netmind
a bts company

06

CI/CD – Artifacts management

Index

01 Artifacts

01

What is a software artifact?

- A software artifact is **an item that is produced during the development process**.
- This can be a **data model, a prototype, workflow diagram, a design document, or a setup script**.
- Once created, artifacts are important **throughout the software development** process.
- Software artifacts help make the process of **developing software less difficult** over time.
- **Keeping relevant artifacts in a repository** enables developers to access artifacts at any time, from one location.

Artifact created in app lifecycle

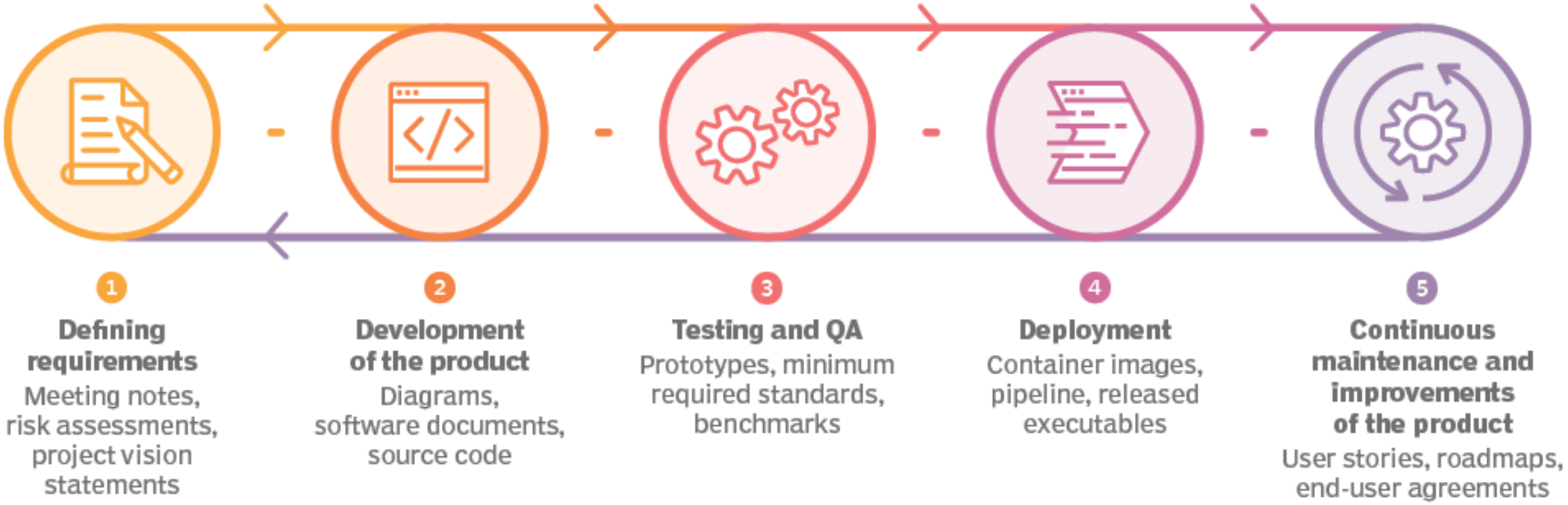


Image Source: techtarget.com

Artifact repository

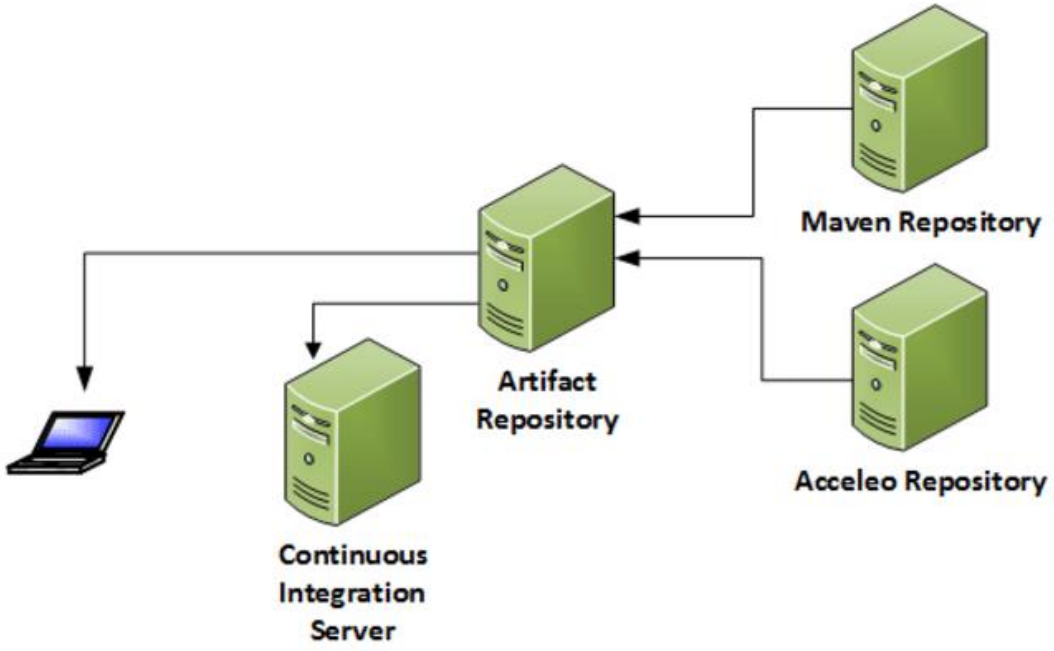


Image Source: researchgate.net

- Software artifacts need to be of easy traceability, so they need to be properly stored in an artifact repository.
- **Remote repository**: Uses a remote URL and is sometimes even hosted by an outside provider. While it is possible to remove software artifacts, it is not possible to add any.
- **Local repository**: In this scenario, software artifacts are stored on a server on-premise and are deployed and managed locally.
- **Virtual repository**: This is a combination of the two repository types above. Held under one single URL, users have access to remote and local artifacts and can simply delete or add them.

netmind
a bts company

02 Nexus

01

Nexus

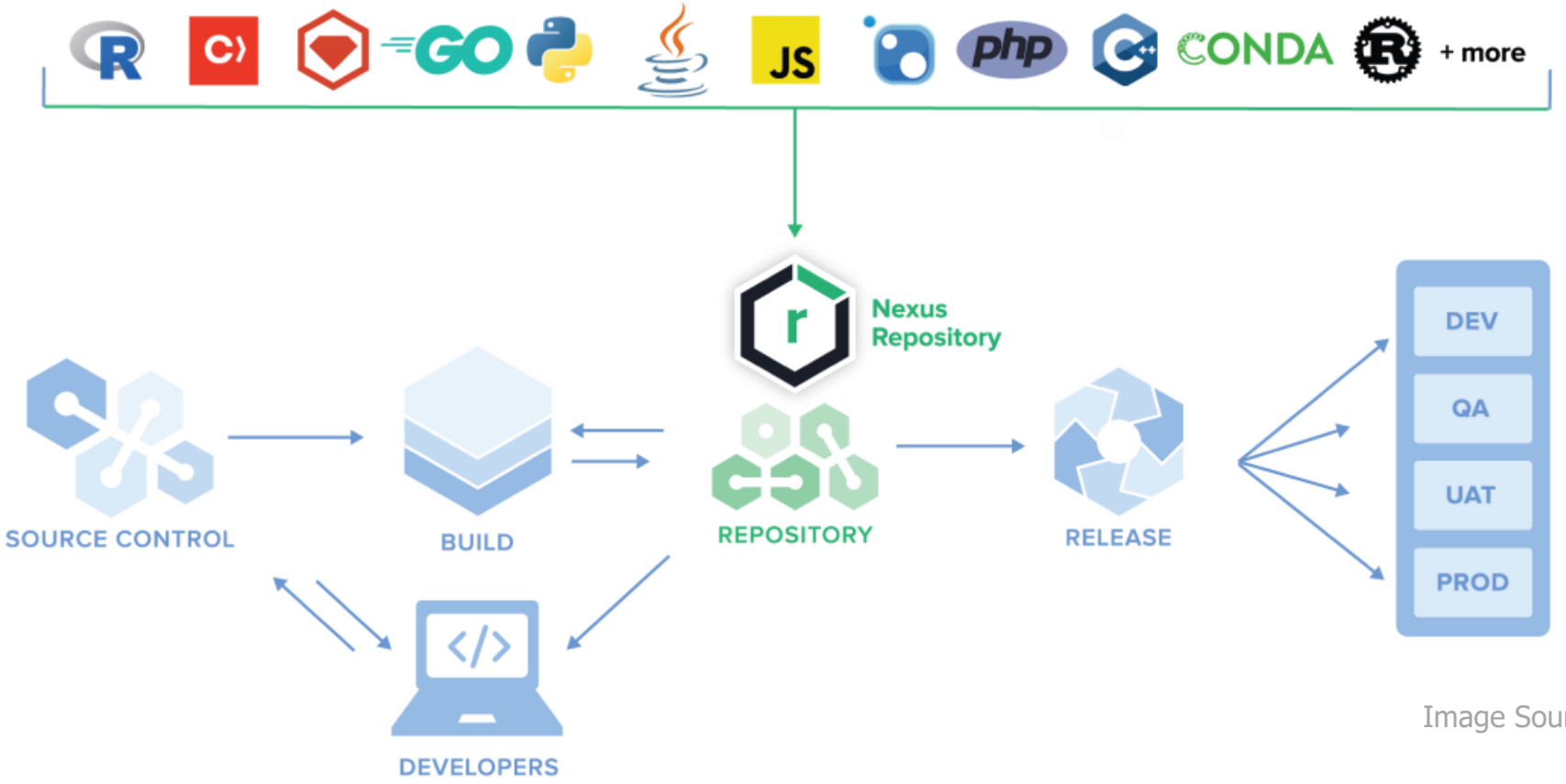


Image Source: sonatype.com

An repository manager allows to store and retrieve build artifacts.
Host your own repositories, but also use Nexus as a proxy for public repositories.

Manage all of packages

Image Source: sonatype.com

- Store and distribute components with native package manager compatibility
- Support ecosystems like Java/Maven, npm, NuGet, PyPI, RubyGems, CocoaPods and more
- Distribute packaged and containerized apps like Docker, Helm, Yum, and APT
- Compatible with popular IDEs and CI like Eclipse, IntelliJ, Visual Studio, Jenkins

OBJECTIVE

Starting and configuring Nexus

INSTRUCTIONS

1. Follow next steps to learn how to start and configure Nexus.

20 min

Starting Nexus

1. To start Nexus, after extracting, the nexus script must be executed with a start parameter in the nexus folder:

```
cd /NEXUS_FOLDER/nexus
./bin/nexus start
or
./bin/nexus /run
```

2. And in case you want to stop Nexus you just have to write stop instead of start:

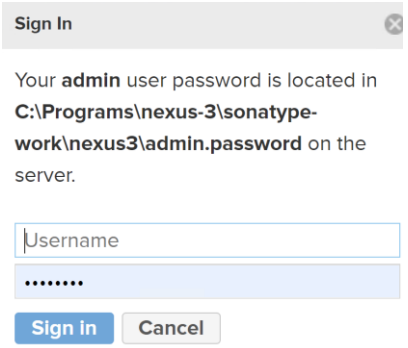
```
Ctrl+C

or

cd /NEXUS_FOLDER/nexus
./bin/nexus stop
```

Loging into Nexus

1. Once the nexus repository manager has been started, its web interface can be accessed under this URL:

   http://localhost:8081/

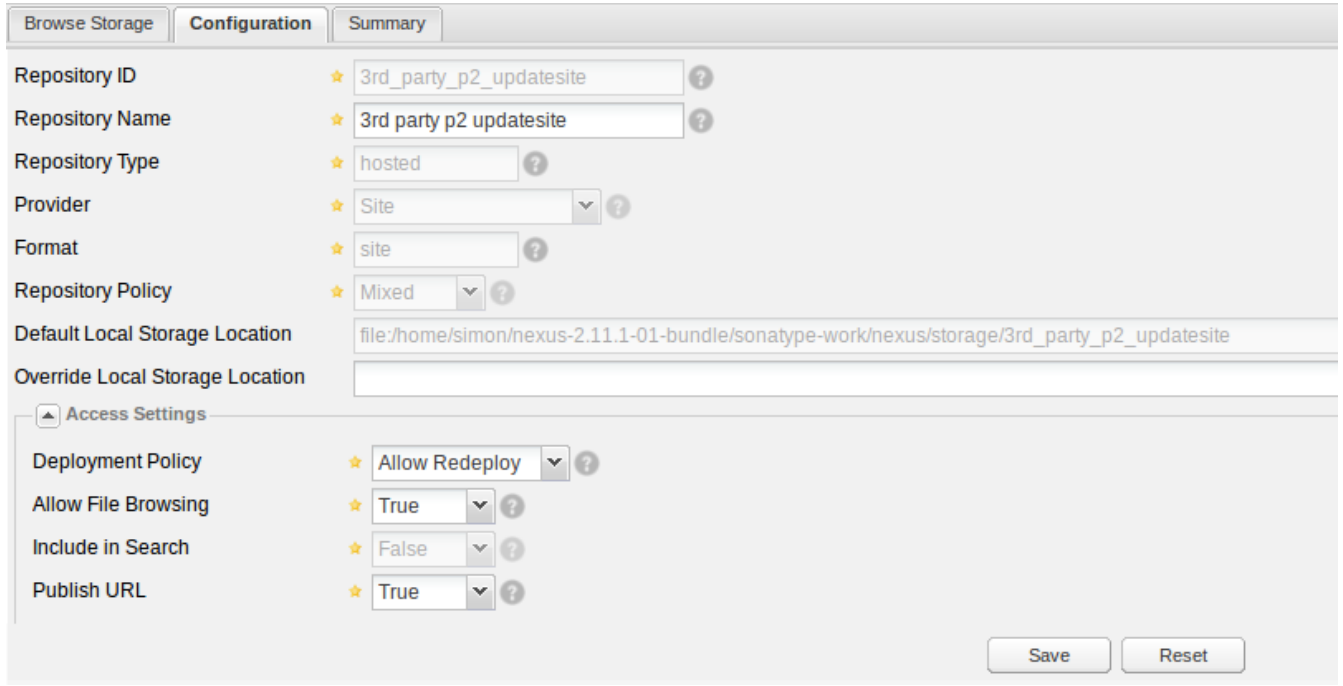
2. Follow the instructions in the modal window to login.



3. After logging in the credentials can be changed in the profile settings.

Creating a repository

1. Click on Repositories on the right hand side.
2. Select Add.. Raw Hosted Repository and use the following data:



3. Now you got a custom repository, which is hosted on your local nexus installation.

Uploading Content to Repository

1. Once you've created the repository, you have to add content to it.
2. To do this, click on the **Upload** button on the homepage, then click on the repository you want to upload the content for.
3. You will have to fill in details about the content and then upload the content.

netmind
a bts company

03 Nexus - Python

01

Upload python packages to Nexus - twine

- Twine is a utility for publishing Python packages on PyPI.
  - https://pypi.org/project/twine/
  - https://twine.readthedocs.io/en/stable/

- It provides build **system independent uploads** of source and binary distribution artifacts for both new and existing projects.

OBJECTIVE

Install and using Twine

INSTRUCTIONS

1. To upload packages you can use Twine.
2. Follow next steps:

20 min

Install

```
pip install twine
```

Create a package

Create some distributions in the normal way:

```
python -m build
```

- If you need, upgrade the build:

```
py -m pip install --upgrade build
```

- Package example:

  https://github.com/BillMills/python-package-example

Using Twine – Edit .pypirc

1. Edit **.pypirc** located at %USERPROFILE%:

```
[distutils]
index-servers =
pypi
[pypi]
repository: https://nexus.your.domain/repository/pypi-hosted/
username: nexususername
password: nexuspassword
```

2. Then:

```
twine upload
```

Using Twine – Command line

- Use command line:

```
twine upload --repository-url https://nexus.your.domain/repository/pypi-hosted/
dist/*
```

- Twine will prompt for your username and password.

Packaging Python Projects

- A Python package is **a library implementing something**, which can be exploited not only by you in the next projects, but also by the whole community.
- A package should be well documented, in order to make it exploitable by others. For this reason, every class or method should be documented, through docstrings, as explained in this very interesting article.
  - https://realpython.com/documenting-python-code/#documenting-your-python-code-base-using-docstrings

OBJECTIVE

Creating a Python package

INSTRUCTIONS

Follow one of next articles to generate a python distributable package.

1. **Option1:** https://towardsdatascience.com/how-to-convert-your-python-project-into-a-package-installable-through-pip-a2b36e8ace10
2. **Option2:** https://packaging.python.org/en/latest/tutorials/packaging-projects/

20 min

OBJECTIVE

Publish to and consume from Nexus

INSTRUCTIONS

1. Take the previous package and publish it to a customer repository
2. Once published, configure your environment for consuming it from Nexus.

20 min

Install or Download Python Package Using Nexus repositories

- To download packages from Nexus instead of Pypi , we need to config **pip.ini** , most of the time, its not only speed up build processes by caching commonly used dependencies but also help ensuring reproducible builds, since one only depends on their Nexus availability and not the public repositories.
- **pip** can also be configured to upload packages to Nexus, enabling the management of artifacts private to an organization.

OBJECTIVE

Downloading/Installing packages from Nexus

INSTRUCTIONS

1. In order to enable pip to download packages from Nexus, it is necessary to edit **pip configuration file**.
2. This can be done on a per-user, per-virtualenv or system-wide basis.

20 min

1. The per-user configuration file is located in different places on different OS'es:
    1. On Linux: ${HOME}/.config/pip/pip.conf.
    2. On Windows: %APPDATA%\pip\pip.ini.
        - Use "pip config -v list" to show the possible locations
        - If doesn't exist, it can be created manually.
2. Edit the corresponding file as follows:

```
[global]
index = https://nexus.example.com/repository/pypi-all/pypi
index-url = https://nexus.example.com/repository/pypi-all/simple
no-cache-dir = false
```

3. This will instruct pip to search for and install packages from the **pypi-all** group, previously configured in Nexus.

Download with pip (yarn) – Options:

- For edit config

```
pip config edit [--editor [nano|code|...]] [--global|--user]
```

  - Will allow to edit:

```
[global]
index = https://nexus.your.domain/repository/pypi/pypi
index-url = https://nexus.your.domain/repository/pypi/simple
```

- Command line args

```
pip install –index
```

Jenkins - Nexus

- It is very common integrate Nexus in the Ci/CD pipeline, allowing Jenkins to publish artifacts to Nexus.
- For doing this, the pipeline must include the proper technology script for uploading packages to Nexus.
- In the case of Python, we can call **Twine** as we have seen previously.
- Alternatively we can use Jenkins **"Nexus Artifact Uploader"** plugin
  - https://plugins.jenkins.io/nexus-artifact-uploader/

OBJECTIVE

Upload a package to Nexus using Twine

INSTRUCTIONS

- Configure a Jenkins pipeline for generating a python package and then upload it to Nexus using Twine.

20 min

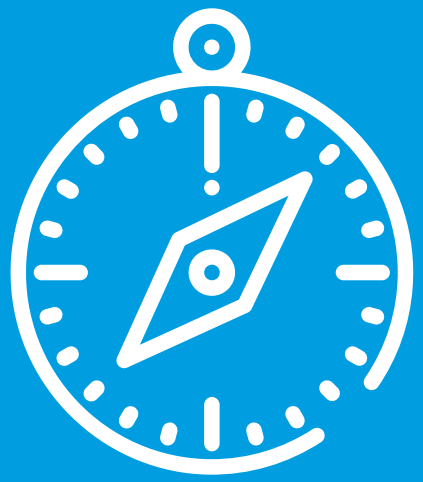
OBJECTIVE

Upload a package to Nexus using "Nexus Artifact Uploader" plugin

INSTRUCTIONS

- Follow next article steps for creating a Nexus repository and upload an artifact to it using the "Nexus Artifact Uploader" plugin.
  - https://appfleet.com/blog/publishing-artifacts-to-nexus-using-jenkins-pipelines/

30 min



Next steps

We would like to know your opinion!

Please, let us know what you think about the content.
From Netmind we want to say thank you, we appreciate time and effort you have taking in answering all of that is important in order to improve our training plans so that you will always be satisfied with having chosen us
quality@netmind.es

netmind
a bts company

Thanks!

Follow us: