[​docx icon DevOps guide](https://uppwise.sharepoint.com/:w:/s/DevTeam/ES4wN5RC06lLmcFwPWF58ZYBTPNZPYsXXkv-9vXNcu579Q?e=WJhVDw)

**SPM PIPELINES AND RELEASES**

11/4/2022

Page Break

Contents

￼

Page Break

Pipelines

A pipeline is a series of build steps for a project or a solution. The output of a pipeline is an *artifact*. For our purposes an artifact is basically a package with the build output of a project (e.g., SPM Web app).

How pipelines are organized

For pipelines we consider the following three source branches:

* D – develop
* H – hotfix
* M – master

The projects involved for SPM pipelines are:

* SPM Web App – TT
* SPM Web App – PO
* SPM Reporting API
* SPM Business layer API
* SPM Analytics Web APP
* SPM Analytics API

For each branch and for each project there is a pipeline configured. Pipelines are organized in folders representing the branch (e.g., D – develop). The naming convention <branch letter> - <project name> is taken.

|  |  |  |  |
| --- | --- | --- | --- |
| **Branch** | **Repository** | **Pipeline** | **.yaml file** |
| <B> | SPMAnalytics | <B> - SPMAnalytics - Api | azure-pipelines – Api.yml |
| <B> - SPMAnalytics - App | azure-pipelines - App.yml |
| SPMBusinessLayerApi | <B> - SPMBusinessLayerApi - PO Api | azure-pipelines.yml |
| SPMReporting | <B> - SPMReporting - Api | azure-pipelines.yml |
| SPMWebApp | <B> - SPMWebApp - PO | azure-pipelines - PO.yml |
| <B> - SPMWebApp - TT | azure-pipelines - TT.yml |

Pipeline steps are defined through .yaml configuration files.

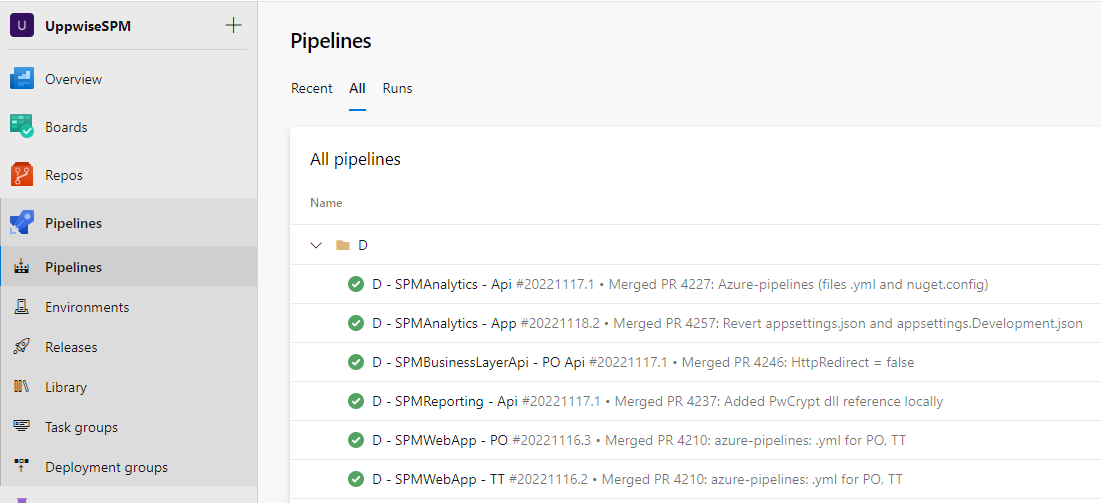
.yaml files are stored at branch level. They are not included in solutions or projects.

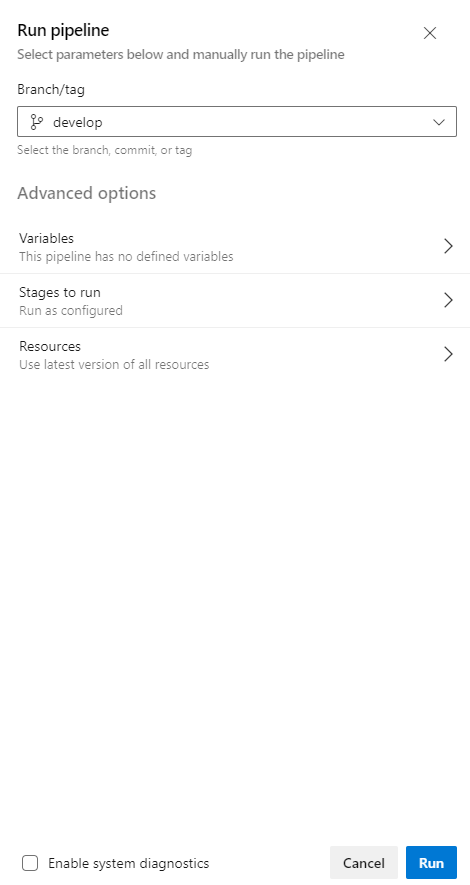
.yaml files should not be changed outside changes to pipelines if needed. Pipelines managers are owners of .yaml files.

When a pipeline runs on a given branch, the steps to build are defined by the .yaml file configured for the pipeline and stored into the given branch.Page Break

How to run a pipeline

To run a pipeline, consider the [Azure DevOps site](https://dev.azure.com/uppwise/UppwiseSPM) for SPM and navigate to *Pipelines*.



Consider the branch and the project you want to build and click on the right “…” and choose *Run pipeline.*

The branch selected should be the default one for that pipeline already configured so is not necessary to change it (in this picture the branch is demonstrative).

No need to set different choices than the default ones, click *Run.*

When the pipeline is completed (it can take some minutes), the pipeline artifact is stored at pipeline level and is ready to use for deployment purposes. No other actions are needed at this level.

Pipeline triggers

Now no automatic triggers are configured for pipelines. CI (*continuous integration*) is disabled for all pipelines, so any change to the source code will not trigger a new pipeline to run.

A new pipeline is immediately queued after a manual request is created. One build agent (default azure agent) is available so, if multiple pipelines are requested, their execution will be one at time.Page Break

Releases

A release is a set of steps that takes as input a pipeline artifact (the output of a build) and executes deploy operations on target servers (in SPM case on target VM on azure).

Releases are composed of *stages*. Each stage represents a component (e.g., reporting APIs) on an environment (e.g., staging, production…).

Each stage is composed of steps to do the necessary deploy operations. The main operations involved in each stage are:

1. Download the necessary artifacts
2. Execute some IIS configurations, if needed (e.g., create an application pool)
3. Execute *variables substitution*
4. Copy the built content to remote deploy folder

All these operations are already configured and ready to be used when a release is run.

How releases are organized

Releases are logically grouped as below:

**SPM Web Application**

* SPM PO Web App
* SPM TT Web App
* SPM Business Layer API

**SPM Analytics**

* Analytics App
* Analytics API
* Analytics Reporting API

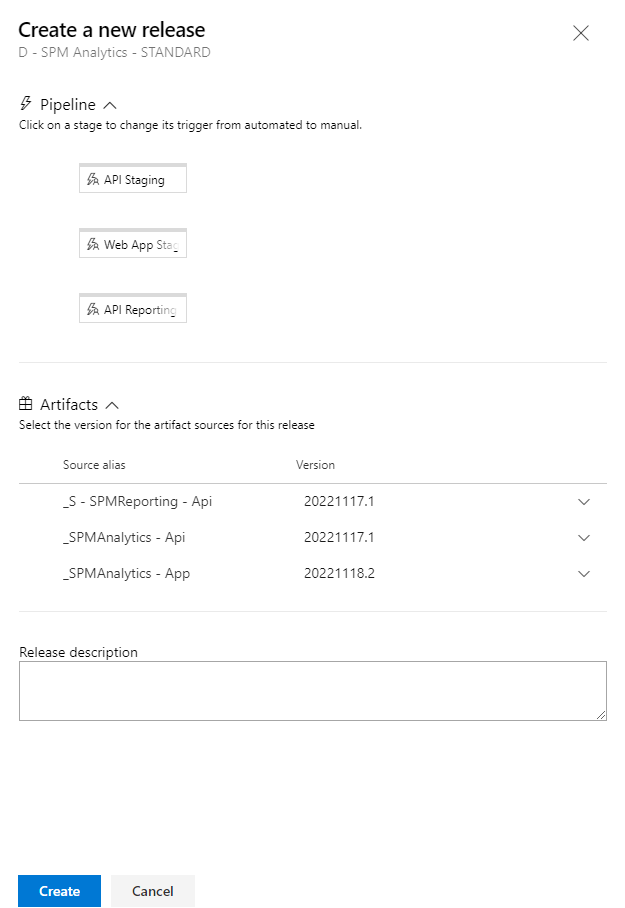
Each of the above groups of components is configured as a release for an environment:

* D - <release name> for D branch
* H - <release name> for H branch
* M - <release name> for M branch

How to run a release

To run a release, consider the [Azure DevOps site](https://dev.azure.com/uppwise/UppwiseSPM) for SPM and navigate to *Releases*.

Select the release needed (e.g., D – SPM is the develop related SPM Web Application release) and click on *Create release* on top right of the screen.



* All the stages will be created for deployment, but the release will not start immediately.
* The latest version of each artifact is selected by default, change is possible
* A release description is possible and useful as an historical information

Click on *Create* *release* in order to create the release.

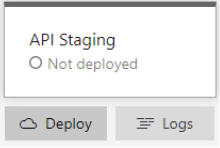
Page Break

Release triggers

Two main possibilities are possible for a release to run.

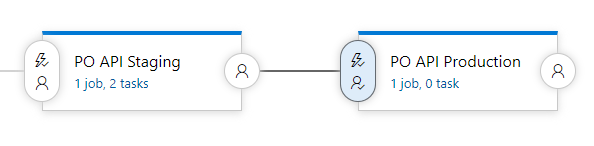
1. After a release is created manually (i.e., *Create release* button), the release will be created. After this happens, a user should select the option “Deploy” for the stage/s she he wants to deploy.
2. After a pipeline (builds) is concluded, automatically a new release for the linked artifact is created. After this happens, a user should select the option “Deploy” for the stage/s she/he wants to deploy.

To select the option “Deploy” go to the specific stage and click on the “Deploy” button.



Approvals for a production environment

For production stages approvals are required. Approvals are people configured at stage level required to approve the deploy of the given stage.



The picture above is a demonstrative example of how approvals can be configured.

The last stage will not be done until an explicit approve action is confirmed by specific people.

Page Break

Variable substitution

Variable substitution is a step of each stage. With variable substitution the automated release overwrites all specific environment keys and values of the configuration files.

Variable substitution works on both Web.config files and appSettings.json files.

Each stage of a release has an associated set of variables in the form of *key – value* pair.

 Each key corresponds to a key contained in a configuration file of the downloaded artifact:

* For Web.config files the considered keys are these of *ConnectionStrings* tag and *AppSettings* tag.
* For AppSettings.json files all keys are considered

When running a release, no action is required for variable substitution. All variables are already configured at release – stage level.

The release owner should maintain the variable values at release level up to date.

So, it’s important that:

**When a new configuration key is added at source code level (development), a new configuration variable is needed at release level. The Dev team must always communicate these kinds of changes to release owners.**

Page Break

Retention policies

Retention policies are applied for the output of both pipelines and releases. These policies are configurable.

Currently the default retention policies are applied both to pipelines and releases.

Releases

Retention configuration is possible at stage level. Below is actual configuration:

* Days to retain a release: 30
* Minimum releases to keep: 3
* Retain associated artifacts: yes

As a release is retained, the linked artifacts are retained. Indeed, the build package linked to the release is always available.

Pipelines

Retention configuration is possible at project level. Below is actual configuration:

* Days to retain a pipeline: 30
* Minimum pipelines to keep: 3

As far as a release is retained and the option to keep linked artifact is flagged, the associated pipeline artifact and run is also retained.Page Break

Quick guide for a deployment

1. Consider the branch where the changes to source code are done (e.g., D)
2. Navigate to pipelines and select the proper pipeline D - <project name> and run it and wait to be completed ([Azure DevOps site](https://dev.azure.com/uppwise/UppwiseSPM))

Graphical user interface, text, application, email

Description automatically generated

1. Navigate to releases and select the proper release D - <release>

Graphical user interface, text, application

Description automatically generated

1. Navigate to the release details of the release just created automatically by the previous pipeline (click on its link)
2. Go with mouse to the stage you want to deploy and click “Deploy”. Do it for all stages you want to deploy

Graphical user interface, text, application, chat or text message

Description automatically generated

1. If approval step is needed approve it or notify the approval request to the approval user