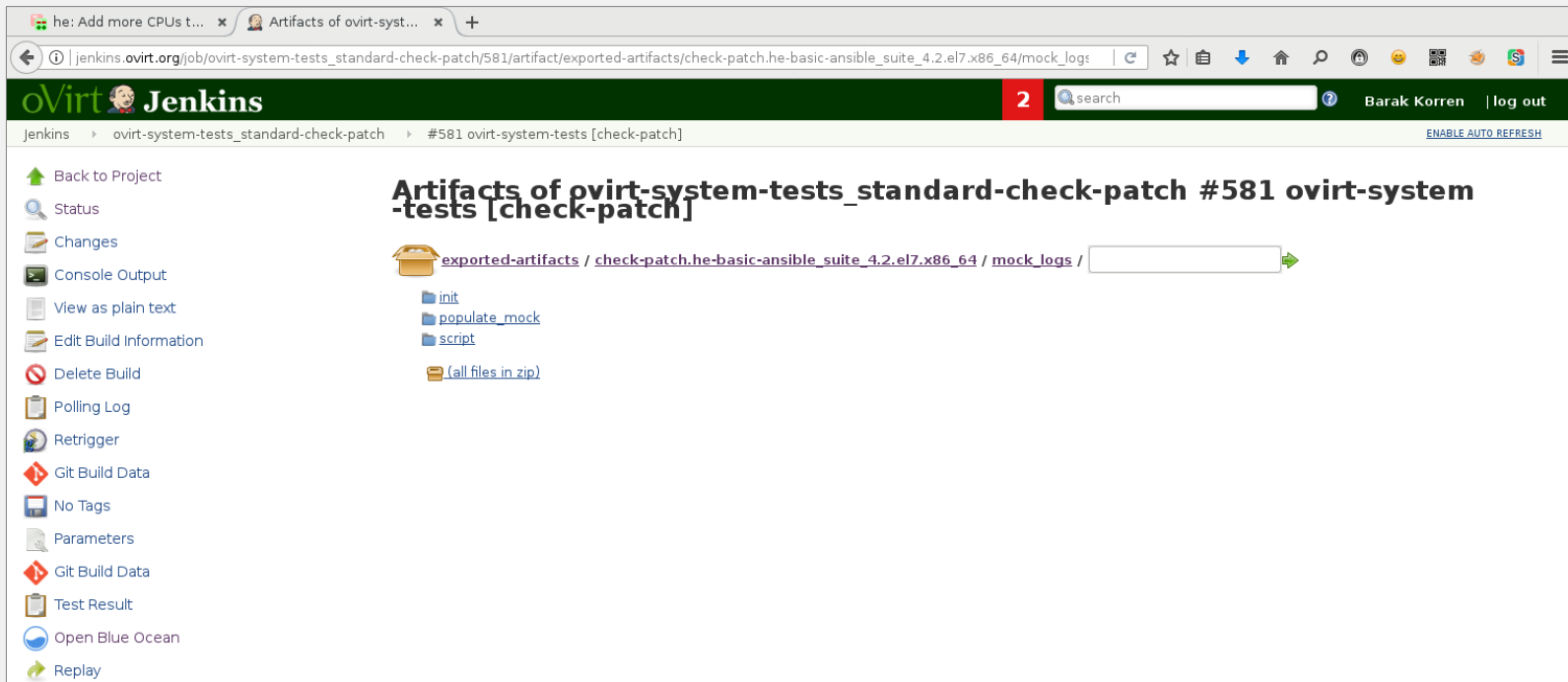
# STDCI – DEBUGGING ISSUES

How to find out why is it broken

The screenshot shows the Jenkins web interface for a build named "ovirt-system-tests [check-patch] #581". The page title is "Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]". The main content area displays a list of artifacts under the path "exported-artifacts / check-patch.he-basic-ansible\_suite\_4.2.el7.x86\_64 /". The artifacts listed are "lago\_logs", "mock\_logs", "test\_logs/he-basic-ansible-suite-4.2/post-he\_deploy", and "(all files in zip)". A large white arrow points from the "test\_logs/he-basic-ansible-suite-4.2/post-he\_deploy" link towards the left side of the page. The left sidebar contains a navigation menu with options such as "Back to Project", "Status", "Changes", "Console Output", "View as plain text", "Edit Build Information", "Delete Build", "Polling Log", "Retrigger", "Git Build Data", "No Tags", "Parameters", "Git Build Data", "Test Result", "Open Blue Ocean", and "Replay". The top navigation bar includes the Jenkins logo, a search bar, and the user name "Barak Korren".

# STDCI – DEBUGGING ISSUES

How to find out why is it broken



The screenshot shows the Jenkins web interface for a failed build. The browser address bar displays the URL: `jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/artifact/exported-artifacts/check-patch.he-basic-ansible_suite_4.2.el7.x86_64/mock_logs`. The Jenkins header includes the logo, the name "Jenkins", a red notification badge with the number "2", a search bar, and the user name "Barak Korren" with a "log out" link. The breadcrumb trail is: `Jenkins > ovirt-system-tests_standard-check-patch > #581 ovirt-system-tests [check-patch]`. A link to "ENABLE AUTO REFRESH" is visible in the top right. The main content area is titled "Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]". On the left, a sidebar lists various actions: "Back to Project", "Status", "Changes", "Console Output", "View as plain text", "Edit Build Information", "Delete Build", "Polling Log", "Retrigger", "Git Build Data", "No Tags", "Parameters", "Git Build Data", "Test Result", "Open Blue Ocean", and "Replay". The main area shows a file browser view for the path `exported-artifacts / check-patch.he-basic-ansible_suite_4.2.el7.x86_64 / mock_logs /`. Below this path, there are three sub-directories: `init`, `populate_mock`, and `script`, and a file `(all files in zip)`. A green arrow icon is positioned to the right of the file browser path.

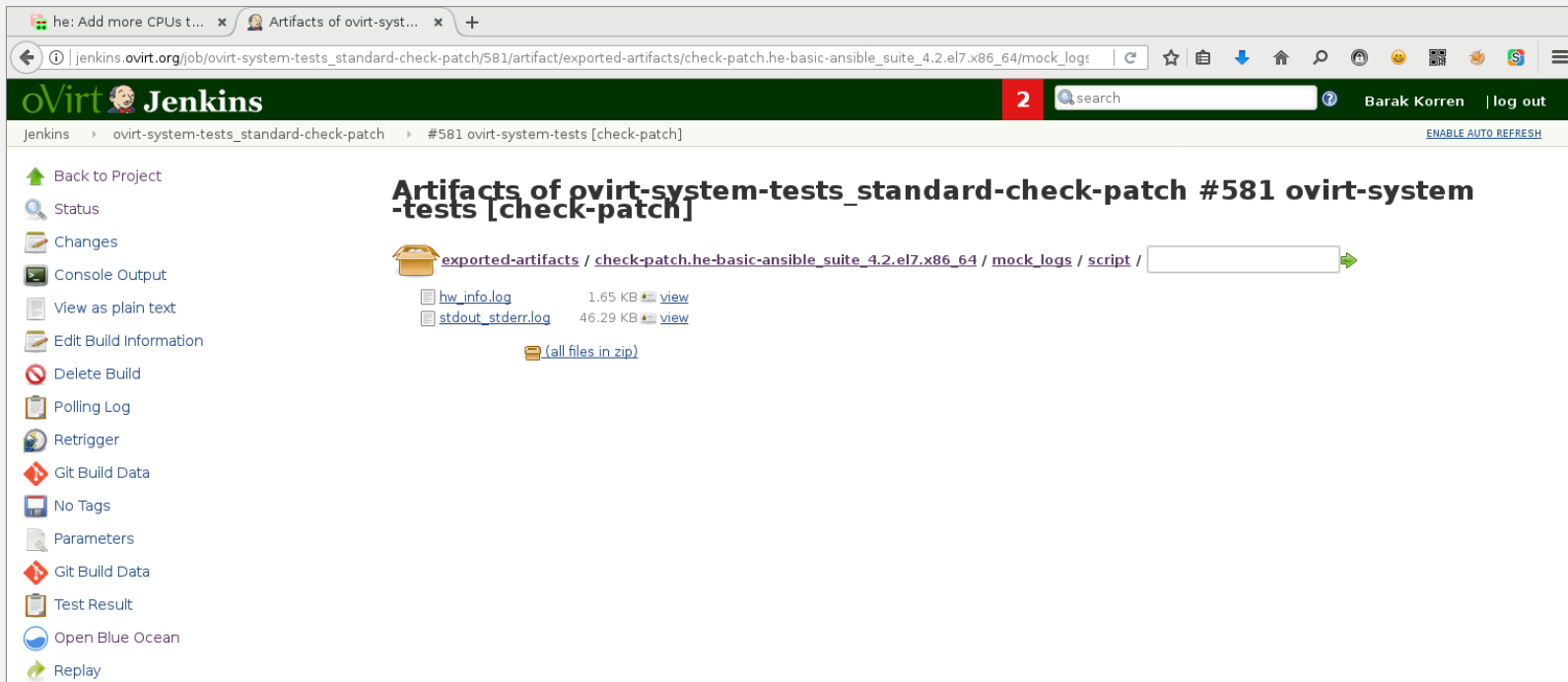
# STDCI – DEBUGGING ISSUES

How to find out why is it broken

The screenshot shows the Jenkins web interface for a failed build. The browser address bar indicates the URL: `jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/artifact/exported-artifacts/check-patch.he-basic-ansible_suite_4.2.el7.x86_64/mock_logs`. The Jenkins header shows the user is logged in as Barak Korren. The main content area displays the title "Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]". Below the title, there is a breadcrumb path: `exported-artifacts / check-patch.he-basic-ansible_suite_4.2.el7.x86_64 / mock_logs /`. A large white arrow points to the `mock_logs` directory in the breadcrumb path. Below the breadcrumb, there are links for `init`, `populate_mock`, `script`, and `(all files in zip)`. The left sidebar contains various navigation options such as "Back to Project", "Status", "Changes", "Console Output", "View as plain text", "Edit Build Information", "Delete Build", "Polling Log", "Retrigger", "Git Build Data", "No Tags", "Parameters", "Test Result", "Open Blue Ocean", and "Replay".

# STDCI – DEBUGGING ISSUES

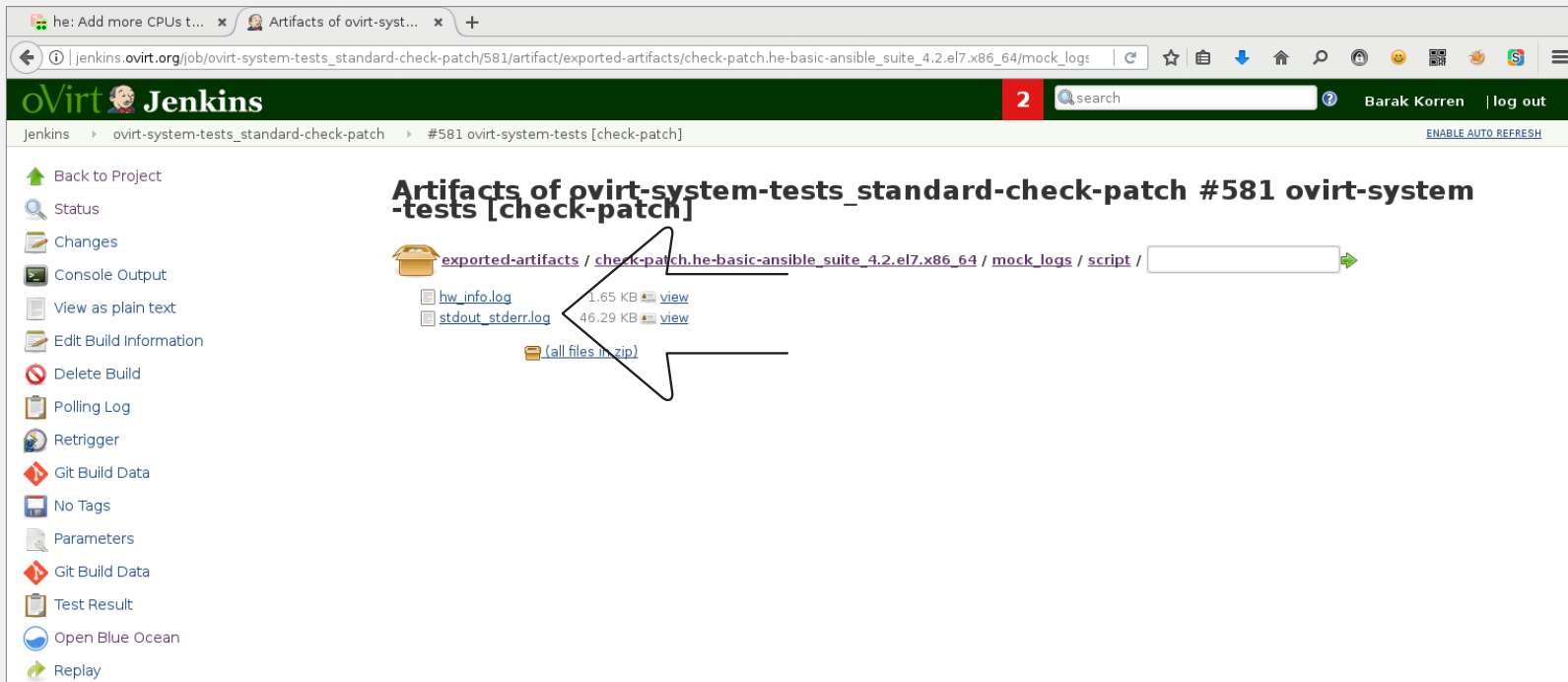
How to find out why is it broken



The screenshot shows a Jenkins web interface. The browser address bar displays the URL: `jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/artifact/exported-artifacts/check-patch.he-basic-ansible_suite_4.2.el7.x86_64/mock_logs`. The Jenkins header includes the logo, the name "Jenkins", a red build status indicator with the number "2", a search bar, and the user name "Barak Korren" with a "log out" link. The breadcrumb trail is: `Jenkins > ovirt-system-tests_standard-check-patch > #581 ovirt-system-tests [check-patch]`. A blue link "ENABLE AUTO REFRESH" is visible in the top right. The main content area is titled "Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]". On the left, a sidebar lists various actions: "Back to Project", "Status", "Changes", "Console Output", "View as plain text", "Edit Build Information", "Delete Build", "Polling Log", "Retrigger", "Git Build Data", "No Tags", "Parameters", "Git Build Data", "Test Result", "Open Blue Ocean", and "Replay". The main area shows a folder icon for "exported-artifacts / check-patch.he-basic-ansible\_suite\_4.2.el7.x86\_64 / mock\_logs / script /" with a search input field. Below this, two files are listed: "hw\_info.log" (1.65 KB) and "stdout\_stderr.log" (46.29 KB), each with a "view" link. At the bottom, there is a "(all files in zip)" link.

# STDCI – DEBUGGING ISSUES

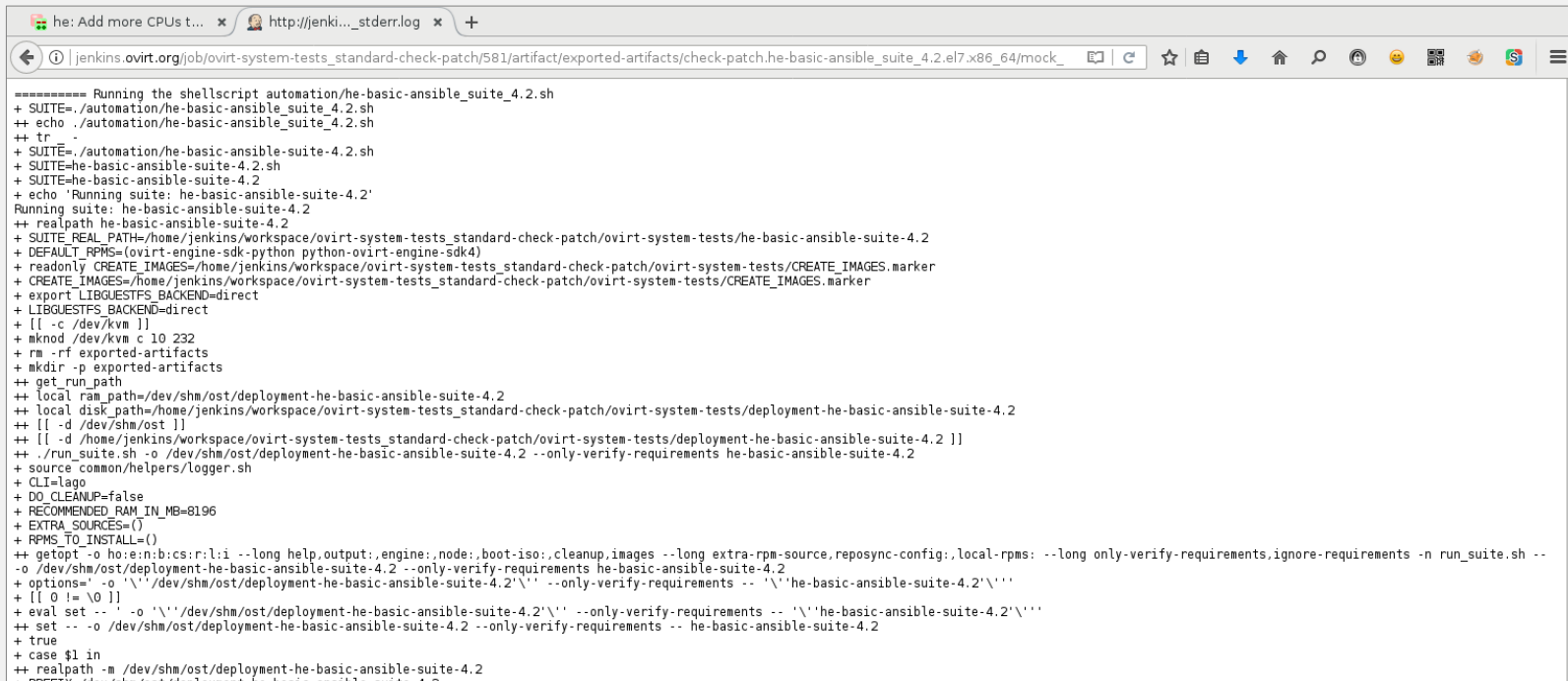
How to find out why is it broken



The screenshot shows the Jenkins web interface for a failed build. The browser address bar indicates the URL: `jenkins.ovirt.org/job/ovirt-system-tests_standard-check-patch/581/artifact/exported-artifacts/check-patch.he-basic-ansible_suite_4.2.el7.x86_64/mock_logs`. The Jenkins header shows the job name: `ovirt-system-tests_standard-check-patch #581 ovirt-system-tests [check-patch]`. The main content area displays the artifacts for the failed build, with the title **Artifacts of ovirt-system-tests\_standard-check-patch #581 ovirt-system-tests [check-patch]**. A breadcrumb path is shown: `exported-artifacts / check-patch.he-basic-ansible_suite_4.2.el7.x86_64 / mock_logs / script /`. Below the breadcrumb, two log files are listed: `hw_info.log` (1.65 KB) and `stdout_stderr.log` (46.29 KB), both with `view` links. A third artifact, `(all files in zip)`, is also listed. A large white arrow points from the `stdout_stderr.log` file to the `hw_info.log` file, indicating a relationship or a specific point of interest. The left sidebar contains various navigation options such as `Back to Project`, `Status`, `Changes`, `Console Output`, `View as plain text`, `Edit Build Information`, `Delete Build`, `Polling Log`, `Retrigger`, `Git Build Data`, `No Tags`, `Parameters`, `Git Build Data`, `Test Result`, `Open Blue Ocean`, and `Replay`.

# STDCI – DEBUGGING ISSUES

## How to find out why is it broken



```
===== Running the shellscript automation/he-basic-ansible_suite_4.2.sh
+ SUITE=/automation/he-basic-ansible_suite_4.2.sh
++ echo ./automation/he-basic-ansible_suite_4.2.sh
++ tr .
+ SUITE=/automation/he-basic-ansible-suite-4.2.sh
+ SUITE=he-basic-ansible-suite-4.2.sh
+ SUITE=he-basic-ansible-suite-4.2
+ echo 'Running suite: he-basic-ansible-suite-4.2'
Running suite: he-basic-ansible-suite-4.2
++ realpath he-basic-ansible-suite-4.2
+ SUITE_REAL_PATH=/home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/he-basic-ansible-suite-4.2
+ DEFAULT_RPMS=ovirt-engine-sdk-python python-ovirt-engine-sdk4
+ readonly CREATE_IMAGES=/home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/CREATE_IMAGES.marker
+ CREATE_IMAGES=/home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/CREATE_IMAGES.marker
+ export LIBQUESTFS_BACKEND=direct
+ LIBQUESTFS_BACKEND=direct
+ [[ -c /dev/kvm ]]
+ mkmod /dev/kvm c 10 232
+ rm -rf exported-artifacts
+ mkdir -p exported-artifacts
++ get_run_path
++ local ram_path=/dev/shm/ost/deployment-he-basic-ansible-suite-4.2
++ local disk_path=/home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/deployment-he-basic-ansible-suite-4.2
++ [[ -d /dev/shm/ost ]]
++ [[ -d /home/jenkins/workspace/ovirt-system-tests_standard-check-patch/ovirt-system-tests/deployment-he-basic-ansible-suite-4.2 ]]
++ ./run_suite.sh -o /dev/shm/ost/deployment-he-basic-ansible-suite-4.2 --only-verify-requirements he-basic-ansible-suite-4.2
+ source common/helpers/logger.sh
+ CLI=lago
+ DO_CLEANUP=false
+ RECOMMENDED_RAM_IN_MB=8196
+ EXTRA_SOURCES=()
+ RPMS_TO_INSTALL=()
++ getopt -o h:en:bc:s:r:l:i --long help,output:,engine:,node:,boot-iso:,cleanup,images --long extra-rpm-source,reposync-config:,local-rpms: --long only-verify-requirements,ignore-requirements -n run_suite.sh --
-o /dev/shm/ost/deployment-he-basic-ansible-suite-4.2 --only-verify-requirements he-basic-ansible-suite-4.2
+ options=' -o '\''/dev/shm/ost/deployment-he-basic-ansible-suite-4.2'\'' --only-verify-requirements -- '\''he-basic-ansible-suite-4.2'\'' '
+ [[ 0 != \0 ]]
+ eval set -- -o '\''/dev/shm/ost/deployment-he-basic-ansible-suite-4.2'\'' --only-verify-requirements -- '\''he-basic-ansible-suite-4.2'\''
++ set -- -o /dev/shm/ost/deployment-he-basic-ansible-suite-4.2 --only-verify-requirements -- he-basic-ansible-suite-4.2
+ true
+ case $1 in
++ realpath -m /dev/shm/ost/deployment-he-basic-ansible-suite-4.2
+ PREFIX=/dev/shm/ost/deployment-he-basic-ansible-suite-4.2
```



# MIGRATING TO STDCI V2

Getting all the new goodies

## ASK US TO DO IT:

Just email [infra-support@ovirt.org](mailto: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/](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