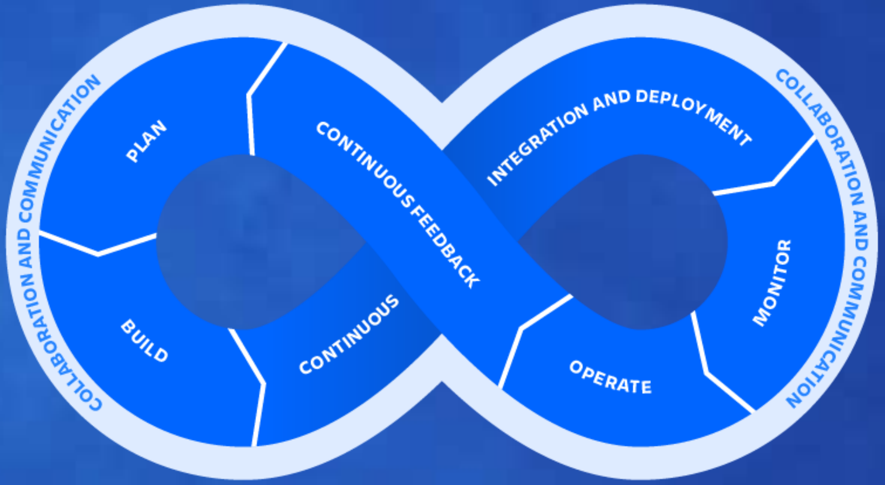

PLAN
CONTINUOUS FEEDBACK
INTEGRATION AND DEPLOYMENT
MONITOR
OPERATE
CONTINUOUS
BUILD
COLLABORATION AND COMMUNICATION
COLLABORATION AND COMMUNICATION

Session 5. Maven Overview & Release Management

Ram N Sangwan

Agenda

- DevCS Support for Maven
- Continuous Integration & Continuous Delivery (CI/CD)
- Release Management in DevOps
- Role of Release Manager in Software Delivery
- DevCS support for Release Management

What is Maven?

- A Java project management and integration build tool.
- Based on the concept of XML Project Object Model (POM).
- Originally developed for building Turbine.
- A small core with numerous plugins (in Jelly).

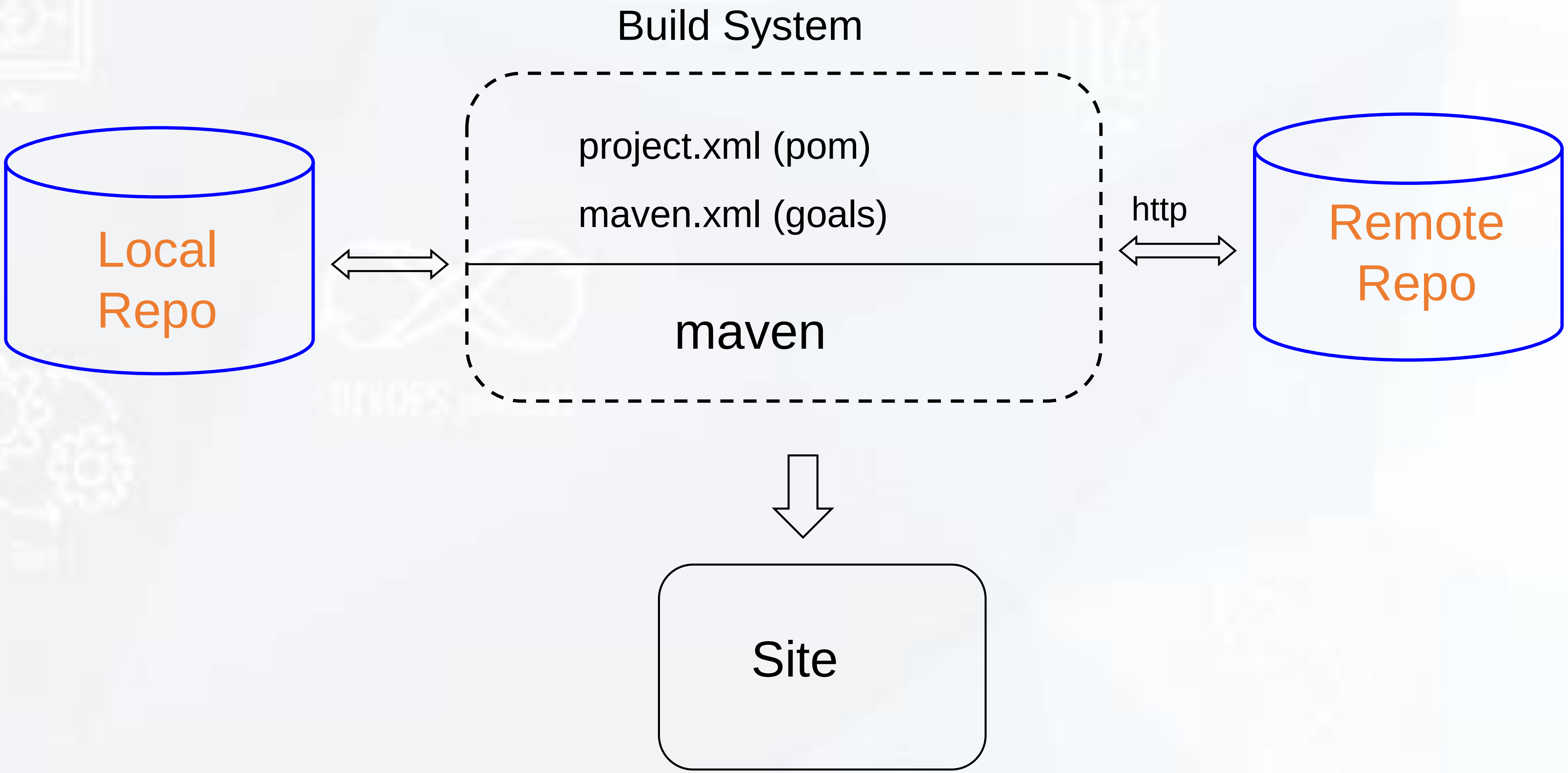
Apache Maven

Maven’s primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time.

In order to attain this goal, there are several areas of concern that Maven attempts to deal with:

- Making the build process easy
- Providing a uniform build system
- Providing quality project information
- Providing guidelines for best practices development
- Allowing transparent migration to new features

Architecture Overview



Maven – POM File

- POM (Project Object Model) is an XML file that contains information about the project and configuration details used by Maven to build the project i.e. sourcecode location, project dependencies etc.
- This file must be named as pom.xml and placed under root folder of project.
- When executing a task or goal, maven reads the POM, gets the needed configuration information, then executes the goal.

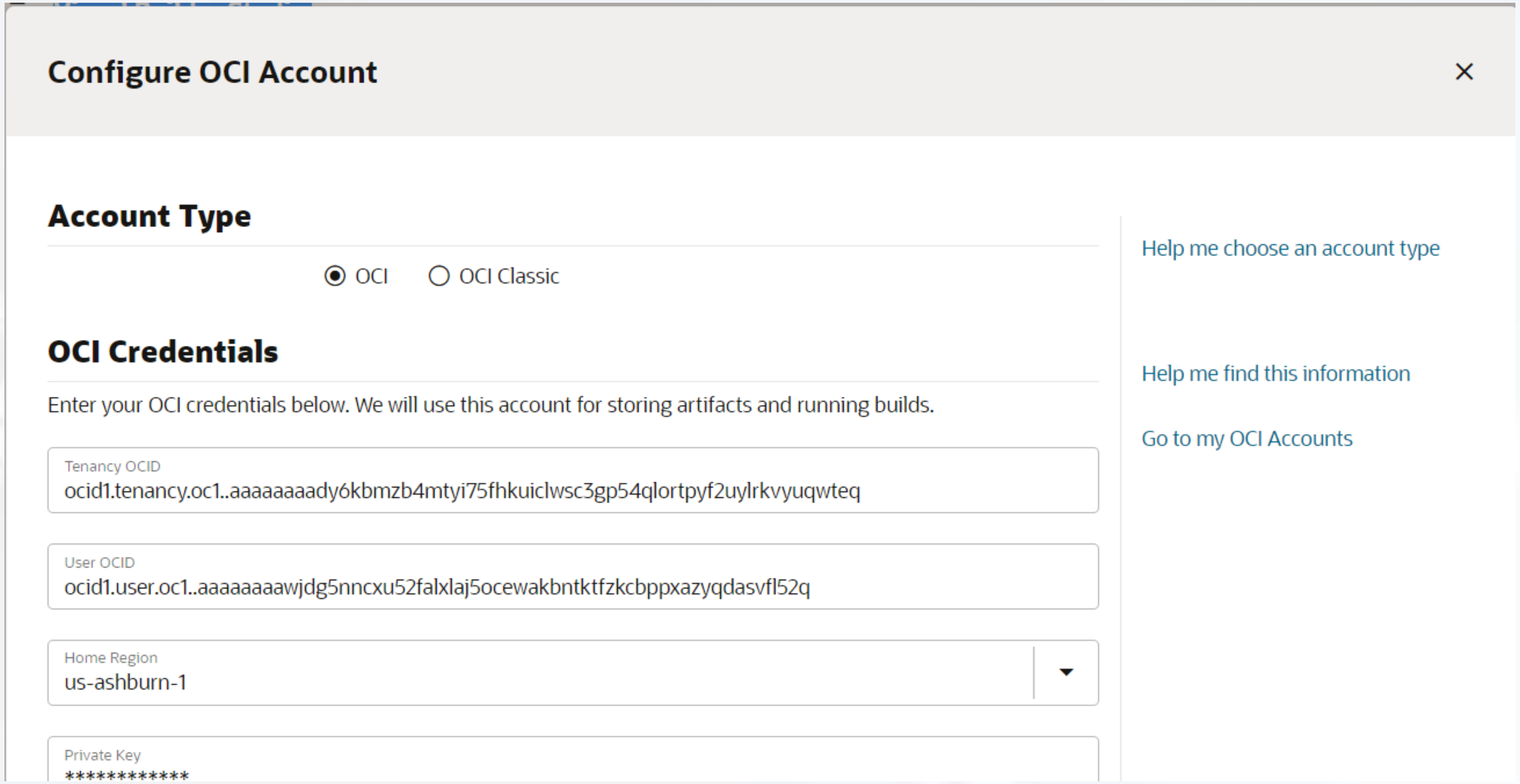
Default POM Configuration - Example

For a quickstart project, this is default generated pom.xml file.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd;
  <modelVersion>4.0.0</modelVersion>
   <groupId>com.howtodoinjava.demo</groupId>
  <artifactId>MavenExamples</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
   <name>MavenExamples</name>
  <url>http://maven.apache.org</url>
   <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
   <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

DevCS Support for Maven

- To includes Maven repository for storing build dependencies and artifacts, first step is to connect Visual Builder Studio to your OCI Account.



Configure OCI Account ×

Account Type

● OCI ○ OCI Classic

Help me choose an account type

OCI Credentials

Enter your OCI credentials below. We will use this account for storing artifacts and running builds.

Help me find this information

Go to my OCI Accounts

Tenancy OCID: ocid1.tenancy.oc1..aaaaaaaady6kbmzb4mtyi75fhkuiclwsc3gp54qlortpyf2uylrkvyuqwteq

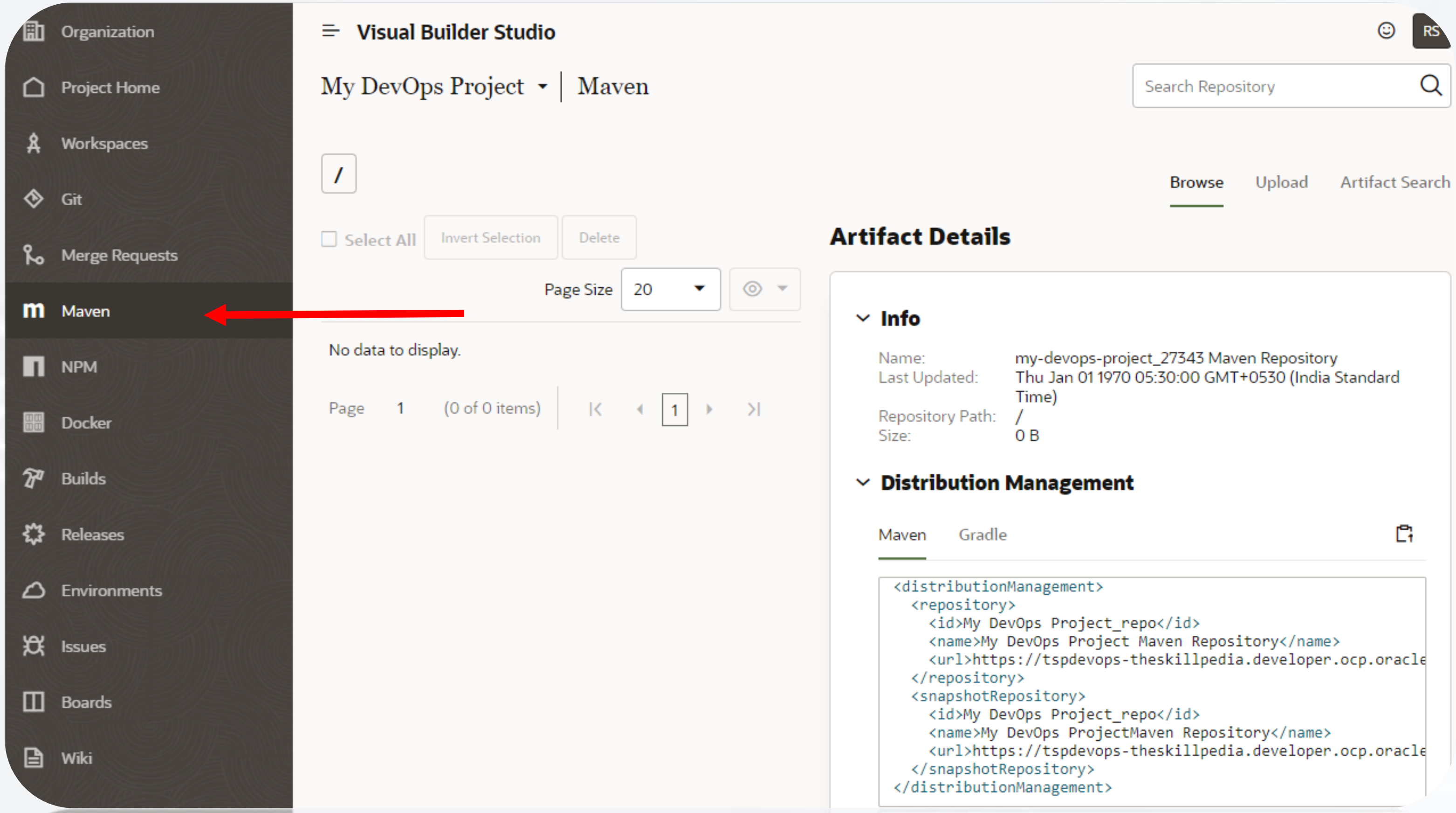
User OCID: ocid1.user.oc1..aaaaaaaawjdg5nncxu52falxlaj5ocewakbntktfzkcbppxazyqdasvfl52q

Home Region: us-ashburn-1

Private Key: ************

DevCS Support for Maven

- Oracle Developer Cloud Service (DevCs) includes Maven repository for storing build dependencies and artifacts.



Artifact Details

Info

Name: my-devops-project_27343 Maven Repository
Last Updated: Thu Jan 01 1970 05:30:00 GMT+0530 (India Standard Time)
Repository Path: /
Size: 0 B

Distribution Management

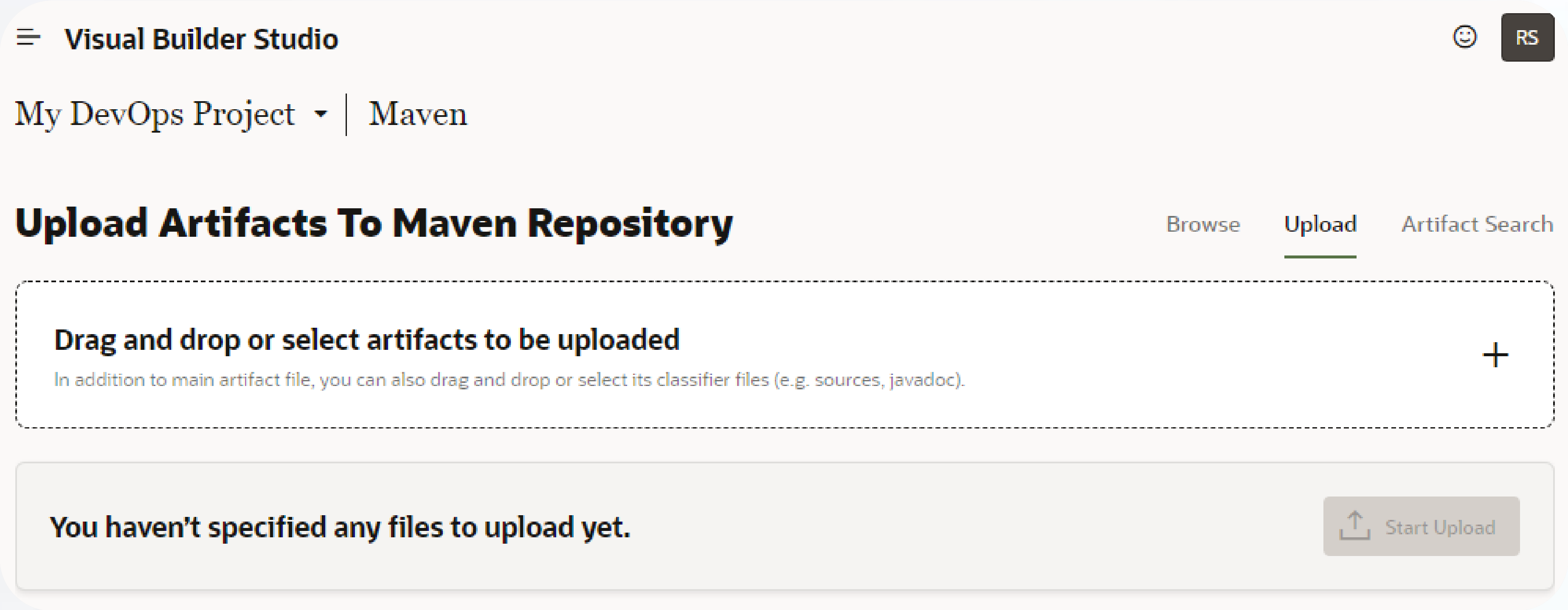
```
<distributionManagement>
  <repository>
    <id>My DevOps Project_repo</id>
    <name>My DevOps Project Maven Repository</name>
    <url>https://tspdevops-theskillpedia.developer.ocp.oracle
  </repository>
  <snapshotRepository>
    <id>My DevOps Project_repo</id>
    <name>My DevOps ProjectMaven Repository</name>
    <url>https://tspdevops-theskillpedia.developer.ocp.oracle
  </snapshotRepository>
</distributionManagement>
```

DevCS Support for Maven

- Maven artifacts can be uploaded to the repository after they are created by your build.
- Dependencies can be manually uploaded ahead of time and pulled into the build jobs that need them.

Upload Files

- Click the Upload button.
- Drag and drop files into the box or click “select artifact files” and select the files.



Upload Artifacts To Maven Repository

Drag and drop or select artifacts to be uploaded

In addition to main artifact file, you can also drag and drop or select its classifier files (e.g. sources, javadoc).

You haven’t specified any files to upload yet.

Select Version and Other Details.

- Let us say the project uses Oracle JET version 6.2.0 so add it to Maven repository.

(Download a Demo project from http://www.oracle.com/webfolder/technetwork/jet/public_samples/FixItFast.zip)

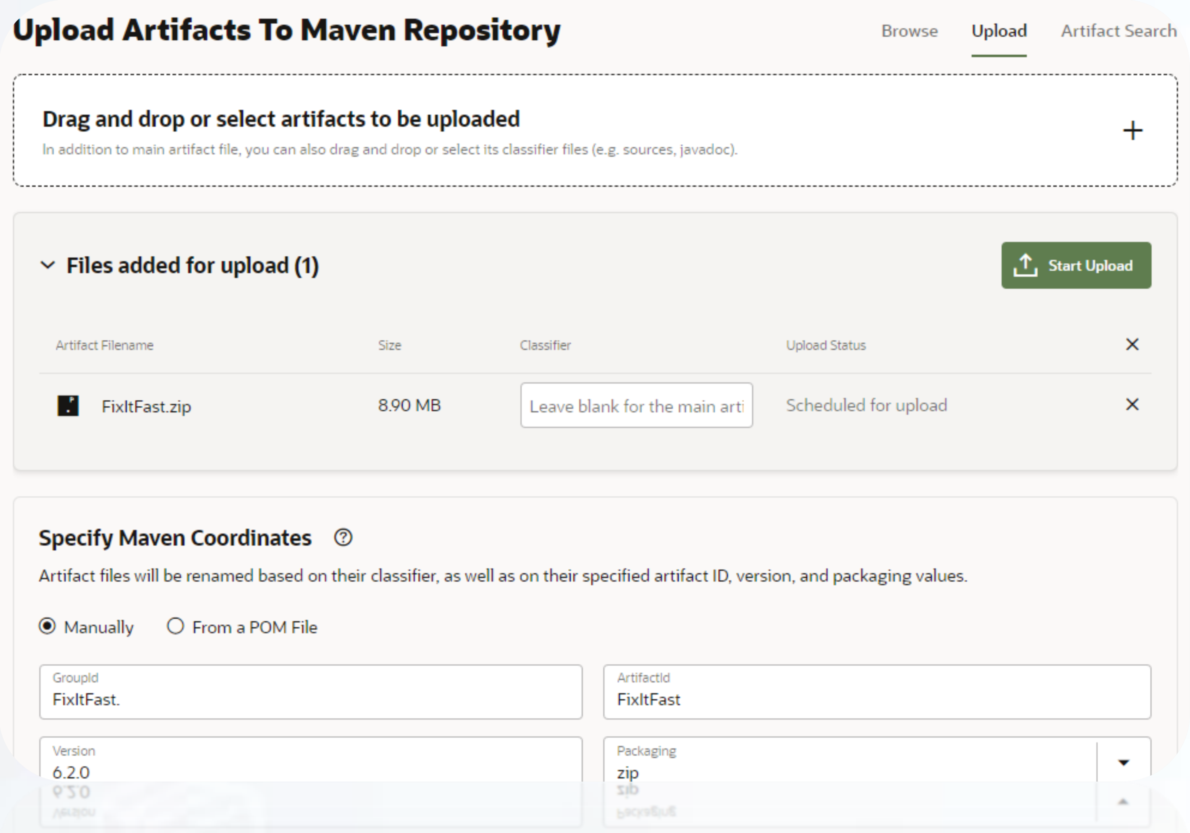
- Populate the file data.

  GroupId: FixItFast

  ArtifactId: FixItFast

  Version 6.2.0

  Packaging: zip
- Click the Start Upload button.
- Now that we have a file in the repository, we can use it in our build.



Upload Artifacts To Maven Repository

Drag and drop or select artifacts to be uploaded

In addition to main artifact file, you can also drag and drop or select its classifier files (e.g. sources, javadoc).

Files added for upload (1)

Artifact Filename	Size	Classifier	Upload Status
FixItFast.zip	8.90 MB	Leave blank for the main arti	Scheduled for upload

Specify Maven Coordinates

Artifact files will be renamed based on their classifier, as well as on their specified artifact ID, version, and packaging values.

● Manually ○ From a POM File

GroupId: FixItFast.

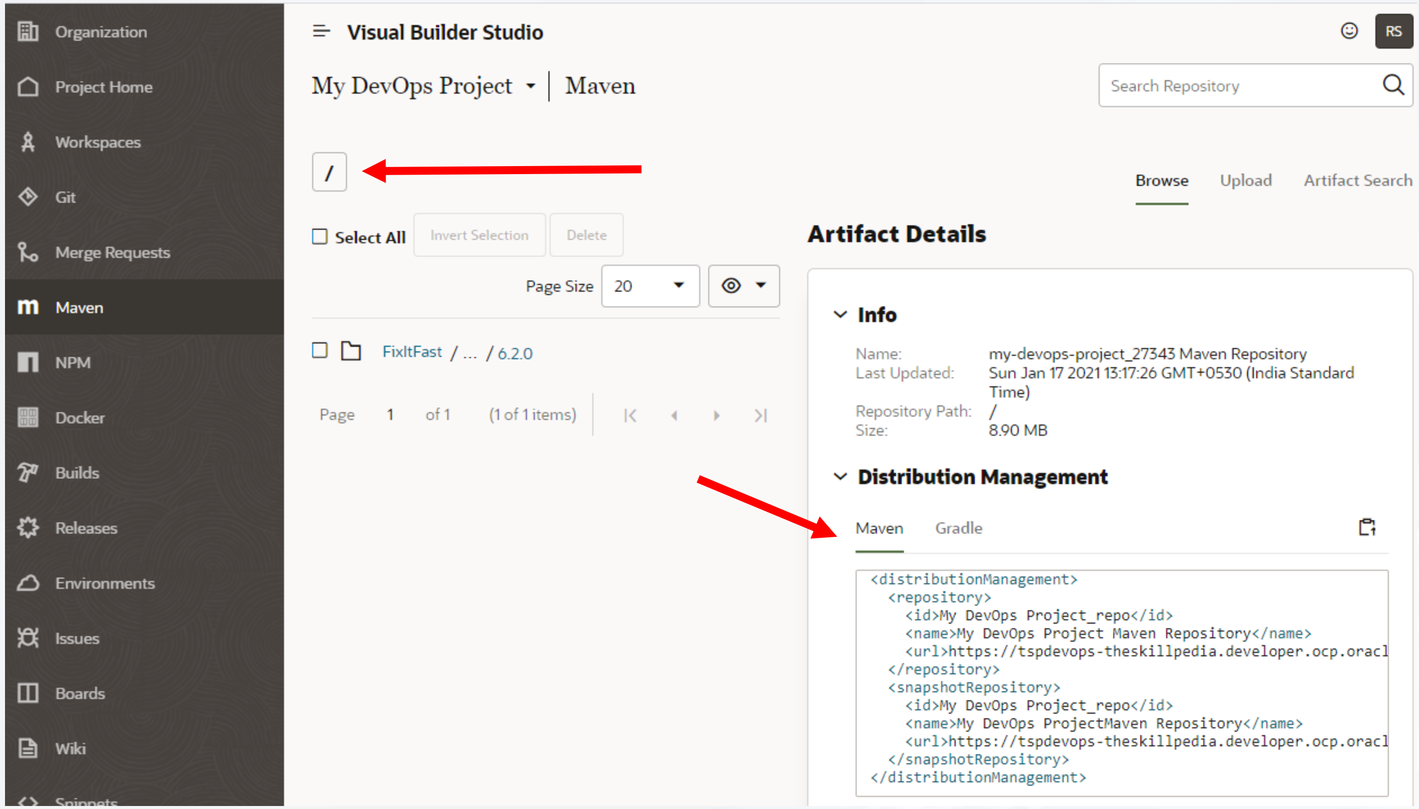
ArtifactId: FixItFast

Version: 6.2.0

Packaging: zip

Use a Build Tool

- If you're using a build tool such as Maven or Gradle, you can get the dependency declaration for the Maven repository by clicking on the root of the project then copying it from the dependency declaration tab for your tool.

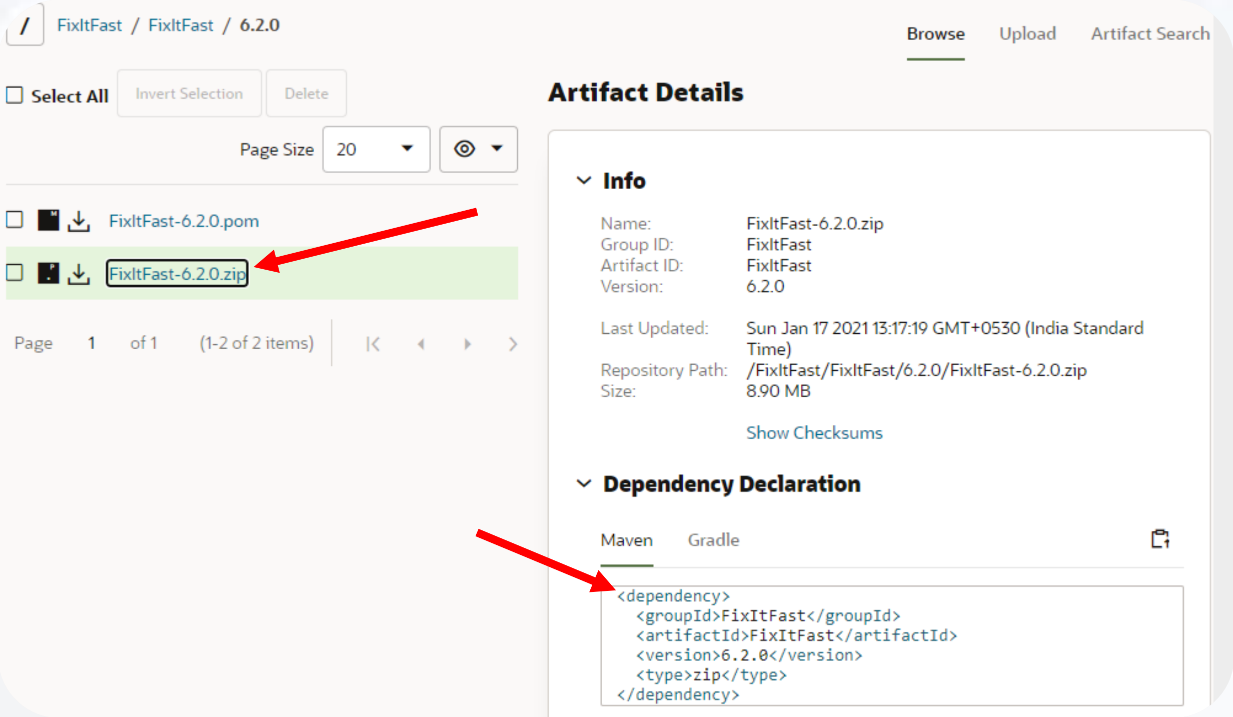


Name: my-devops-project_27343 Maven Repository
Last Updated: Sun Jan 17 2021 13:17:26 GMT+0530 (India Standard Time)
Repository Path: /
Size: 8.90 MB

```
<distributionManagement>
  <repository>
    <id>My DevOps Project_repo</id>
    <name>My DevOps Project Maven Repository</name>
    <url>https://tspdevops-theskillpedia.developer.ocp.oracl
  </repository>
  <snapshotRepository>
    <id>My DevOps Project_repo</id>
    <name>My DevOps ProjectMaven Repository</name>
    <url>https://tspdevops-theskillpedia.developer.ocp.oracl
  </snapshotRepository>
</distributionManagement>
```

Get File Details

- To get the file information:
  - Drill down into the folder.
  - Click on the file you need.
  - Copy the dependency declaration for the file.



Artifact Details

Info

Name: FixItFast-6.2.0.zip
Group ID: FixItFast
Artifact ID: FixItFast
Version: 6.2.0

Last Updated: Sun Jan 17 2021 13:17:19 GMT+0530 (India Standard Time)
Repository Path: /FixItFast/FixItFast/6.2.0/FixItFast-6.2.0.zip
Size: 8.90 MB

Show Checksums

Dependency Declaration

Maven Gradle

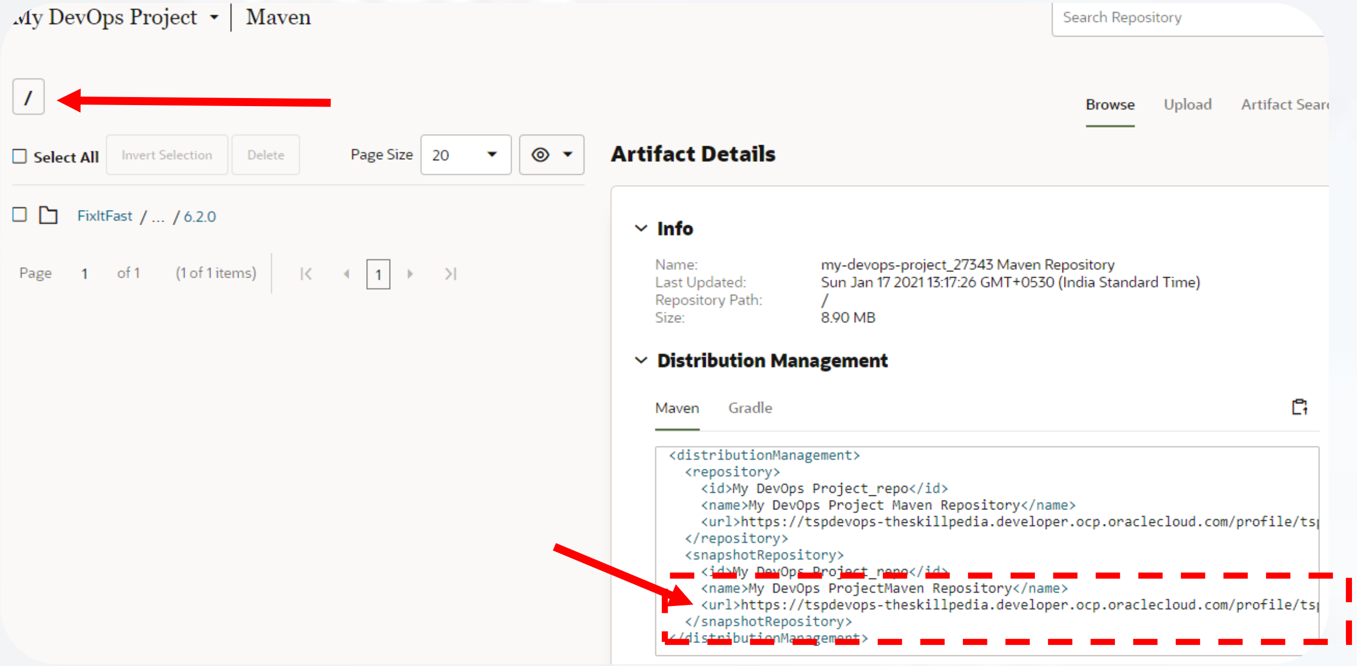
```
<dependency>
  <groupId>FixItFast</groupId>
  <artifactId>FixItFast</artifactId>
  <version>6.2.0</version>
  <type>zip</type>
</dependency>
```

Run the Build Tool

- At this point, you can run your build tool in a build job as normal.
- But if you’re not using a build tool there is another way.

cURL

- To use a tool such as cURL you need to get the URL for the file.
- First, we’ll need the DevCs Maven repository URL for the project.
  - Click /
  - Copy URL from the dependency declaration panel



Name: my-devops-project_27343 Maven Repository
Last Updated: Sun Jan 17 2021 13:17:26 GMT+0530 (India Standard Time)
Repository Path: /
Size: 8.90 MB

```
<distributionManagement>
  <repository>
    <id>My DevOps Project_repo</id>
    <name>My DevOps Project Maven Repository</name>
    <url>https://tspdevops-theskillpedia.developer.ocp.oraclecloud.com/profile/ts
  </repository>
  <snapshotRepository>
    <id>My DevOps Project_repo</id>
    <name>My DevOps ProjectMaven Repository</name>
    <url>https://tspdevops-theskillpedia.developer.ocp.oraclecloud.com/profile/ts
  </snapshotRepository>
</distributionManagement>
```

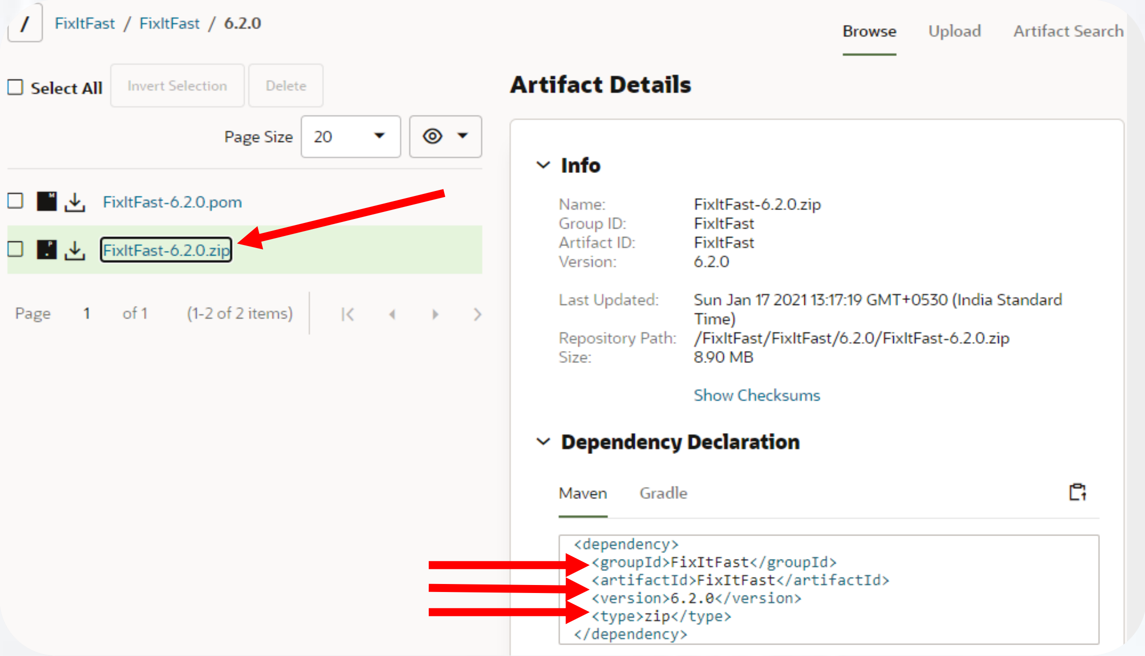
The repository URL

- The repository URL will be similar to :

https://tspdevinst-theskillpedia.developer.ocp.oraclecloud.com/profile/tspdevinst-theskillpedia/s/tspdevinst-theskillpedia_my-devops-project_28344/maven/

Find the Data for a File

- This is the same data we entered when we uploaded the file.
- If you need to find the data for a file:
  - Drill down into the folder.
  - Click on the file you need.
  - Copy the information from the dependency declaration for the file.



Artifact Details

Info

Name: FixItFast-6.2.0.zip
Group ID: FixItFast
Artifact ID: FixItFast
Version: 6.2.0

Last Updated: Sun Jan 17 2021 13:17:19 GMT+0530 (India Standard Time)
Repository Path: /FixItFast/FixItFast/6.2.0/FixItFast-6.2.0.zip
Size: 8.90 MB

Show Checksums

Dependency Declaration

Maven Gradle

```
<dependency>
  <groupId>FixItFast</groupId>
  <artifactId>FixItFast</artifactId>
  <version>6.2.0</version>
  <type>zip</type>
</dependency>
```

Use cURL in the Build Job

- The different components in DevCs already have the permissions to access each other in place so the build job can simply cURL from the Maven repository.
- Add an “Execute Shell” build step and add the following (with your repository URL):

```
curl -O https://tspdevinst-theskillpedia.developer.ocp.oraclecloud.com/profile/tspdevinst-theskillpedia/s/tspdevinst-theskillpedia_my-devops-project_28344/maven/FixItFast/FixItFast/6.2.0/FixItFast-6.2.0.zip
```

- You can use any of the cURL features as you normally would, for example, to get multiple files in one call.

Why Store Dependencies in the Project?

- Using the DevCs Maven Repository to store dependencies has many benefits, including:
  - It will be faster than pulling dependencies across the internet.
  - You won’t have to worry about a remote repository being offline or files being removed.
  - Helps protect you from security problems if a remote repository is compromised.

Role of Release Manager in software delivery

- The Release Manager reports to the manager in the Release Management team.
- Release Manager, in DevOps, is responsible for scheduling, planning, and controlling the software's development and delivery process.
- As a Release Manager, you will be responsible for handling the DevOps team to make them deliver the services on time and will manage both IT operations and developers.
- You will be responsible for the Release Management lifecycle that involves the stages such as scheduling the release, coordinating between teams, and deployment of release as per the schedule and within budget.

Release Manager in software delivery

- As a DevOps Release Manager, you will have to release the software after completing the testing stage and deployment stage, and work closely with the application development team, testing team, and production team.
- You will have to maintain proper coordination between these teams to update the project related information.
- You need to define the strategic usage of release management tools to manage the revenue for the release.
- Overall, you will be concerned with planning, testing, tracking, release, deployment, communication, and risk management.

Release Manager Activities

- Plan the release of project deliverables and release life cycle.
- Communicate the project-related tasks such as plans, timelines, requirements, etc. between different teams.
- Coordinate the release schedule and resources required depending upon the third-party applications, defect backlogs, planned releases, and infrastructure updates.
- Identify the risks that can delay the release and manage them, such that the scope scheduled, and quality of the release is not affected.
- Track the progress and find issues, if any. Always work to improve the process of release.
- Make sure that the release is planned, according to the requirements and budget.
- Schedule the release readiness reviews before deployment and milestone reviews after each release.

Release Manager Activities

- Create plans for the implementation and deployment as per the release schedule.
- Plan and give weekly updates on the release activities
- Make sure the allocation of Release Engineers to every release.
- Communicate with release managers from different IT departments.
- Lead the Go-Live activities to deploy the software successfully.
- Team up with relevant development teams responsible for building the automation tools used to develop and deploy the software.
- Schedule the CAB meetings to discuss the release schedules with the team and find roadblocks, if any.
- Maintain documentation related to procedures on build and release, various notifications lists, and dependencies.

DevCS Support for Release Management

- For a release, you can specify tags or branches of Git repositories with stable code, artifacts of project Maven repository, build artifacts of stable builds, and binary files.
- For example, you can you create a release titled *V18-Q1* to mark stable code files, artifacts, and binaries of your application for the first quarter release of 2018 release. Project users then won't have to look around or ask which Git repository or branch has the stable code. They can then download Git repository archives and other artifacts of the *V18-Q1* release from the Release page itself.
- You can access and manage releases from the **Releases** page. When a project user opens a release, the user can download source snapshots of a specified branch or tag of the project Git repository, artifacts from the project Maven repository, specified binaries, and archived build artifacts.

Create a Release

- When you create a release, you specify the build artifacts, Git repositories and branches, and Maven artifacts.

Action	How To
Create a release	1. In the navigation bar, click Releases
	2. Click + Create Release.
	3. In Name and Description, enter a release name and description.
	4. In Status, specify the status of the release.
	5. Add the artifacts.
	6. In Notes, enter the release notes in the Page Text tab. Preview the notes in the Preview tab. You can use the project’s wiki markup language to format the notes.
	7. Scroll to the top of the page and click Save.
Clone a release	1. In the navigation bar, click Releases
	2. Select the release that you want to edit or clone, click Actions and then select Clone.
	3. In Name and Description, enter a release name and description.
	4. In Status, specify the status of the release.
	5. Add, update, or remove the artifacts.
	6. In Notes, enter the release notes in the Page Text tab. Preview the notes in the Preview tab.
	You can use the project’s wiki markup language to format the notes.
	7. Scroll to the top of the page and click Save.

Thank You