



GitHub and Cloud Integration: Together at last

Morten Wittrock, SIT Weinheim, September 2026

Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP's intentional or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Roadmap Disclaimer

Any roadmaps shown in this presentation are for informational purposes only. The items, dates and timing within such roadmaps are subject to change or delay and the continued development or release of a project on the roadmaps is at the sole discretion of SAP.

Agenda

Who is Morten?

Version control, Git and GitHub

Cloud Integration and version control

Demo

What's next?

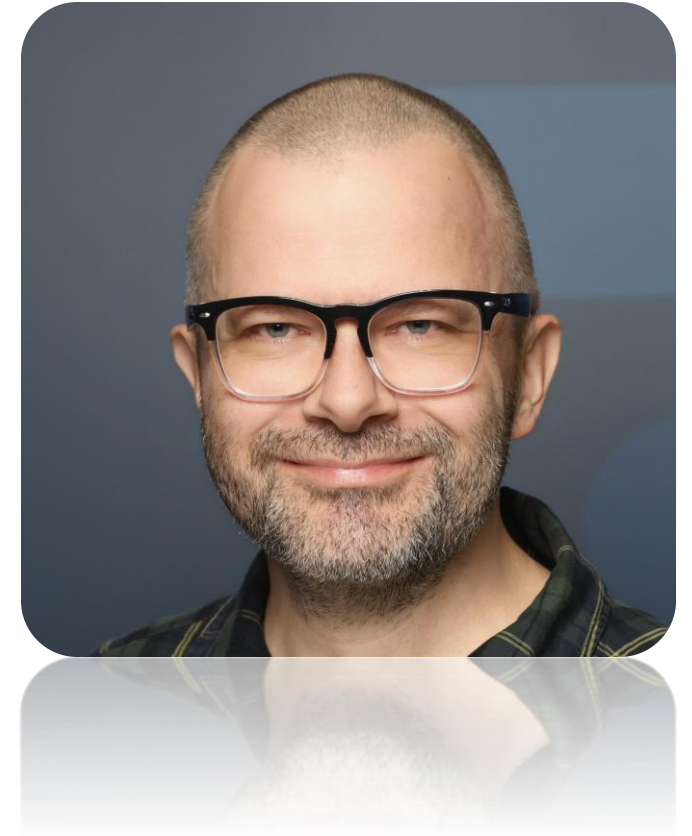
How to try this out

Q&A



Who is Morten?

- Morten Wittrock
- Part of the SAP Integration Suite product management team in Walldorf
- Based in Heidelberg, Germany
- German last name but I'm actually Danish
- Member of the SAP Mentors until I joined SAP
- SAP PRESS author
- Frequent speaker at industry and community events
- One of the organisers of the SAP Inside Track event in Copenhagen
- Founder of the CPILint open-source project



A light blue geometric shape, resembling a parallelogram with a diagonal cut, serves as a background for the title text.

Version control, Git and GitHub

Version control overview

- Version control systems record the changes you make to a set of files over time
- Often the files are source code, but they could be anything
- Such a captured change is called a commit
- A commit records:
 - *What* was changed
 - *Who* did it
 - *When* they did it
 - *Why* they did it
- That last one depends on people writing decent commit messages (🙌) or on a linked ticket
- Previous versions can be:
 - Inspected
 - Compared → A “diff”
 - Restored
- People can work in parallel in so-called branches that can later be merged back together

Why do you need version control?

Fix mistakes

You can always go back to a previous version if a change causes problems.

Complete history

The commit history documents everything that ever happened to your files from day one up to the current version.

Isolate new development

New features can be built on a separate branch that is separate from the main line.

Manage releases

Tags identify specific releases so that they can be, for instance, restored or patched later.

Work in parallel

Multiple people can work on the same files in parallel. The changes are merged later (and any conflicts dealt with).

Audit and compliance

Hardening and cryptographic hashes give auditors a tamper-proof history of changes.

The Git version control system

- Version control systems existed decades before Git, like
 - Revision Control System (RCS)
 - CVS
 - Subversion
- And there are contemporary alternatives like:
 - Fossil
 - Mercurial
 - Perforce
- But Git is far and away the most widely used version control system today
- Created in 2005 by Linus Torvalds to manage the Linux source code
- Git is different to earlier systems in central ways:
 - Distributed → Each cloned repository contains the whole history
 - Works offline → Push and pull to/from a remote repository when you need to
 - Commits, branching and merging are local operations → Very fast



Git hosting platforms

- Git hosting platforms offer a shared central copy of your repositories
- Plus additional features like:
 - User management
 - Code reviews
 - Issue management
 - Automation
- Some Git hosting platforms are GitLab, Azure DevOps and Bitbucket
- But GitHub is the largest one by a wide margin
- Founded in 2008 and acquired by Microsoft in 2018
- The number one platform for open-source software projects
- GitHub in numbers (2025):
 - 630M repositories in total → Public and private
 - 180M developers in total
 - >1 new developer per second



A light blue geometric shape, resembling a parallelogram with a diagonal cut, serves as a background for the title text.

Cloud Integration and version control

Version management in Cloud Integration before GitHub

- Didn't Cloud Integration have versions already?
- Yes – but only to some extent
- You could:
 - Save named versions of your integration flow with a message
 - Revert to a previous version
- This gets you some of the way
- But compared to a full version control system it is rudimentary
- You *could* still store your integration flows in Git
- They're just files and folders after all
- But this would either be:
 - manual 😞
 - or using external tools → like Project Piper or FlashPipe

Version History		
Draft	Last Modified: 9/18/2	Last Modified by: morten.wittrock@sap.com
1.0.1	Created: 9/18/26	Created by: morten.wittrock@sap.com
1.0.0	Created: 9/18/26	Created by: morten.wittrock@sap.com
1.0.0	Created: 9/18/26	Created by: morten.wittrock@sap.com

But then!

GitHub integration with basic Github operations

Click a tag to explore related items in the board. ↗

Products:

SAP Integration Suite >

DELIVERED

DELIVERED IN Q3 2026

Focus Topics:

Integration >

Digital Transformation >









Overview

GitHub integration with Cloud Integration capability in SAP Integration Suite to manage integration flow artifact in Git repositories with basic clone/import, pull and push Git operations

Benefits

Help integration developers natively manage integration flow in Git repositories and improve the Git version management

Features

GitHub support in SAP Integration Suite Available Now

You can now integrate GitHub repositories and perform Git operations in your integration flows

[Integrating with GitHub](#)

First iteration of the GitHub integration

- The first iteration of the GitHub integration arrived in Q3 2026
- It introduced the basic GitHub operations
- All supported natively in the developer tools → No external utilities needed
- Meaning you can now:
 - connect your tenant to repositories on GitHub
 - commit and push iflows from Cloud Integration to a repository
 - pull iflows from a repository to Cloud Integration
- This enables:
 - proper diffs between versions
 - separate branches for feature development, fixes, etc.
 - a well-established workflow for local development → writing Groovy scripts, for instance
 - CI/CD pipelines → meaning GitHub Actions in the first release

Demo



A light blue parallelogram graphic, tilted to the right, positioned in the upper middle of the slide.

What's next?

Upcoming features

- The first release was the baseline but a lot more is coming
- Support for more Git platforms
- Visual comparison of artifacts → Text is fine for code, for iflows not so much
- Graphical conflict resolution for both artifacts and resources
- Support for more artifact types → Going beyond integration flows
- Support for packages → Right now operations are per artifact
- OData API for Git operations → Example: Trigger a pull in the next environment from CI/CD
- On-prem GitHub support
- Persisted tokens → No more copy-pasting 🗨️

Iteration two (planned for Q1 2027)

GitHub integration with support for conflict resolution and integration flow comparison

Click a tag to explore related items in the board. ↗

Products: SAP Integration Suite > FUTURE RELEASE | PLANNED FOR Q1 2027

Focus Topics: Digital Transformation > Integration >



Overview

GitHub integration with the Cloud Integration capability in SAP Integration Suite to enable management of integration artifacts with Git repositories along with conflict resolution

Benefits

Enable integration developers to manage integration flow with Git repositories along with conflict resolution and integration flow comparison



How to try this out

How to try this out

- The GitHub integration feature is available in the SAP Integration Suite trial tenant
- Your next steps:
 1. Create a [free trial tenant](#)
 2. Create a [free GitHub account](#)
 3. Check out [this blog post](#) to get started
- The full documentation is available on the [SAP Help Portal](#)
- If you are joining TechEd in Berlin, check out [hands-on workshop IN160](#)

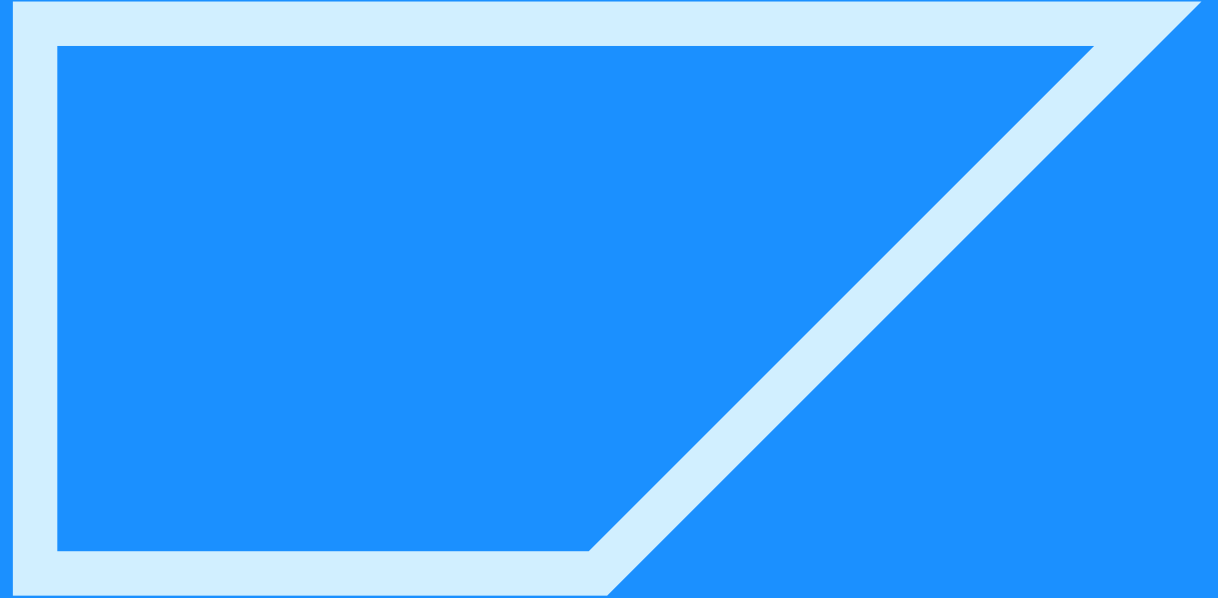
IN160 | Explore version control and automation with GitHub in Cloud Integration



Hands-on workshop

Bring integration development into the workflow other developers already use. Set up GitHub access in the Cloud Integration capability of SAP Integration Suite, commit and version integration flows from the tools in Cloud Integration, work collaboratively on the same artifacts, and trigger a continuous integration and continuous delivery pipeline.

Q&A



Thank you!

Contact information:

morten.wittrock@sap.com

<https://www.linkedin.com/in/mwittrock/>