MIGRATION PLAYBOOK · BITBUCKET

How to migrate Bitbucket Server or Data Center to Bitbucket Cloud

Move Git repositories and pull requests from Bitbucket Server or Data Center to Bitbucket Cloud with Atlassian's free migration assistant, then rebuild what it leaves behind: Git LFS files, webhooks, default reviewers, merge checks and CI.

Checked against vendor documentation – 20 November 2026 5 phases
15 detailed steps 9 official sources

START HERE

How do you migrate Bitbucket Server or Data Center to Bitbucket Cloud?

SHORT ANSWER

This moves Git repositories and pull requests from Bitbucket Server or Data Center to Bitbucket Cloud. The migration assistant copies repositories in waves while each wave is frozen, then you finish what it does not move: large files stored in Git LFS, webhooks, default reviewers, merge checks, apps and builds.

- **Recommended route:** Bitbucket Cloud Migration Assistant
- **Order:** Decide the destination and workspace first, prove the assistant on a test migration, then move repositories in waves, finishing each wave's LFS, webhooks, reviewers and builds before developers switch.
- **5 phases:** Decide and plan, Install and test, Migrate production, Rebuild settings and integrations and Switch and retire the server.
- **You're done when:** Branch, tag and commit counts match, and the default branch head is the same commit on both sides.

KEY DATES

30 Mar 2026	Data Center subscriptions no longer sold to new customers.	Passed

30 Mar 2028	Last date for existing customers to buy new Data Center licences and expansions.	Upcoming
28 Mar 2029	Data Center licences expire and become read-only, unless you move to the Bitbucket Hybrid License.	Upcoming

A test migration shows how long each repository takes, and reconnecting CI and Jira is its own piece of work.

RECOMMENDED ROUTE

Bitbucket Cloud Migration Assistant

Free, built and maintained by Atlassian, and the method Atlassian recommends for repositories and pull requests.

▼ **Other options and when to use them**

- **Bitbucket Hybrid License (stay on Data Center for now)**
  : You are an existing Bitbucket Data Center or Server customer and need to keep self-managed repositories past March 2029.
- **GitHub Enterprise Importer**
  : Your destination is GitHub Enterprise Cloud rather than Bitbucket Cloud. It moves repositories and pull requests; CI pipelines stay in Bitbucket.
- **git push –mirror (code only)**
  : A handful of repositories where pull request history is not needed.

TOOLS FOR THIS PLAN

Are you ready?
22-question readiness check

Plan your dates
Dated countdown you can add to your calendar

New to the jargon?
Plain-English migration glossary

ORDER TO MIGRATE IN, AND WHY

Decide the destination and workspace first, prove the assistant on a test migration, then move repositories in waves, finishing each wave's LFS, webhooks, reviewers and builds before developers switch. The server is made read-only last.

1 **Destination decision: Bitbucket Cloud, Hybrid License or another platform**
Decides the tool, the timeline and whether CI has to move at all.

2 **Inventory: repository sizes, LFS use, forks, personal repositories, webhooks, builds and apps**

Each of these is handled outside the assistant, and oversized repositories fail its preflight checks.

3 **Cloud workspace on Standard or Premium, with users and email addresses cleaned up**

The assistant refuses a Free plan and the personal default workspace, and users are matched by valid, unique email.

4 **Test migration of a few representative repositories**

Shows timings, flattened pull request comments and preflight warnings before any freeze.

5 **Production waves of repositories that build together**

A stopped migration has to start again, so waves are sized to finish inside a freeze window.

6 **Per wave: LFS files, default reviewers, branch permissions, merge checks, webhooks**

Without these, builds fail and the default branch has no protection on day one.

7 **CI and Jira reconnected, then developers switch remotes**

Builds and Jira links point at the server until you change them.

8 **Server read-only, then retired**

Keeps old links working and gives you a fallback until every build runs from Cloud.

SETTINGS CHECKLIST: DECIDE THESE BEFORE YOU COPY

Built-in and third-party tools treat these differently. Tick each one off once you have decided.

Download this checklist (Excel)

☐ **Where Bitbucket goes after March 2029**

Whether you move to Bitbucket Cloud, keep self-managed Bitbucket under the Hybrid License, or move to another platform such as GitHub.

▼ How each tool handles it

Built-in method	Atlassian: Data Center licences expire and become read-only on 28 March 2029. Existing Bitbucket and Bamboo Data Center customers can buy the Bitbucket Hybrid License, which covers self-managed Data Center and Bitbucket Cloud with Pipelines, available from early 2027.

Tip: Make this decision before any inventory work: it changes the tool, the CI workstream and the deadline. If GitHub is the destination, GitHub Enterprise Importer moves repositories and pull requests from Bitbucket Server to GitHub Enterprise Cloud.

Atlassian: Bitbucket Hybrid License

- [ ] **Versions, plan and workspace**

Which Cloud workspace receives the repositories, and whether the source needs an upgrade first.

▼ How each tool handles it

Built-in method	Assistant: Bitbucket Server or Data Center 7.0 or later, and a Bitbucket Cloud workspace on the Standard or Premium plan. Atlassian says to use a workspace you created, not the default workspace named after your user account, which has limited functionality.

Tip: Create the workspace early and name it for the organisation, not a person.

Atlassian Support: Use the Bitbucket Cloud Migration Assistant

- [ ] **Repository size**

Which repositories need history clean-up or their own wave.

▼ How each tool handles it

Built-in method	Assistant: preflight checks block repositories over the Cloud size limit. Atlassian's pages give different figures: the migration methods page says 4 GB, and the what-gets-migrated page says the maximum is currently 10 GB. Treat the preflight result as the deciding check.

Tip: Run a test migration with your largest repositories first, and read the size warnings in the preflight checks.

Atlassian Support: What gets migrated with the Bitbucket Cloud Migration Assistant

- [ ] **Git LFS files**

Which repositories use Git LFS (large files kept outside Git), and who pushes them after each wave.

▼ How each tool handles it

Built-in method	Assistant: LFS content is not migrated. Atlassian documents a separate mirror-clone and git lfs push procedure after the repository is migrated.

Tip: Find LFS repositories by looking for filter=lfs in .gitattributes, and add the push to each wave's runsheet.

Atlassian Support: Migrating repositories containing LFS to Bitbucket Cloud

- [ ] **Pull requests, comments and tasks**

Which open pull requests to merge or decline before the freeze, and whether flattened comments are acceptable.

▼ **How each tool handles it**

Built-in method	Assistant: pull request titles, descriptions, status, diffs, comments, tasks, authors and reviewers migrate. Atlassian says comments and tasks are flattened and may not display as they did on Server. Pull request attachments and the activity feed do not migrate.

Tip: In the test migration, ask owners to compare a few busy pull requests on both sides.

Atlassian Support: What gets migrated with the Bitbucket Cloud Migration Assistant

- [ ] **Forks, personal repositories and duplicate names**

What happens to forks and to repositories in personal projects.

▼ **How each tool handles it**

Built-in method	Assistant: forks arrive as normal repositories with the fork link lost, so Atlassian recommends merging open pull requests from forks first. Personal repositories and repositories with duplicate names are not migrated; Atlassian suggests moving personal repositories into normal projects first.

Tip: Move personal repositories that matter into a project, and rename duplicates, before the test migration.

Atlassian Support: What gets migrated with the Bitbucket Cloud Migration Assistant

- [ ] **Users, groups and permissions**

Which users to bring, and whether groups and permissions are migrated or rebuilt.

▼ **How each tool handles it**

Built-in method	Assistant: you choose all users, users with access to or involved in pull requests, or only those involved in pull requests. Groups, permissions, branch permissions and branching models can be migrated only into workspaces administered through Atlassian Administration. Atlassian's

methods and after-migration pages still say permissions need recreating, so check the options your workspace shows. Inactive users arrive with no permissions, and avatars, passwords, SSH keys and app passwords never migrate.

Tip: Export the current permission list from the server before the first wave, so you can rebuild or check it the same day.

Atlassian Support: Use the Bitbucket Cloud Migration Assistant

- [ ] **Repository settings and integrations**

Which webhooks, default reviewers, merge checks and apps each repository needs on day one.

▼ **How each tool handles it**

Built-in method Assistant: default reviewers, webhooks, merge checks, build statuses, project settings and Marketplace apps with their data are not migrated. Atlassian lists each as a manual step after migration.

Tip: Export each repository's webhooks and merge checks from the server, and recreate them straight after each wave.

Atlassian Support: After migrating to Bitbucket Cloud

- [ ] **Builds and deployments**

Where builds run after the move: Bitbucket Pipelines, an existing Jenkins or Bamboo server, or another CI service.

▼ **How each tool handles it**

Built-in method Assistant: Atlassian's migration pages do not cover CI, and build statuses are not migrated. Any build server that clones from the Bitbucket server keeps doing so until you change it.

Tip: List every build job, deploy key and script that clones from the server, and give each repository a CI owner in the wave plan.

Atlassian Support: After migrating to Bitbucket Cloud

- [ ] **Freeze and reruns**

How long each wave is frozen, and how the server is locked during it.

▼ **How each tool handles it**

Built-in method Assistant: Atlassian asks people to avoid changing anything during the transition and suggests locking down

the server. Stopping a migration means starting again, and already-migrated repositories should be deleted first to avoid Invalid repository errors. Run only one migration at a time.

Tip: Size each wave to finish well inside its window, and keep a list of what has already landed in Cloud.

Atlassian Support: Bitbucket pre-migration checklist

THE PLAN, PHASE BY PHASE

1 Decide and plan

Before you start

- Decide between Bitbucket Cloud, the Bitbucket Hybrid License and another platform.
- List every repository with its size, LFS use, forks, webhooks, builds and owner.
- Create the Cloud workspace on

Standard or Premium, and fix blank or duplicate user email addresses.

Group repositories that build together into waves, each with a freeze window.

> **Watch out:** Atlassian's pages give different repository size limits (4 GB on the methods page, 10 GB on the what-gets-migrated page).
>
> **WAYS AROUND IT**
>
> - Use the preflight check result for each repository, and test the largest ones first.

Show the detailed steps (3) →

2 Install and test

Install the Bitbucket Cloud Migration Assistant from

the Atlassian Marketplace on the server.

Allow the server to reach atlassian.com if it sits behind a firewall.

Connect to the Cloud workspace and migrate a few representative repositories.

Compare branches, tags and pull requests with the owners, and

record timings.

> **Watch out:** Pull request comments and tasks are flattened and may not display as they did on Server.
>
> **WAYS AROUND IT**
>
> - Show owners examples in the test site and agree it is acceptable before production.

Show the detailed steps (3) →

3 Migrate production

In each wave's freeze window

Merge or decline open pull requests, especially from forks.

Lock the wave's repositories, then create the migration, choose repositories and users, and run preflight checks.

Fix red errors,

review warnings, then run the migration and watch the details screen.

Push LFS files for repositories that use them.

> **Watch out:** Stopping a migration means starting again, and only one migration should run at a time.
>
> **WAYS AROUND IT**
>
> - Delete repositories that already landed before rerunning, unless they have not changed.

Show the detailed steps (3) →

4 Rebuild settings and integrations

Recreate default reviewers, branch permissions and merge checks.

Recreate webhooks and

install the Cloud versions of your apps.

Connect Bitbucket Cloud to Jira.

Point builds and deployments at the Cloud repositories.

> **Watch out:** Webhooks, default reviewers, merge checks, build statuses and Marketplace apps are not migrated.
>
> **WAYS AROUND IT**
>
> - Export them per repository from the server before the wave, and recreate them before developers switch.

Show the detailed steps (3) →

5 Switch and retire the server

Ask developers to change their remote with git remote set-

url, and to add SSH keys in Cloud.

Keep the server read-only for an agreed period so old links still work.

Export what your records policy requires, then decommission the server.

> **Watch out:** Uninstalling or deleting the server cannot be undone, and anything the assistant did not move goes with it.
>
> **WAYS AROUND IT**
>
> - Keep the server read-only until every build runs from Cloud and the decommission checklist is signed off.

Turn this phase into dated tasks with the cutover planner

Show the detailed steps (2) →

✓ You are finished when

- Branch, tag and commit counts match, and the default branch head is the same commit on both sides.
- Pull request counts match, and LFS files download with their content.
- Each repository has its default reviewers, branch permissions, merge checks and webhooks.
- Every build runs from Bitbucket Cloud, and nothing has been pushed to the server since the freeze.

CLICK BY CLICK

What does each step involve?

Open a part when you reach it in the plan. Each step says where to go, what to do and what you should see.

About this migration

What it covers, limits at a glance —

Atlassian's Bitbucket Cloud Migration Assistant copies repositories and pull requests from Bitbucket Server or Data Center into a Bitbucket Cloud workspace. It does not move Git LFS files, webhooks, default reviewers, merge checks, apps or build configuration, so a Bitbucket move is a source-control project with a CI and integrations workstream, not only a copy.

Atlassian has set 28 March 2029 as the end of life for its Data Center products, when licences expire and become read-only. For Bitbucket there is a third option: the Bitbucket Hybrid License, which Atlassian says will cover both Bitbucket Data Center and Bitbucket Cloud from early 2027. If you are moving Jira and Confluence too, run this alongside the Jira and Confluence Server to Cloud playbook. If GitHub is the destination, see the GitLab to GitHub and Azure DevOps to GitHub playbooks, which use the same importer family.

NATIVE TOOL	SOURCE	DESTINATION
Bitbucket Cloud Migration Assistant,	**Bitbucket Server or Data Center 7.0 or later**	**A Bitbucket Cloud workspace on the**

free from the Atlassian Marketplace

Standard or Premium plan

MIGRATED

Repositories (branches, tags, commits), pull requests with comments and tasks, and the users you choose

NOT MIGRATED

Git LFS files, forks as forks, webhooks, default reviewers, merge checks, Marketplace apps, SSH keys

DEADLINE

Atlassian Data Center end of life on 28 March 2029, unless you move to the Bitbucket Hybrid License

1 What do you need before you migrate Bitbucket Server or Data Center to Bitbucket Cloud?

Need	Details	Where to get it
Bitbucket Server or Data Center 7.0 or later	Older versions must be upgraded before the assistant can run.	Bitbucket administration > About
Bitbucket Cloud workspace	Standard or Premium plan, created for the organisation, not the default personal workspace.	bitbucket.org; check current plans with Atlassian
Admin access	System administrator on the server, and admin of the Cloud workspace.	Your Atlassian administrators
Network	The server must reach atlassian.com if it is behind a firewall.	Your network team
Git LFS client	Needed on the migration machine for repositories that use LFS.	https://git-lfs.com
Inventory	Every repository with size, LFS use, forks, webhooks, builds and owner.	Application inventory and dependency map template

2 Decide the destination and plan the waves

3 steps

Bitbucket item	Moves with the assistant?	What to do
Branches, tags, commits	Yes	Check size against the preflight result
Pull requests with comments, tasks and reviewers	Yes (comments and tasks flattened)	Merge or decline stale pull requests first
Users	Yes, by the option you choose	Fix email addresses first
Groups, permissions, branch permissions	Only into Atlassian-administered workspaces	Otherwise recreate after each wave
Git LFS files	No	Push separately (Step 4)
Forks	As normal repositories, link lost	Merge open pull requests from forks first
Personal repositories, duplicate names	No	Move into projects or rename first
Default reviewers, webhooks, merge checks, build statuses	No	Recreate (Step 5)
Marketplace apps and their data	No	Install Cloud versions where they exist
Avatars, passwords, SSH keys, app passwords	No	Users set them up again in Cloud

1

Record the destination decision

Where: Your project brief, signed off by engineering leads

Write down whether you are moving to Bitbucket Cloud, staying self-managed under the Bitbucket Hybrid License, or moving to another platform, and why.

✓ **You should see:** A signed-off decision and a target date before 28 March 2029.

2

Inventory the repositories

Where: Bitbucket administration on the server, and your Application inventory and dependency map

List every project and repository with its size, owner, whether it uses Git LFS, its forks, webhooks, merge checks, default reviewers and the build jobs that clone it. Mark personal repositories and duplicate names.

✓ **You should see:** One row per repository with these flags, an owner and a wave.

3

Set the waves and freeze windows

Where: Your Migration wave plan

Group repositories that build together into the same wave, give each wave a test date and a production freeze window, and name a CI owner for each repository.

✓ **You should see:** The wave plan is signed off by each team lead.

3 Install the assistant and run a test migration

3 steps

1

Install the migration assistant

Where: Bitbucket Data Center or Server administration > Manage apps, with the app from the Atlassian Marketplace

Install the Bitbucket Cloud Migration Assistant. If the server is behind a firewall, allow it to reach atlassian.com.

✓ **You should see:** Administration shows a Migrate to cloud item.

2

Connect to the Cloud workspace

Where: Bitbucket Data Center or Server: Administration > Migrate to cloud (the migration assistant) > Manage your migration > Create a new migration > Connect to cloud

Name the migration plan and select **I already have a workspace**. Copy the Base URL from **Administration** > **Server settings** into **Migration from**, choose the workspace under **Migrate to**, tick **Allow Atlassian to access migration data** and select **Confirm**.

✓ **You should see:** The plan shows your server as the source and your workspace as the destination.

3

Migrate a few representative repositories

Where: Bitbucket Data Center or Server: Administration > Migrate to cloud (the migration assistant)

Choose a small, a large and a busy repository, select **Add to migration**, pick the user, group and permission options, then run the preflight checks and the migration. Compare the results with each owner and record how long it took.

✓ **You should see:** The test repositories appear in the workspace, and each owner signs off the comparison in your Pilot test plan.

4 Migrate production repositories

3 steps

1

Freeze the wave

Where: Bitbucket repository and project permissions on the server

Merge or decline open pull requests, especially from forks. At the start of the window, remove write access for the wave's repositories and tell developers the freeze has started.

✓ **You should see:** Pushes to the wave's repositories are refused.

2

Create the migration and run preflight checks

Where: Bitbucket Data Center or Server: Administration > Migrate to cloud (the migration assistant) > Create a new migration

Select the wave's repositories, choose users, groups, permissions and branch permissions, and run the preflight checks. A red error blocks the migration; a warning lets you continue. Fix errors, then use **Refresh all**.

✓ **You should see:** Every check is green, or shows only warnings you have accepted.

3

Run and monitor

Where: Bitbucket Data Center or Server: Administration > Migrate to cloud (the migration assistant) > Review migration > Run

Select **Run** and watch the details screen. Run one migration at a time. If it finishes with errors, check the workspace for what arrived; if it fails, create a new migration that avoids the cause.

✓ **You should see:** The migration completes and Go to cloud workspace shows the wave's repositories.

5 Push Git LFS files

1 step

1

Push LFS objects to each Cloud repository

Where: Terminal on your migration machine with Git LFS installed and SSH access to the server

For each repository that uses LFS, mirror-clone it from the server, fetch every LFS object, point LFS at the Cloud repository and push. This follows Atlassian's LFS migration article.

```
git clone --mirror ssh://git@bitbucket.contoso.com:7999/PAY/payments-api.git
cd payments-api.git
git lfs install --local
git lfs fetch --all
git config lfs.url https://bitbucket.org/contoso/payments-api.git/info/lfs
git remote set-url origin https://bitbucket.org/contoso/payments-api.git
git lfs push --all https://bitbucket.org/contoso/payments-api.git
```

✓ **You should see:** A fresh clone from Bitbucket Cloud downloads LFS files with their content, not pointer files.

6 Rebuild settings, integrations and builds

3 steps

1

Recreate reviewers, permissions and merge checks

Where: Bitbucket Cloud > each repository > Repository settings

Add default reviewers, recreate branch permissions if they were not migrated, set merge checks, and give groups their repository access.

✓ **You should see:** The default branch rejects direct pushes from developers, and pull requests show the right reviewers.

2

Recreate webhooks and apps, and connect Jira

Where: Bitbucket Cloud > Repository settings > Webhooks, workspace settings for apps, and your Jira site

Recreate each webhook from your export, install Cloud versions of the apps you need, and connect Bitbucket Cloud to Jira using the guide for your Jira deployment.

✓ **You should see:** Each webhook shows a successful test delivery, and Jira work items show their branches and commits.

3

Point builds and deployments at Cloud

Where: Your CI server (Jenkins, Bamboo or other) or Bitbucket Pipelines

Change each build job and deployment script to clone from `bitbucket.org`, with new access keys or tokens. Keep the server read-only until every build is green.

✓ **You should see:** Every repository builds from Bitbucket Cloud, and no job still clones from the server.

7 Switch developers and retire the server

2 steps −

1

Point local clones at Bitbucket Cloud

Where: Each developer's terminal, inside their existing clone

Developers add their SSH key in Bitbucket Cloud, change the remote URL and fetch to confirm. SSH keys and app passwords do not migrate.

```
git remote set-url origin git@bitbucket.org:contoso/payments-api.git
git fetch origin
git remote -v
```

✓ **You should see:** git remote -v shows bitbucket.org and git fetch succeeds.

2

Keep the server read-only, then retire it

Where: Bitbucket administration on the server, and your Decommission checklist

Leave the server read-only for an agreed period so links in tickets and documents still resolve. Export what your records policy requires, then decommission it.

✓ **You should see:** The decommission checklist is signed off and nothing still clones from the server.

WORTH KNOWING

What can go wrong, and how do you avoid it?

The detail behind the plan: options compared, and what experienced engineers check first. Tap any item.

Compare all the ways to do this 3 options –

Bitbucket Cloud Migration Assistant

MEDIUM EFFORT

Best for: Moving to Bitbucket Cloud with pull request history

+ Free and maintained by Atlassian
+ Moves repositories, pull requests and chosen users
+ Preflight checks catch blockers before the run
– No Git LFS files, webhooks, default reviewers, merge checks or apps
– A stopped migration starts again
– One migration at a time

Read more →

Bitbucket Hybrid License

LOW EFFORT

Best for: Existing Bitbucket Data Center customers who need to stay self-managed past March 2029

+ Keeps self-managed repositories
+ Adds Bitbucket Cloud with Pipelines under one subscription
– Only for existing Bitbucket and Bamboo Data Center or Server customers
– Available from early 2027

Read more →

GitHub Enterprise Importer

MEDIUM EFFORT

Best for: Organisations standardising on GitHub Enterprise Cloud

+ Moves Git repositories and pull requests
+ Same importer family as GitLab and Azure DevOps moves
– CI pipelines stay in Bitbucket
– Permissions rebuilt as teams; user activity attributed to placeholders until reclaimed

Read more →

OUR ADVICE

Use the migration assistant when Bitbucket Cloud is the destination and pull request history matters. Look at the Hybrid License first if your teams need self-

managed repositories after March 2029. If the organisation is standardising on GitHub, use GitHub Enterprise Importer instead. Whichever you choose, plan Git LFS, integrations and CI as their own workstream.

Decide the destination before anything else

COST LEVER

Atlassian has set Data Center end of life for 28 March 2029, when licences expire and become read-only. For Bitbucket, it also offers a Hybrid License to existing Bitbucket and Bamboo Data Center customers that covers both self-managed and Cloud, from early 2027.

WHAT TO DO

Choose between Bitbucket Cloud, the Hybrid License and another platform with your engineering leads, and record it in the plan. The rest of this playbook assumes Bitbucket Cloud.

Atlassian: Bitbucket Hybrid License

Push Git LFS files separately

REDUCES RISK

Atlassian says the assistant migrates repository source, branches, tags, pull requests and users, but not LFS files. A repository that looks complete fails its first build when they are missing.

WHAT TO DO

List repositories with `filter=lfs` in `.gitattributes`, and run the LFS push in Step 4 straight after each one migrates.

Atlassian Support: Migrating repositories containing LFS

Freeze each wave; a stopped migration starts again

REDUCES RISK

Atlassian asks people not to change anything during the transition. If you stop a migration, you start over, and repositories that already landed should be deleted first.

WHAT TO DO

Agree a freeze window per wave, lock the repositories at the start, and run one migration at a time.

Atlassian Support: Use the Bitbucket Cloud Migration Assistant

Plan CI as its own workstream

SAVES TIME

Atlassian's migration pages do not cover CI, and build statuses are not migrated. Every Jenkins, Bamboo or deployment job that clones from the server keeps pointing there.

WHAT TO DO

List build jobs, deploy keys and scripts per repository, decide where each will run, and keep the server read-only, not deleted, until every build runs from Cloud.

Atlassian Support: After migrating to Bitbucket Cloud

Recreate repository settings before developers switch

COMPLIANCE

Default reviewers, branch permissions (unless migrated), webhooks and merge checks are manual steps after migration. Without them the default branch has no protection on day one.

WHAT TO DO

Export each repository's settings from the server before its wave and recreate them in Step 5.

Atlassian Support: After migrating to Bitbucket Cloud

Clean up users and email addresses first

USER IMPACT

The preflight checks look for unique, valid user email addresses. Users with invalid addresses can only come across as inactive users, with no permissions.

WHAT TO DO

Fix blank, shared or duplicate email addresses on the server before the test migration.

Atlassian Support: Use the Bitbucket Cloud Migration Assistant

FINISH SAFELY

How do you check it worked, or roll back?

Full validation checklist

6 checks

Run these before you declare the migration complete. Record the results in the data validation report.

Check	How to check	What good looks like
Branches, tags and commits match	In a fresh clone from each side run `git branch -r | wc -l`, `git tag | wc -l` and `git rev-list --all --count`, and compare.	Counts are identical, and the default branch head is the same commit on the server and in Cloud.
Pull requests arrived	Compare open, merged and declined pull request counts per repository, and open a few busy ones with their owners.	Counts match, and sampled pull requests keep comments, tasks and reviewers.
LFS content is real	Clone from Bitbucket Cloud and run `git lfs ls-files`, then open a few of the files.	Files download with their content, not as text pointer files.
Settings and integrations are in place	Try a direct push to the default branch from a developer account, and send a test delivery from each webhook.	The push is rejected, each webhook shows a successful delivery, and Jira shows linked commits.
Builds run from Cloud	Trigger a build for each repository and check the clone URL in its log.	Every build is green and clones from bitbucket.org.
The server is no longer written to	Check the last push date on each migrated repository on the server.	No pushes after the freeze started.

Rollback plan

If it goes wrong —

ROLL BACK IF

- The production run fails for key repositories and cannot be rerun inside the freeze window
- Builds or deployments cannot run from Bitbucket Cloud and the server is still needed to ship
- Pull request history is missing in a way the owners will not accept

POINT OF NO RETURN

The assistant copies from the server and leaves it in place, so until developers push to Cloud you can restore write access and carry on. Once people commit and merge in Cloud, that work exists only there and must be pushed back by hand. Decommissioning the server cannot be undone.

1

Restore write access on the server

Where: Bitbucket repository and project permissions on the server

Give developers write access to the wave's repositories again and tell them to keep working on the server.

✓ **You should see:** Developers can push and open pull requests on the server.

2

Push back any commits made in Cloud

Where: Terminal on your migration machine

For each repository where people pushed to Cloud, mirror-clone it from Cloud and push the new branches and tags to the server. Recreate any pull requests opened in Cloud.

```
git clone --mirror https://bitbucket.org/contoso/payments-api.git
cd payments-api.git
git push --all ssh://git@bitbucket.contoso.com:7999/PAY/payments-api.git
git push --tags ssh://git@bitbucket.contoso.com:7999/PAY/payments-api.git
```

✓ **You should see:** The server shows every branch and tag created in Cloud.

3

Revert remotes and builds

Where: Each developer's clone, and your CI settings

Set each clone's origin back to the server URL with `git remote set-url origin`, and point build jobs back at the server.

✓ **You should see:** git remote -v shows the server and builds run from it again.

Agree the triggers and who decides before the go / no-go meeting. Use the rollback plan template.

Common problems and fixes

6 errors

`Preflight check fails: repository too large`

Cause: The repository is over the Cloud size limit the preflight check applies.

Fix: Clean up history or move large files to Git LFS on the server, then rerun the checks.

`Preflight check fails: plan or workspace`

Cause: The workspace is on the Free plan, or it is the default personal workspace.

Fix: Use a workspace you created, on the Standard or Premium plan.

`Invalid repository error when rerunning`

Cause: Repositories from a stopped or failed run are already in the workspace.

Fix: Delete the repositories that already landed (unless unchanged), then create a new migration.

`Users arrive as inactive with no permissions`

Cause: Their email addresses were invalid, so they were migrated as inactive users.

Fix: Fix the addresses, then give the users access in Cloud.

`Build fails after migration: LFS files are pointers`

Cause: LFS files are not migrated by the assistant.

Fix: Run the LFS push in Step 4 for that repository.

`Fork link missing`

Cause: Forks are migrated as normal repositories.

Fix: Merge open pull requests from forks before migrating; recreate any forks you still need.

Search the full error library →

FAQ

Frequently asked questions

Is Bitbucket Data Center going away?

Atlassian has set 28 March 2029 as Data Center end of life, when licences expire and become read-only. Existing Bitbucket customers can instead buy the Bitbucket Hybrid License, which Atlassian says covers Data Center and Bitbucket Cloud from early 2027.

Does the migration assistant move pull requests?

Yes. Titles, descriptions, status, diffs, comments, tasks, authors and reviewers migrate. Comments and tasks are flattened, and pull request attachments and the activity feed do not migrate.

Are Git LFS files migrated?

No. Atlassian documents a separate mirror-clone and git lfs push after the repository is migrated.

Are permissions migrated?

Groups, permissions and branch permissions can be migrated only into workspaces administered through Atlassian Administration. Otherwise recreate them after each wave.

Do Jenkins or Bamboo builds move?

No. Atlassian's migration pages do not cover CI, and build statuses are not migrated. Point each build job at the Cloud repository, or move it to Bitbucket Pipelines.

Can I move Bitbucket Server to GitHub instead?

Yes. GitHub Enterprise Importer migrates Git repositories and pull requests from Bitbucket Server to GitHub Enterprise Cloud. CI pipelines stay in Bitbucket, and permissions are recreated as GitHub teams.

SOURCES

Official sources

9 official pages this guide is based on

Vendors change limits and features, so confirm details on the source before you rely on them. Last checked 4 Nov 2026.

- Atlassian Support: Cloud migration methods for Bitbucket
- Atlassian Support: Use the Bitbucket Cloud Migration Assistant to migrate
- Atlassian Support: What gets migrated with the Bitbucket Cloud Migration Assistant
- Atlassian Support: Bitbucket pre-migration checklist
- Atlassian Support: After migrating to Bitbucket Cloud
- Atlassian Support: Migrating repositories containing LFS to Bitbucket Cloud
- Atlassian: Bitbucket Hybrid License
- Atlassian: Ascend to the cloud (Data Center end of life)

GitHub Docs: Overview of a migration from Bitbucket Server to GitHub Enterprise Cl...