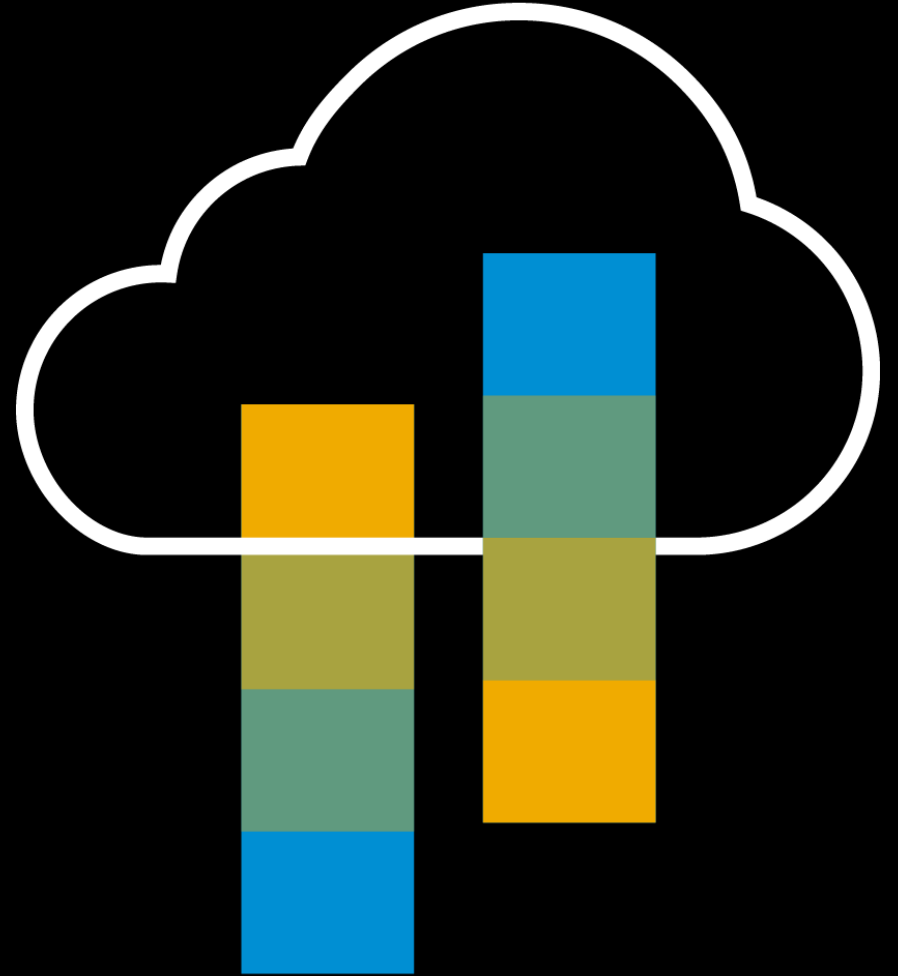


SAP Cloud SDK

Continuous Delivery

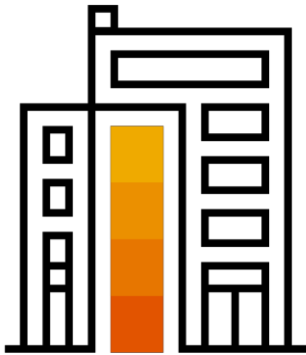




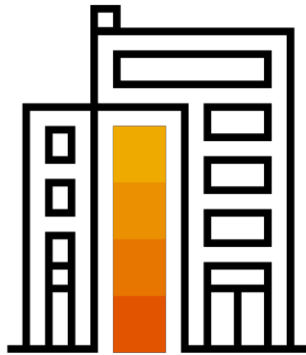
Continuous Delivery

Introducing DevOps and Continuous Delivery

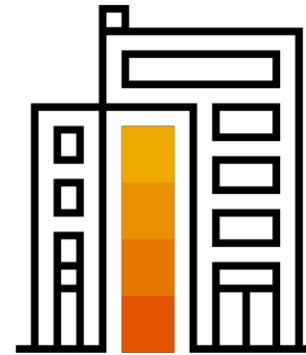
DevOps in the IT industry



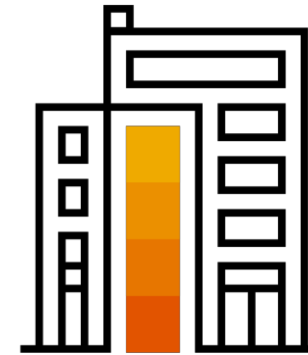
Amazon



Netflix



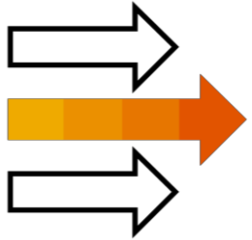
ebay



Linkedin

Introducing DevOps and Continuous Delivery

DevOps principles



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



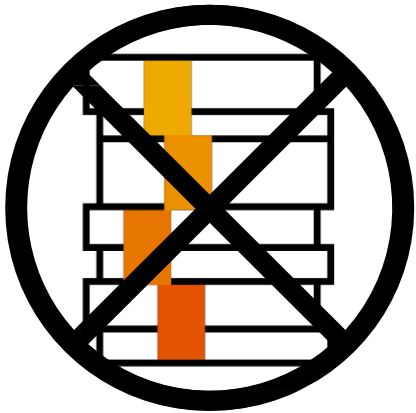
Feedback aims to quickly sense problems in the value chain, leading to a continuous improvement of process and product quality.



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

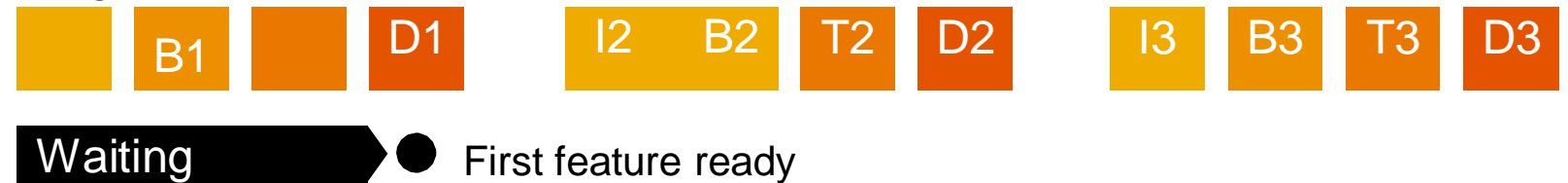


Avoid batch work style

Large batches

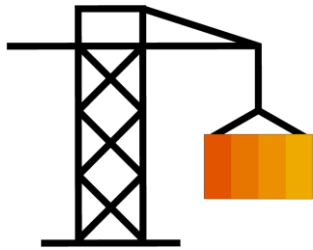


Single piece flow



Introducing DevOps and Continuous Delivery

DevOps principles – Typical flow inhibitors



Creation of
environments



Manual testing



Overly tight
architectures
on technical or
organizational level

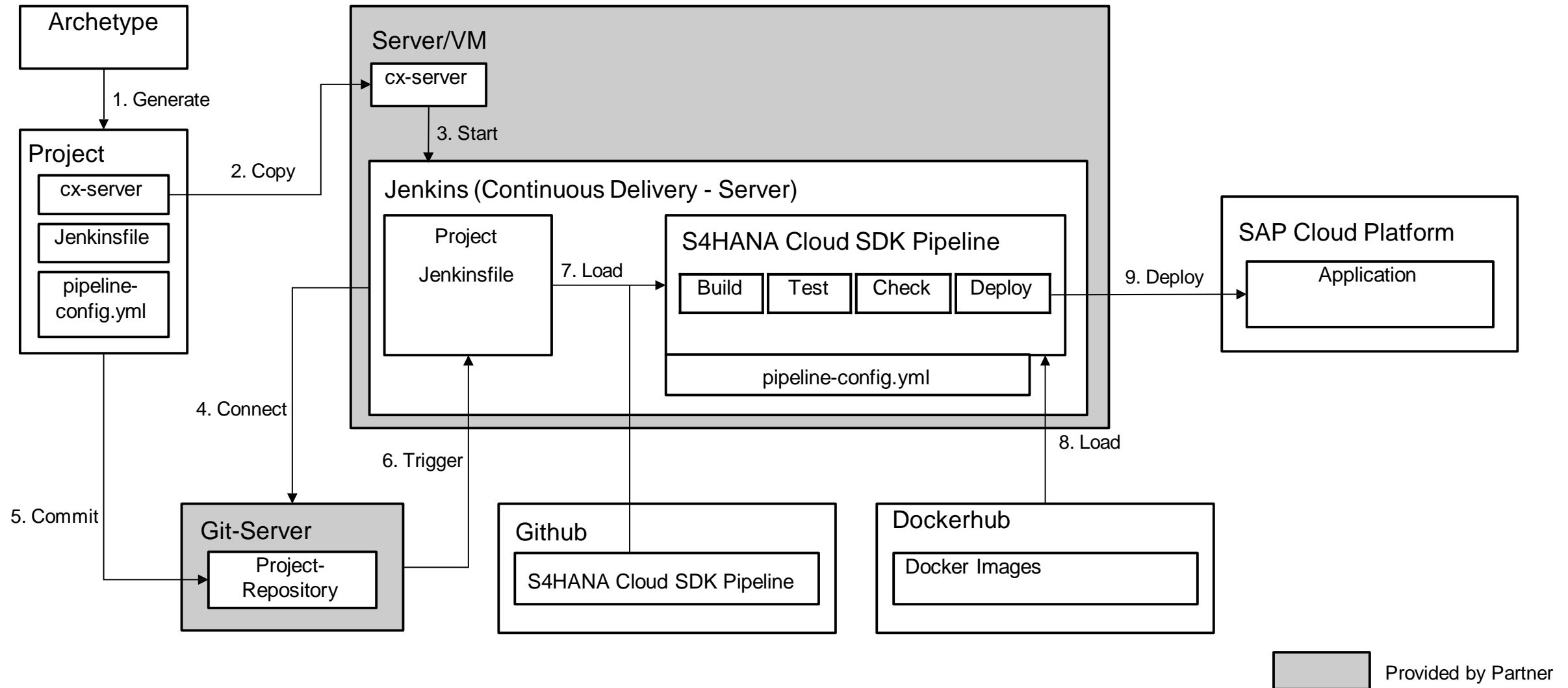


Complex and manual
deployment processes

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.

GitHub Enterprise Servers


API endpoint	<input type="text" value="https://github.corp/api/v3"/>	
Name	<input type="text" value="My Corp GitHub Enterprise"/>	
	<input type="button" value="Add"/>	<input type="button" value="Delete"/>

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**

Enter an item name


» Required field




Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.




Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.




Pipeline
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



MultiJob Project
MultiJob Project, suitable for running other jobs




GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.



Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

if you want to create a new item from other existing, you can use this option:



Copy from

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

Branch Sources

GitHub

API endpoint

Credentials Add

Credentials are recommended

Owner

Repository

Behaviors

Discover branches

Strategy Delete

Discover pull requests from origin

Strategy Delete

Add

Property strategy

Add property

Delete source

Add source

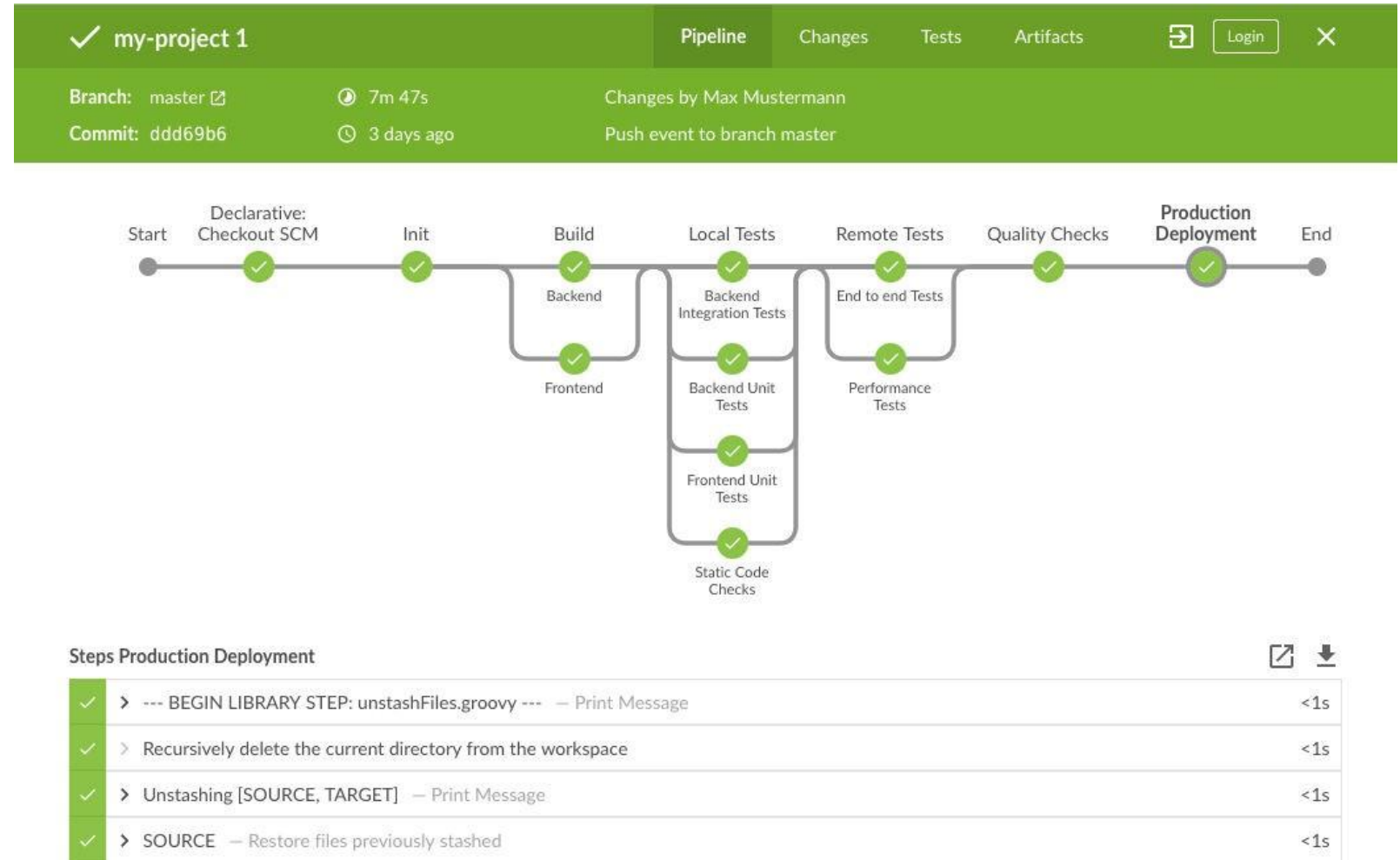
Check your project (1/2)

- 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)

Jenkins search Benjamin Heilbrunn

Jenkins > Credentials > System

New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Job Config History
Open Blue Ocean
Job Import Plugin
Restart Safely
Lockable Resources
Credentials
System
Add domain

System

Domain	Description
Global credentials (unrestricted)	Credentials that should be available irrespective of domain specification to requirements matching.

Icon: [S](#) [M](#) [L](#)

Jenkins Benjamin Heilbrunn | log out

Jenkins > Credentials > System > Global credentials (unrestricted)

[Back to credential domains](#)
[Add Credentials](#)

Kind:

Scope:

Username:

Password:

ID:

Description:

OK

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 an example of mapping it to the system alias “MyErpSystem”, defined earlier in the systems.yml or json file.

Jenkins search Benjamin Hellbrunn | log out

Jenkins > Credentials

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Job Config History
- Open Blue Ocean
- Job Import Plugin
- Restart Safely
- Lockable Resources
- Credentials**
- System

Credentials

T	P	Store ↓	Domain	ID	Name
		Jenkins	(global)	MY-ERP	BUSUSER/*****

Icon: [S](#) [M](#) [L](#)

Stores scoped to **Jenkins**

P	Store ↓	Domains
	Jenkins	(global)

```
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”.

Maven Project Configuration

Global MAVEN_OPTS

Local Maven Repository

of executors

Default (~/.m2/repository)



Questions?

Thank you.