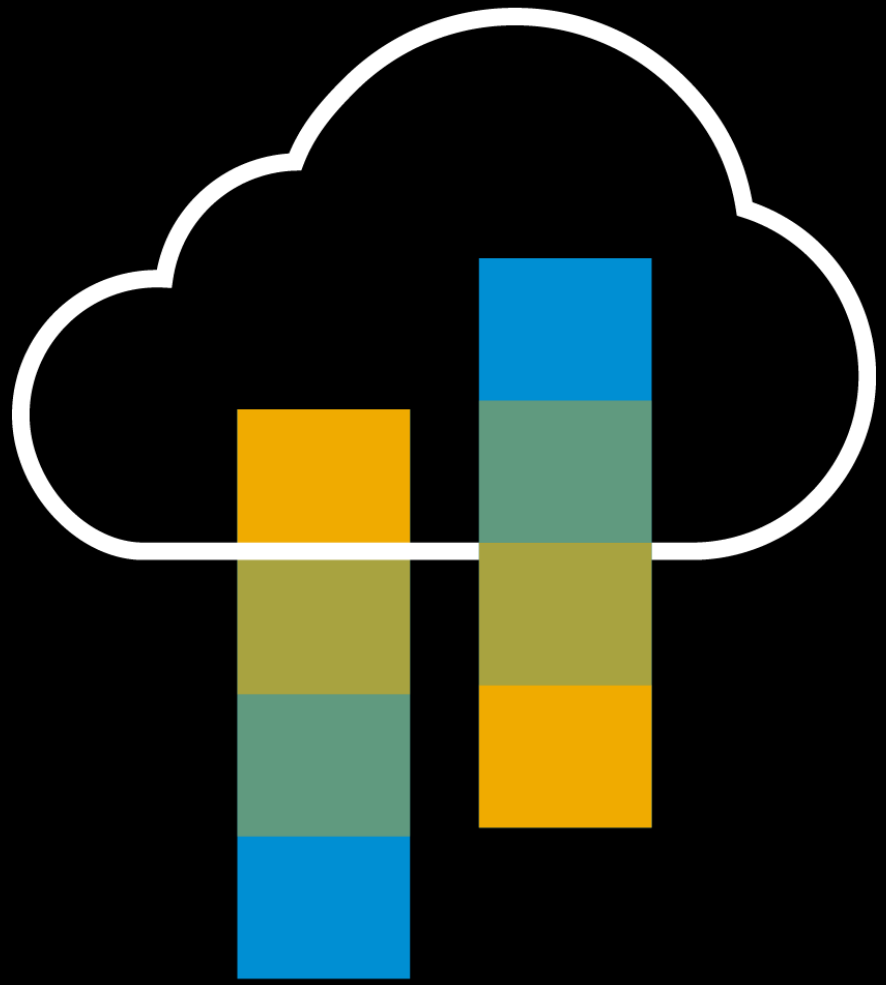
SAP Cloud SDK
Continuous Delivery

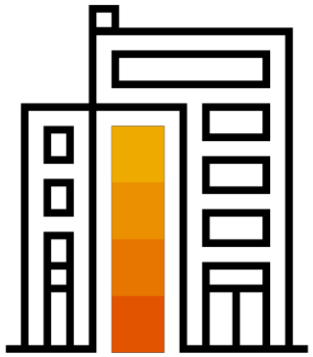


SAP Run Simple

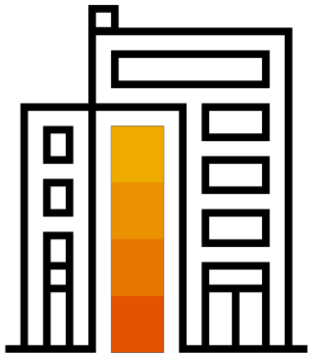
Continuous Delivery

Introducing DevOps and Continuous Delivery

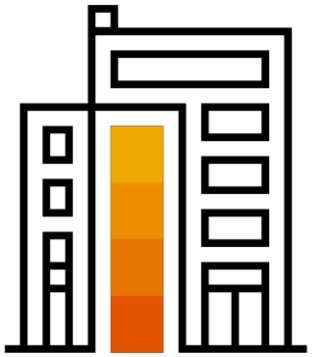
DevOps in the IT industry



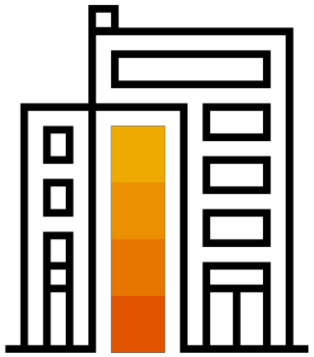
Amazon



Netflix



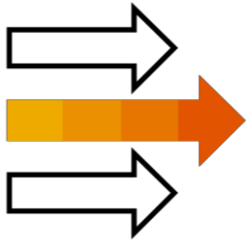
ebay



Linkedin

Introducing DevOps and Continuous Delivery

DevOps principles



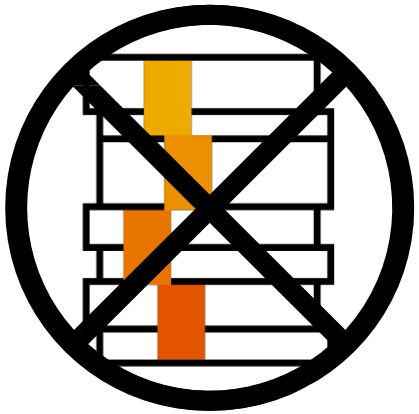
Flow aims for the quick delivery of software from development to customers.

Feedback aims to quickly sense problems in the value chain, leading to a continuous improvement of process and product quality.

Continuous learning and experimentation aims at establishing a culture that motivates experimentation, risk-taking, and learning from failure.

Introducing DevOps and Continuous Delivery

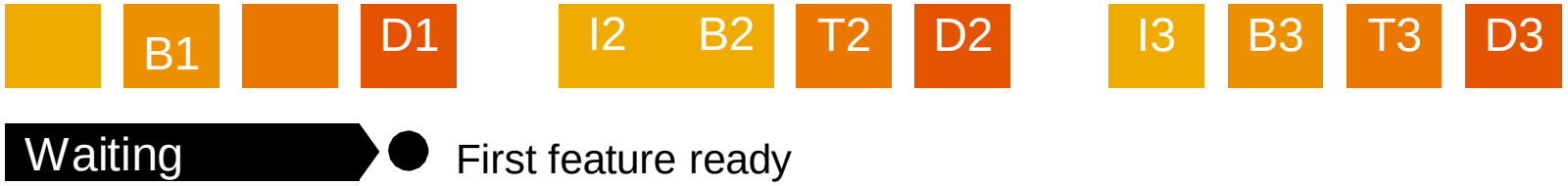
DevOps principles – Flow



Avoid batch work style

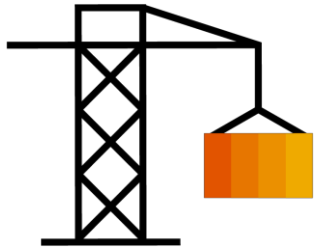
Large batches

Single piece flow

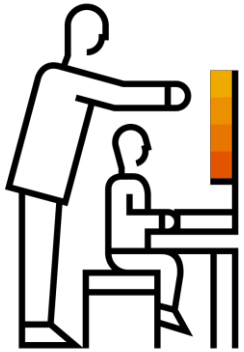


Introducing DevOps and Continuous Delivery

DevOps principles – Typical flow inhibitors



Creation of environments



Manual testing

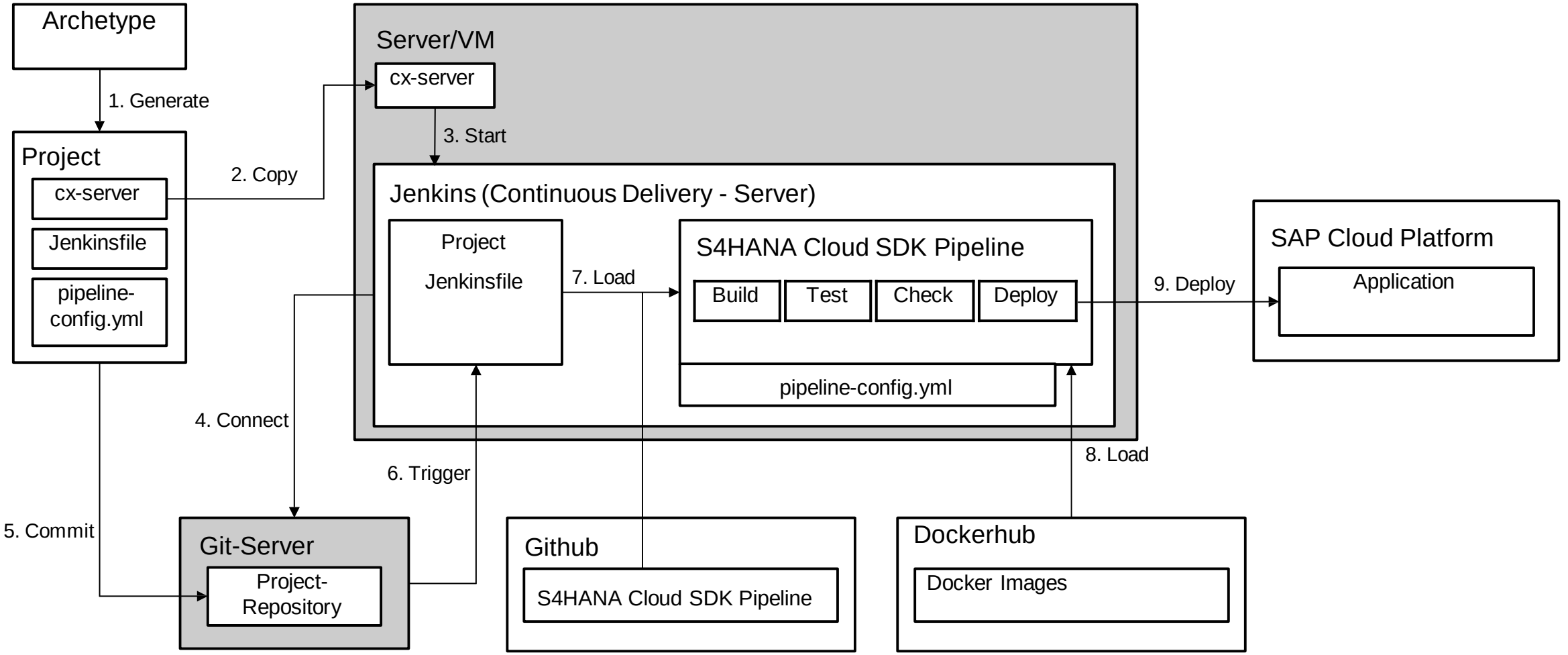
Overly tight architectures on technical or organizational level

Complex and manual deployment processes

Value Propositions

- **Rapid productivity** for partners starting development of S/4HANA extension apps using the SAP Cloud SDK
- Partner **compliance to SAP guidelines, quality standards**, legal regulations, and continuous delivery best practices.

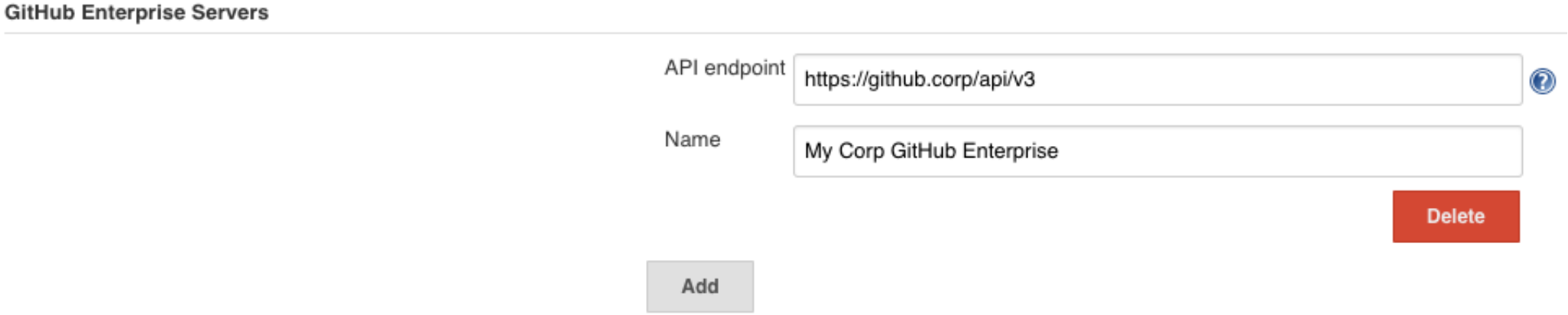
Rapid Productivity



Project setup

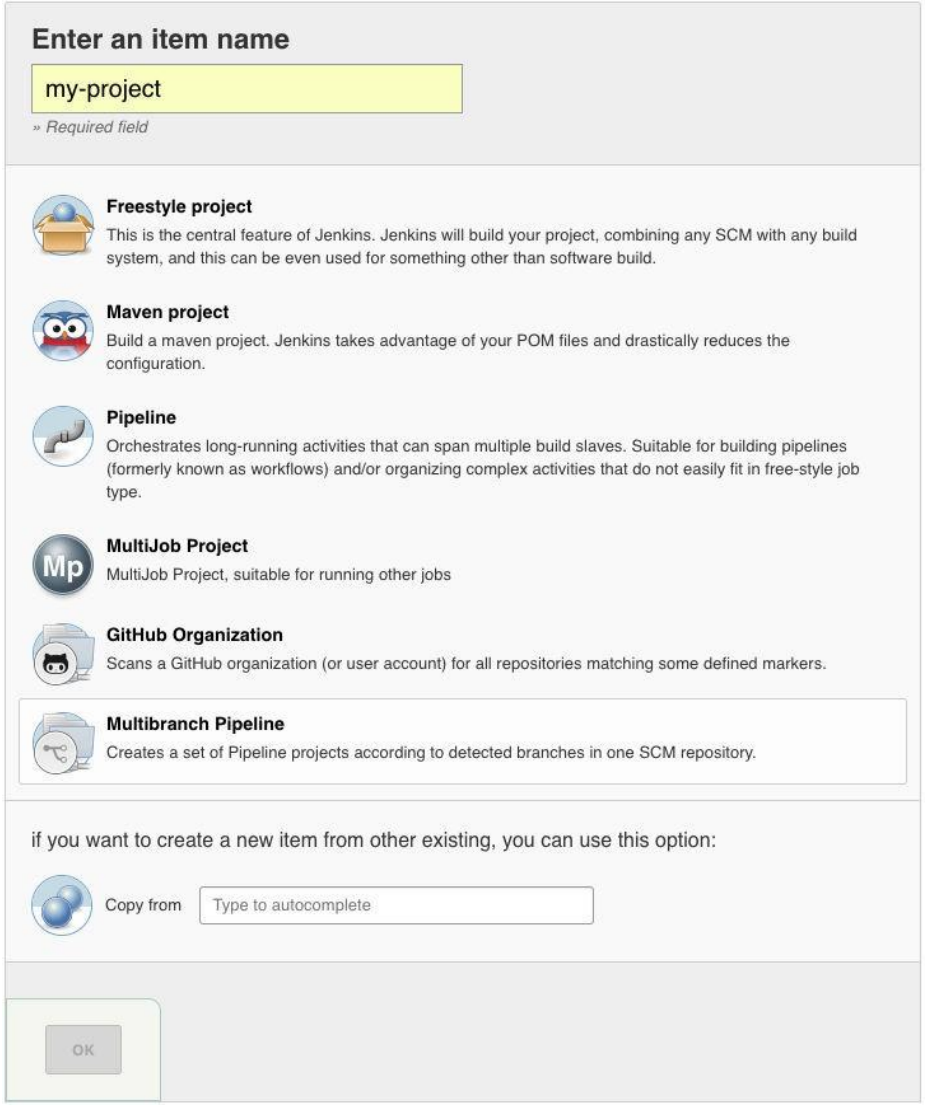
Setup GitHub Enterprise

- Go to your Jenkins main page and navigate to **Manage Jenkins -> Configure System**
- On the configuration page, look for **GitHub Enterprise Servers.**
- Here, add the API endpoint of your GitHub Enterprise Server, which typically ends with ***/api/v3***, and assign a name to it.
- Now you are ready to create a build job for your project.



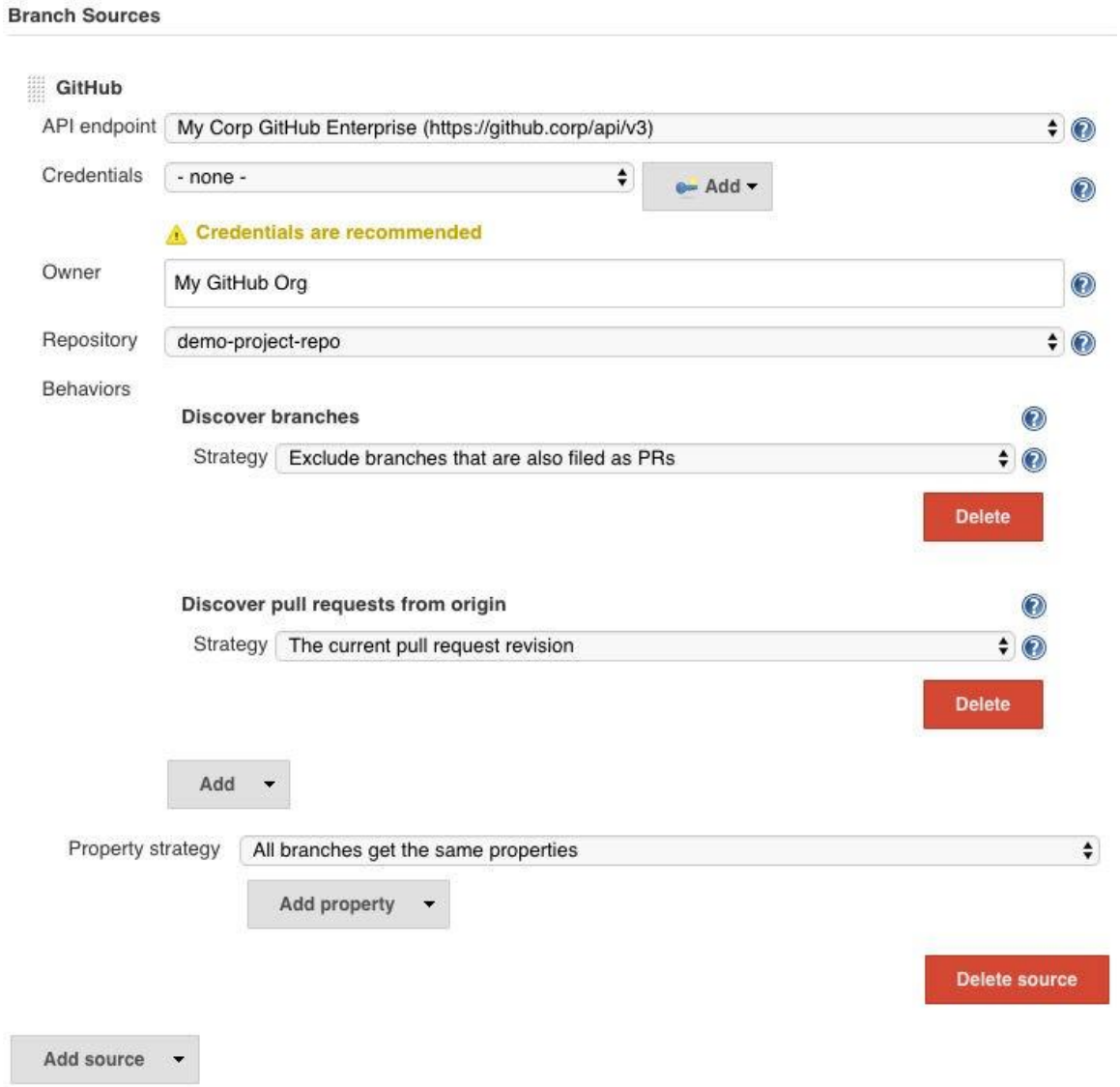
Create a Jenkins job for your project

- To create a build job for your project, navigate to **New Item** in the Jenkins main menu.
- On the following screen, specify a name for your project's build job
- Choose **Multibranch Pipeline**
- Click **OK**



Setup the SCM source of your project's build job

- Look for **Branch Sources** and add one by specifying your GitHub Enterprise endpoint, organization (via field "Owner"), and repository
- If your repository is set to private, please also create and use a pair of suitable credentials and choose them for your repository
- Finally, save your updated configuration by clicking the **Save** button at the bottom of the page

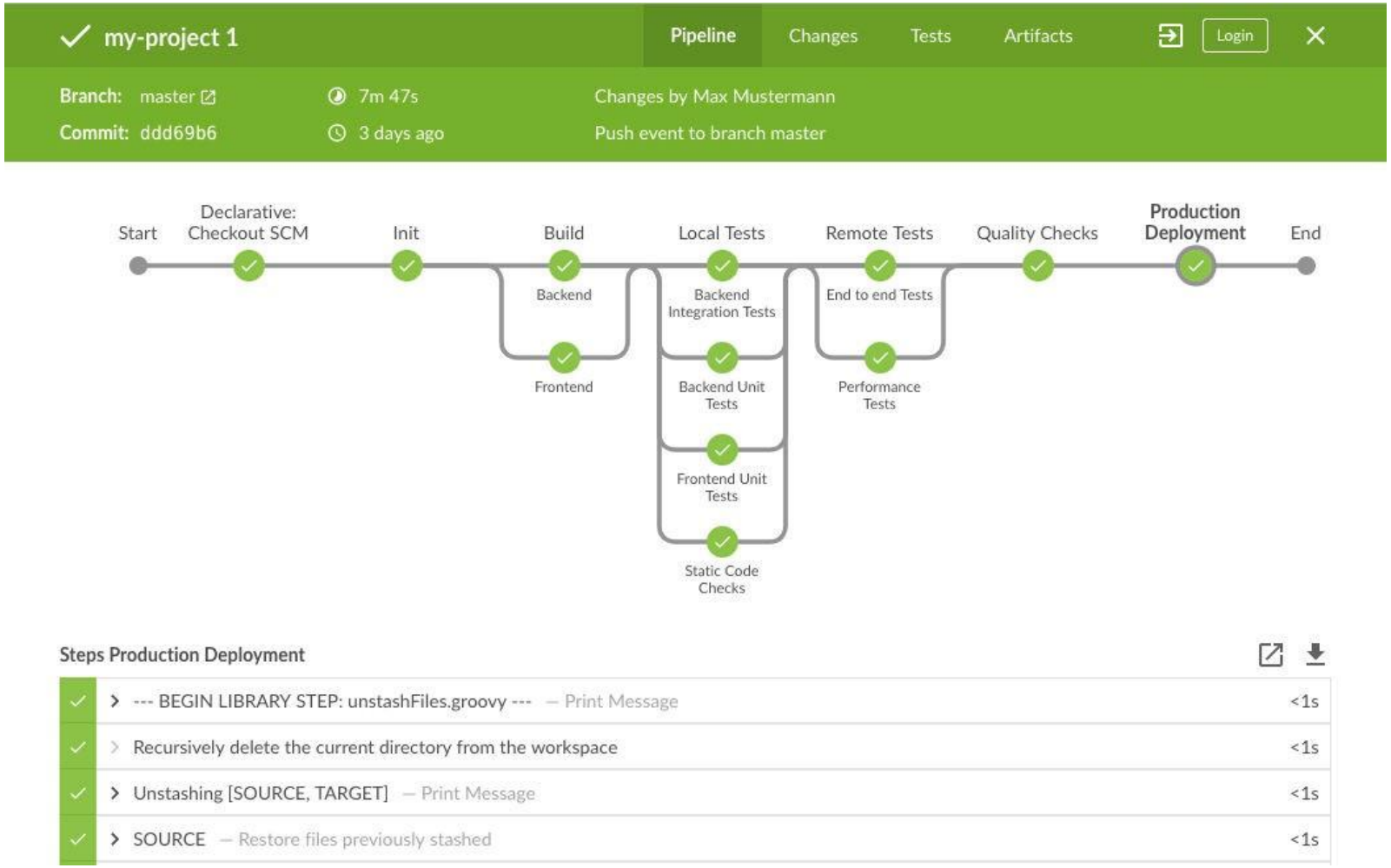


Check your project (1/2)

- To check the building lifecycle of your project you can click on the **Open Blue Ocean** link in the left side menu
- This will open the Pipeline view of your project

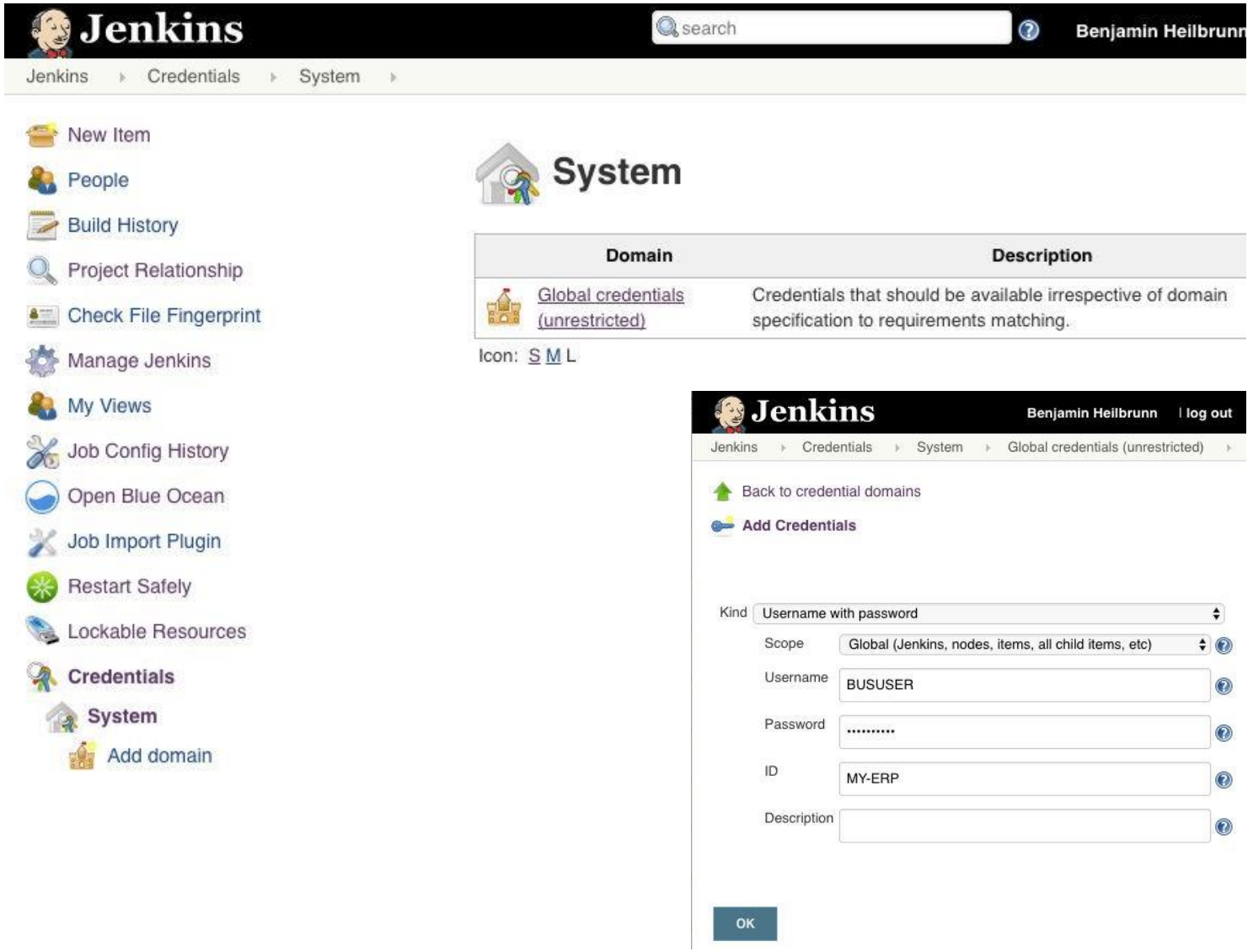
Check your project (2/2)

- The Pipeline views opens and you can check how the build process goes
- Production Deployment phase has been skipped because for the job in this example there is not deployment target configured



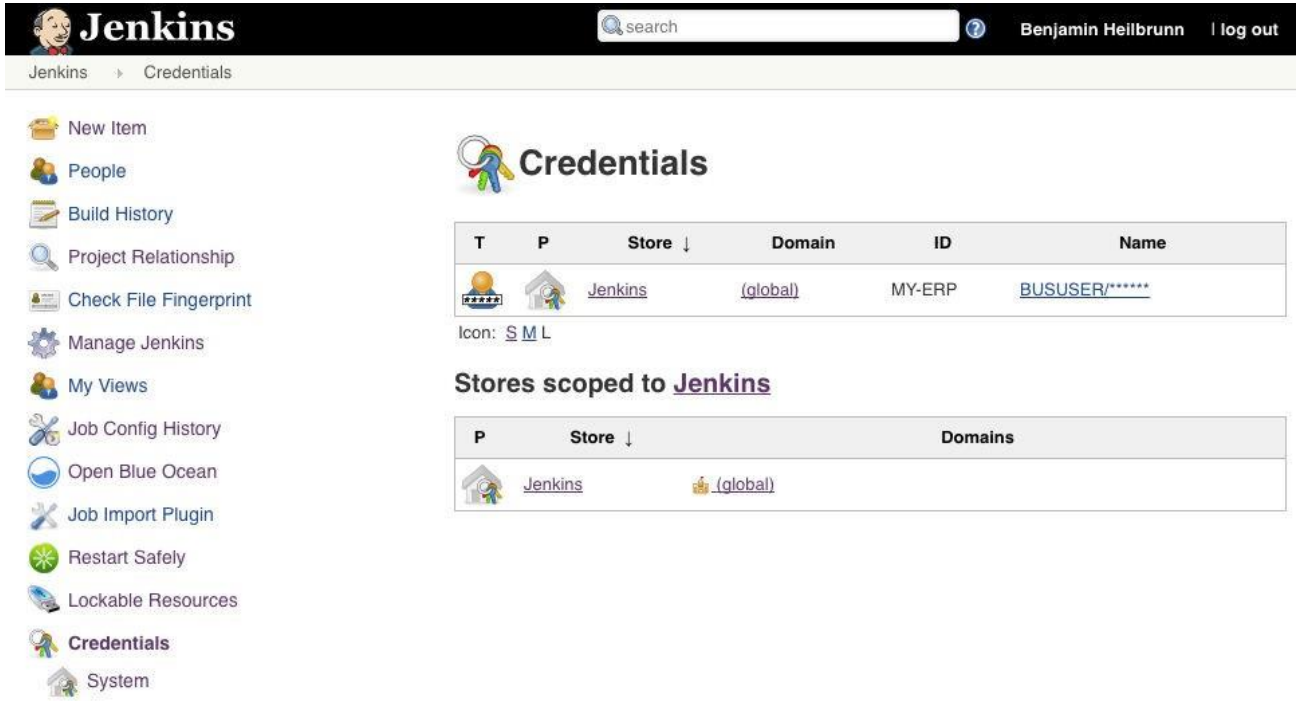
Configure your credentials (1/2)

- Credentials can be maintained directly in Jenkins
- Click on the Credentials item in the left side menu and add the required credentials (i.e. for accessing GitHub or your ERP Endpoint)



Configure your credentials (2/2)

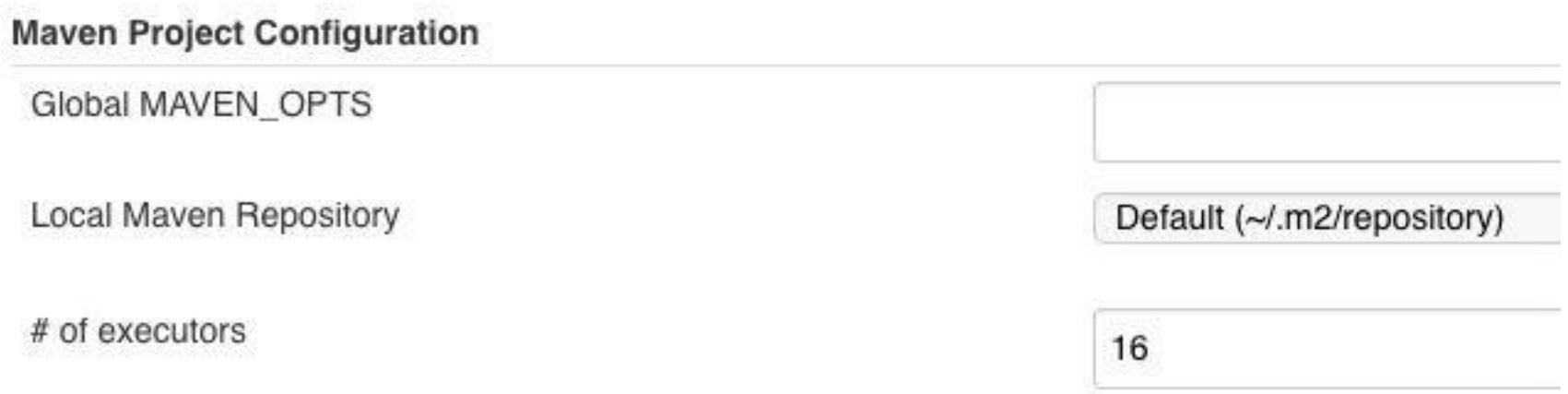
- Finally, you can now leverage this credentials record by adding the credentials configuration property to the integrationTests stage of your pipeline_config.yml.
- The example below shows am example of mapping it to the system alias “MyErpSystem”, defined earlier in the systems.yml or json file.




```
stages:
  integrationTests:
    credentials:
      - alias: 'MyErpSystem'
        credentialId: 'MY-ERP'
```

Configure number of executors

- Another parameter to setup is the number of executors. You can adapt the number of executors to match the power of your host machine.
- By default, the Cx Server uses 4 executors, which means that up to three stages of build jobs can run in parallel. We recommend to increase the number executors to **at least the number of cores of your machine**. This will help your Cx Server to finish build jobs faster, especially when running multiple in parallel. The number of executors can be configured in “Manage Jenkins » Configure System” in the section “Maven Project Configuration”.



Questions?

Thank you.