

How to use Jenkins Doc-Site jobs for Deployments

- [Doc server hostnames](#)
- [S3 bucket links](#)
- [Complete list of all Paligo-related Jenkins jobs](#)
- [CloudOps contacts as of 2022-11-21](#)
 - Required setup for all Paligo jobs: manual upload to S3 of generated zipfile of content
 - Required input to start a production and to run Jenkins Jobs
 - Canonical list of all pubsettings used in production
 - With Paligo API, create the content zipfile to be uploaded
 - [Preqrequisites](#)
 - [Calling sequence](#)
 - Upload zipfile to S3 staging bucket /stage directory
 - For new Platform version, push current version down a level
 - New architecture to staging server
 - New architecture to prod server doc.privacera.com
 - Previous Platform (such as 6.5 or 7.2) jobs
 - [Paligo previous Platform version to staging server privacera-docs.privacera.us](#)
 - [Paligo previous Platform version to prod http://doc.privacera.com](#)
- [Linkcheck on staging privacera-docs.privacera.us](#)
 - [Example run](#)
 - [View linkcheck results](#)
 - [Ignore the apparent failure of job](#)
 - [Interpreting linkcheck results: false positives and red herrings](#)

Doc server hostnames

- Staging server: privacera-docs.privacera.us
 - PrivaceraCloud: <https://privacera-docs.privacera.us/cloud/en/index-en.html>
 - Privacera Platform: <https://privacera-docs.privacera.us/platform/6.5/en/index-en.html>
- PrivaceraCloud Operational Test Environment (OTE) doc server: ote-docs.privaceracloud.com
- Public doc server: <https://doc.privacera.com/>

S3 bucket links

- Prod: <https://s3.console.aws.amazon.com/s3/buckets/privacera-doc?region=us-east-1&tab=objects>
- Dev (or "staging"): <https://s3.console.aws.amazon.com/s3/buckets/paligo-dev?region=ap-south-1&prefix=stage/&showversions=false>

Complete list of all Paligo-related Jenkins jobs

What	New Name
New architecture to Staging	http://jupiter.privacera.us:8080/view/doc-site/job/Doc-site/job/paligo-newarch/
New architecture to Production	http://privacera-prod-jenkins.privacera.us:8080/view/Paligo-Prod/job/Paligo_Newarch_Prod/
Previous Platform (such as 6.5 or 7.2) Paligo to Staging	Paligo_Platform_to_Staging
Previous Platform (such as 6.5 or 7.2) Paligo to Production	Paligo_Platform_Staging_to_Prod
Linkchecking of staging	http://jupiter.privacera.us:8080/view/doc-site/job/Doc-site/job/paligo-newarch-linkcheck/
Invalidate staging files	http://jupiter.privacera.us:8080/view/doc-site/job/Doc-site/job/paligo-newarch-only-invalidation/
Invalidate production files	http://privacera-prod-jenkins.privacera.us:8080/view/Paligo-Prod/job/Prod-Paligo-Newarch-Only-invalidation/

CloudOps contacts as of 2022-11-21

- Arif Shaik, night shift operator
- Suraj Sharma, night shift operator, 2nd level
- Pranav Kunderkar, first level

Required setup for all Paligo jobs: manual upload to S3 of generated zipfile of content

As of 20220715, publishing requires:

- Using the Paligo API, we manually generate the content to be published. This produces a zipfile.
- We upload that zipfile to an S3 bucket directory called `/test` provided by CloudOps.

Required input to start a production and to run Jenkins Jobs

The following values are needed to start a production. They are also need for all Jenkins jobs.

- Publication ID
- Pubsettings number

Canonical list of all pubsettings used in production

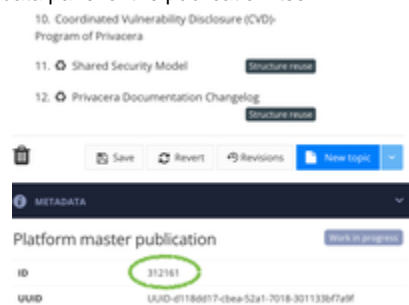
The following link has the running IDs for publications <https://docs.google.com/document/d/1i2LIS3f1689sdgFKY91xt03rb70UAkJ29IbSTvcr6I/edit>

With Paligo API, create the content zipfile to be uploaded

We use three Paligo APIs to generate a zipfile to upload.

Prequisites

- The `<publication_ID>` value is visible in the metadata panel of the publication itself.



- Get all the publish settings:

`/publishsettings`

You must correlate the `<publication_ID>` with its corresponding `<publishsettings_ID`

```

"resource":229973,"created_at":1656604155,"modified_at":1656604155,"reuse":"","shared":
":"","creator":"","format":"html5","user":"13"}, {"id":25,"name":"Platform_65_sett
ings","resource":312161,"created_at":1657888522,"modified_at":1657888522,"reuse"
":"","shared":"","creator":"","format":"html5","user":"3"}, {"id":21,"name":"Publi
cations_14_copy_1","resource":312161,"created_at":1656604155,"modified_at":1656604155

```

Calling sequence

- Start a production:

`/productions/<publication_ID>` and in POST body the `<publishsettings_ID>`

Returns a `<production_ID>`

- Get the `<output_ID>`:

`/productions/<production_ID>`

- Download the zipfile:

/outputs/<output_ID>

Upload zipfile to S3 staging bucket /stage directory

Upload your zipfile to <https://s3.console.aws.amazon.com/s3/buckets/paligo-dev?region=ap-south-1&prefix=stage/&showversions=false>

For new Platform version, push current version down a level

The latest version of all docs (Cloud and Platform) sits at the top of the doc website.

When a new version of Platform is to be published, you must first push the current version down one level. You do this by running the Platform-specific jobs for the version being superceded, as detailed in **Previous Platform (such as 6.5 or 7.2) jobs**.

New architecture to staging server

Job link: <http://jupiter.privacera.us:8080/view/doc-site/job/Doc-site/job/paligo-newarch/>

Required data:

- Publication ID
- Pubsettings ID

New architecture to prod server doc.privacera.com



Before publishing to production, you must first run the staging job shown above.

Job link: http://privacera-prod-jenkins.privacera.us:8080/view/Paligo-Prod/job/Paligo_Newarch_Prod/

Required data:

- Publication ID
- Pubsettings ID

Previous Platform (such as 6.5 or 7.2) jobs

These jobs are used exclusively for updates to previous Platform versions (such as 6.5 or 7.2)

Paligo previous Platform version to staging server privacera-docs.privacera.us

Job link: [Paligo_Platform_to_Staging](#)

Required data:

- Publication ID
- Pubsettings ID
- Target S3 directory name; for example, 6 . 5.

Paligo previous Platform version to prod <http://doc.privacera.com>



Before publishing to production, you must first run the staging job shown above.

Job link: [Paligo_Platform_Staging_to_Prod](#)

Required data:


- Publication ID
- Pubsettings ID
- Target S3 directory name; for example, 6 . 5.

Linkcheck on staging privacera-docs.privacera.us

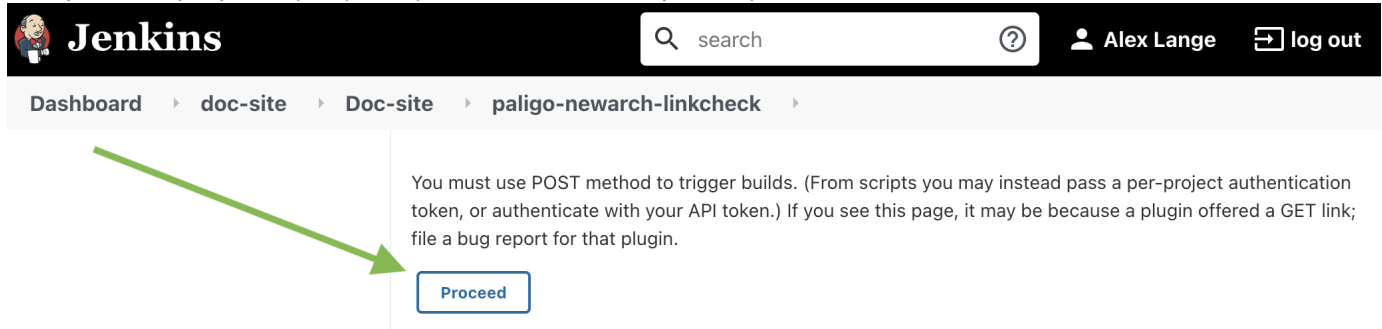
The public domain [linkcheck software](#) is available for check the doc staging server.

- linkcheck is run by Jenkins job <http://jupiter.privacera.us:8080/view/doc-site/job/Doc-site/job/paligo-newarch-linkcheck/>
- This is not a scheduled job and is run only on demand by Docs team. It alters nothing in either dev or prod and merely scans links on the staging server.

Example run

 This annotated real example of the job was successful and found no actual links of any concern.

When you run the job, you are prompted to proceed. There are not any other input values:

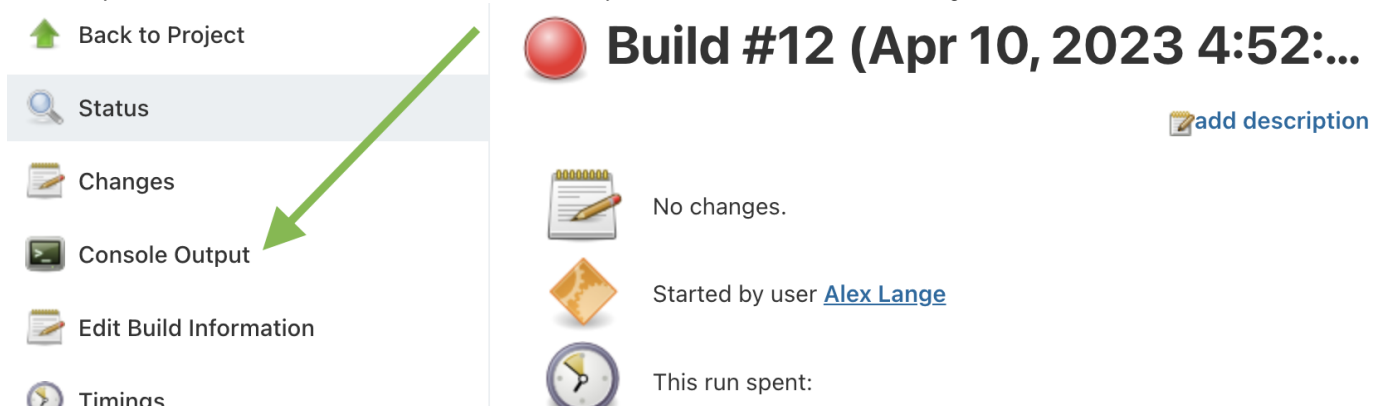


When the job runs, there is no visible output to your screen.

View linkcheck results

To see the results:

- Look at the console output for that job in Jenkins itself. Or
- Check your email for a link to the latest run. This link directs you back to Jenkins to look at that log.



Ignore the apparent failure of job

Jenkins marks any job it sees as failed as **failed-red**.

When linkcheck finds what it thinks is a bad link, it sets its own exit status to the Linux **ERROR** signal (some non-zero integer).

This means that Jenkins always receives that signal and thus marks the job as **failed-red**.

```
4 warnings
5 errors
```

```
Build step 'Execute shell' marked build as failure
Email was triggered for: Always
Sending email for trigger: Always
Sending Email to: Alex.lange@privacera.com pranav.kunkerkar@privacera.com
Finished: FAILURE
```

Because linkcheck's purpose is to find such bad links, it's "failing" indicates that it worked.

Interpreting linkcheck results: false positives and red herrings

linkcheck understandably sometimes flags links as broken or malformed. (This is the plague of all link checkers: the deal with false positives in different ways depending on what they are looking for.)

For example, in each of the red herrings below, clicking the actual link in your browser reveals no difficulty.

1. <https://mvnrepository.com/artifact/io.delta/delta-core>
2. https://www.tpc.org/tpc_documents_current_versions/current_specifications5.asp

1

```
https://privacera-docs.privacera.us/en/configure-emr-with-privacera-platform.html
- (66:19558) 'Delta Core' => https://mvnrepository.com/artifact/io.delta/delta-core
(HTTP 403)
```

2

```
https://privacera-docs.privacera.us/en/example--create-basic-policies-for-databricks-
sql-table-access-424608.html
- (64:325988) 'TPC DS' =>
https://www.tpc.org/tpc_documents_current_versions/current_specifications5.asp
(connection failed)
```

```
https://privacera-docs.privacera.us/en/example--create-basic-policies-for-databricks-
sql-table-access.html
- (64:326399) 'TPC DS' =>
https://www.tpc.org/tpc_documents_current_versions/current_specifications5.asp
(connection failed)
```

```
https://privacera-docs.privacera.us/en/example--manage-access-to-databricks-sql-with-
privacera.html
- (64:326090) 'TPC DS' =>
```

1. [403 Forbidden?](#) Some web servers check the user agent of a requesting browser. In the case of linkcheck, it is not a human-operated web browser. The Maven site might be programmed to reject [robots](#).
2. And following lines. Likewise, this could be robot-screening, but "connection failed" indicates possibly something else. These links to the TPC benchmark are valid. The connection might have failed for reason in #1 above or perhaps a web server's refusal of the scanning request by anti-slamming settings. (linkcheck's HTTP requests come very fast, because it is not a slow human.)

In the following false positive, Privacera's Zendesk system requires the user to login, which linkcheck doesn't do:

~~https://www.privacera.com/privacy/privacy-policy/privacy-policy.html~~
(connection failed)

<https://privacera-docs.privacera.us/en/how-to-get-support.html>

- (64:324675) 'Privacera..' => <https://privacera.zendesk.com> (HTTP 301 -> 301 => 403)

- redirect path:

- <https://privacera.zendesk.com> (301)

- <https://privacera.zendesk.com/hc> (301)

- <https://privacera.zendesk.com/hc/en-us> (403)

- (64:324970) 'Zendesk' => <https://privacera.zendesk.com> (HTTP 301 -> 301 => 403)

- redirect path:

- <https://privacera.zendesk.com> (301)

- <https://privacera.zendesk.com/hc> (301)

- <https://privacera.zendesk.com/hc/en-us> (403)

- (64:326363) 'Privacera..' => <https://privacera.zendesk.com/hc/en-us> (HTTP 403)

<https://privacera-docs.privacera.us/en/privaceracloud-release-notes.html>

- (64:325273) 'Privacera..' => <https://privacera.zendesk.com/hc/en-us/articles/4415267760151-Privacera-Product-Support-Policy> (HTTP 403)