Pipeline workflows

The pipelines are YAML based pipelines and are using an updated build-agent pool in Azure Devops. At this time of writing, they are utilizing the Microsoft hosted build agent pool and it contains most of the dependencies we need (like kubectl, docker, helm and python). The only thing missing is `boto3` and `cdk` plus its dependencies, so you will see them appear in the pipeline where these are needed.

The general flow is that the pipeline YAML itself contains little to no logic unless it is required; most logic lies in external scripts. Like this it is easier to test specific changes locally, troubleshooting becomes simpler and the general pipeline logic becomes more modular and generic. It means that it will be easier to port over to a different system in the future, should that be necessary.

As mentioned in the top-level page, the main logic makes assumptions on parameters living in the SSM parameter store. At this time of writing the majority of the pipeline parameters needed to function live in the **dev/test** account, not the **build** account. The reason for this is practical more than anything; there is limited to no access to the build account and getting access to that is difficult. Access to the dev/test account is easier and therefore it makes it easier to modify the parameters where needed.

Within the *kubernetes-mendix-deployment* repository there is a `.ci` folder which houses Azure pipeline YAML files. There are three distinct files both the build release and deploy pipeline: `build_pipeline.yaml`, `release_pipeline.yaml` and `release_template.yaml`. The latter is, as its name implies, a template and contains the actual logic of deploying code. The release pipeline is basically an interface to this template and executes everything in the right order, based on its input parameters.

The two pipelines can run independently of each other; as in if you already have a container image that was previously built and you wish to redeploy it, you can run the release pipeline immediately without running the build pipeline. Conversely, if you have a new commit that must be deployed, the continuous integration functionality coming from `mendix-vcs-polling` will automatically trigger the build pipeline, though you can run this manually yourself as well of course. The build pipeline will trigger the release pipeline to release the built artifact to the dev/test environment, after which is will require manual approval for the other environments.

You can also run the release pipeline on specific environments; if you select the acceptance environment in the release pipeline the release pipeline will skip the dev/test environment and start at acceptance (and will also do production after approval). Of course, selecting production will only deploy to production.

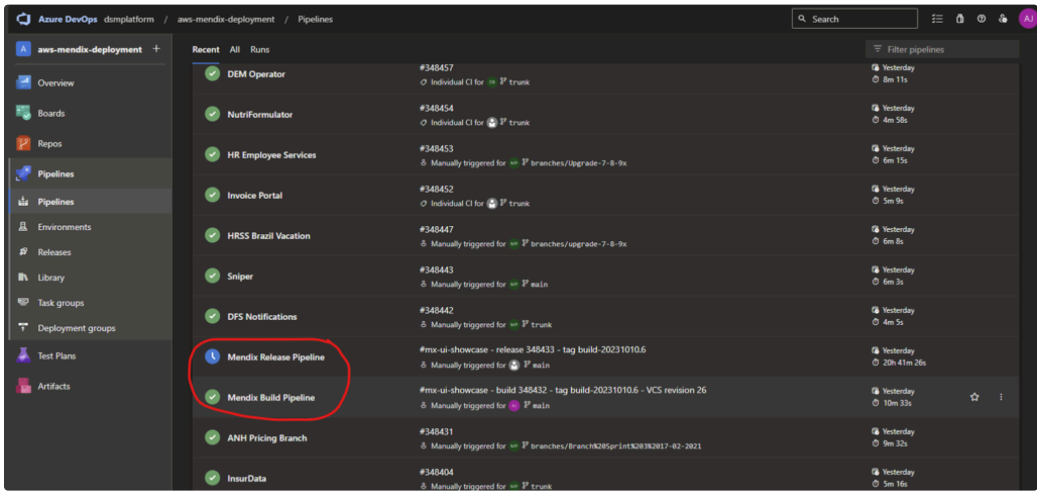
Technically it makes no difference if the build account is used at all or not; the old pipeline uses an old build agent based off Ubuntu 16; the new pipeline uses Ubuntu 22 and is Microsoft provided.

Please refer to the underlying pages of this page to zoom into the detail of the pipeline and its various execution models.

Running the build pipeline

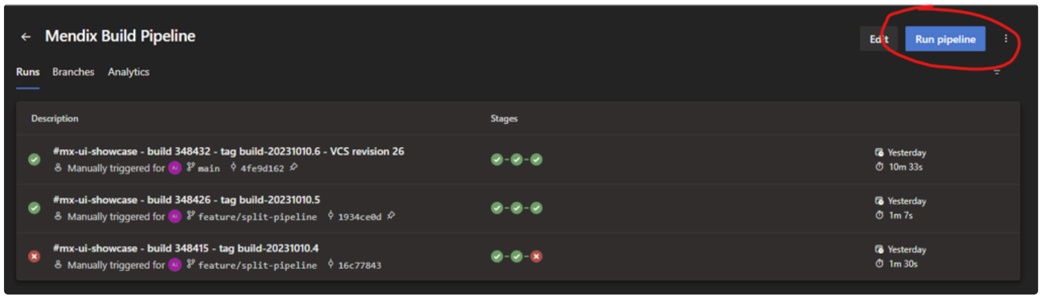
As mentioned above, there are two distinct pipelines, one for building and one for releasing, depending on what you do. In here we will look at the full cycle.

Navigate to https://dsmplatform.visualstudio.com/aws-mendix-deployment and click pipelines. Find the build pipeline here in this screen:



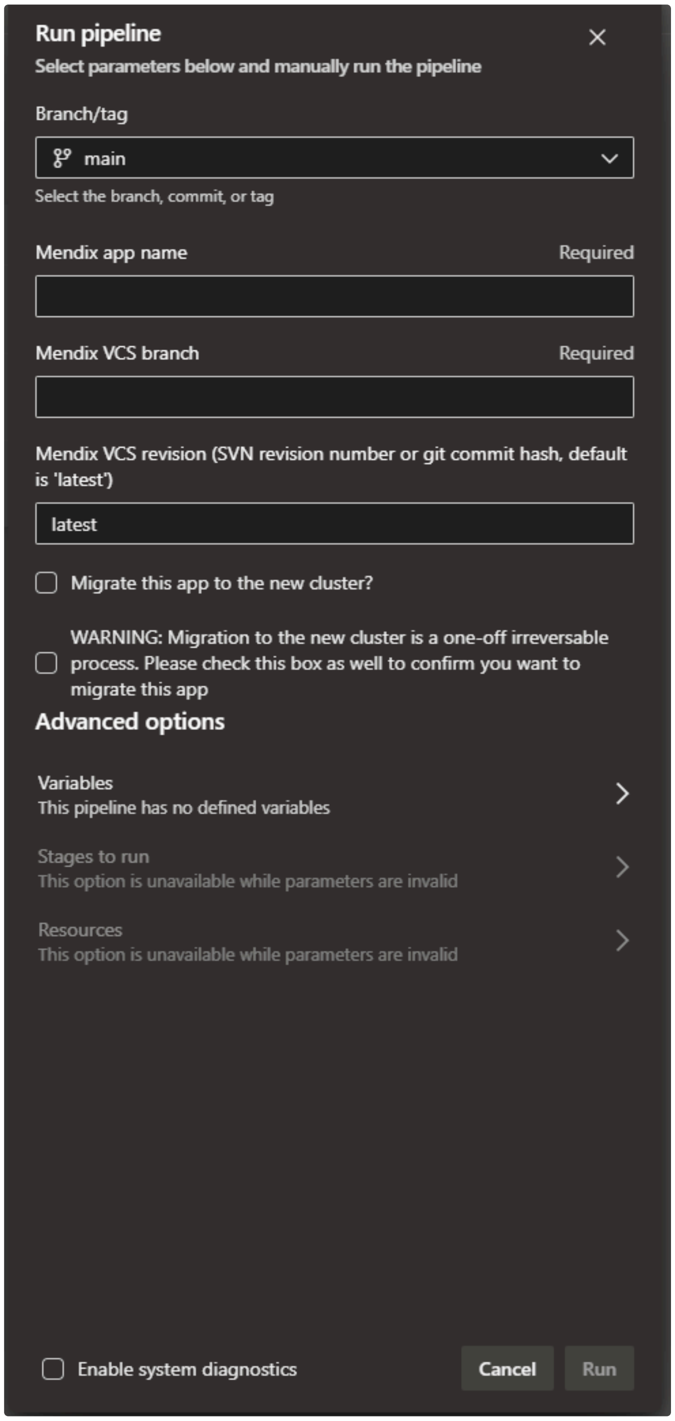
DEM Operator	#348457 Individual CI for trunk	Yesterday 8m 11s
NutriFormulator	#348454 Individual CI for trunk	Yesterday 4m 58s
HR Employee Services	#348453 Manually triggered for branches/Upgrade-7-8-9x	Yesterday 6m 15s
Invoice Portal	#348452 Individual CI for trunk	Yesterday 5m 9s
HRSS Brazil Vacation	#348447 Manually triggered for branches/upgrade-7-8-9x	Yesterday 6m 8s
Sniper	#348443 Manually triggered for main	Yesterday 6m 3s
DFS Notifications	#348442 Manually triggered for trunk	Yesterday 4m 5s
Mendix Release Pipeline	#mx-ui-showcase - release 348433 - tag build-20231010.6 Manually triggered for main	Yesterday 20h 41m 26s
Mendix Build Pipeline	#mx-ui-showcase - build 348432 - tag build-20231010.6 - VCS revision 26 Manually triggered for main	Yesterday 10m 33s
ANH Pricing Branch	#348431 Manually triggered for branches/Branch%20Sprint%203%2017-02-2021	Yesterday 9m 32s
InsurData	#348404 Manually triggered for trunk	Yesterday 5m 16s

In the new screen, you can click on **Run Pipeline**



Description	Stages	
#mx-ui-showcase - build 348432 - tag build-20231010.6 - VCS revision 26 Manually triggered for main 4fe9d162	✓-✓-✓	Yesterday 10m 33s
#mx-ui-showcase - build 348426 - tag build-20231010.5 Manually triggered for feature/split-pipeline 1934ce0d	✓-✓-✓	Yesterday 1m 7s
#mx-ui-showcase - build 348415 - tag build-20231010.4 Manually triggered for feature/split-pipeline 16c77843	✓-✓-✗	Yesterday 1m 30s

On the right hand side a modal pops up:



Run pipeline

Select parameters below and manually run the pipeline

Branch/tag: main

Select the branch, commit, or tag

Mendix app name — Required

Mendix VCS branch — Required

Mendix VCS revision (SVN revision number or git commit hash, default is 'latest'): latest

☐ Migrate this app to the new cluster?

☐ WARNING: Migration to the new cluster is a one-off irreversable process. Please check this box as well to confirm you want to migrate this app

Advanced options

Variables
This pipeline has no defined variables

Stages to run
This option is unavailable while parameters are invalid

Resources
This option is unavailable while parameters are invalid

☐ Enable system diagnostics

These are the pipeline parameters, and depending on what parameters you set there will be two distinct modes that it runs. Note that the top level parameters **“Branch/tag”** refers to the branch of the *kubernetes-mendix-deploy* repository that houses this YAML code, **not** the branch or tag of the upstream Mendix repository. Always keep the top level branch on `main` and only use the `Mendix VCS branch` parameter to set the branch you want.

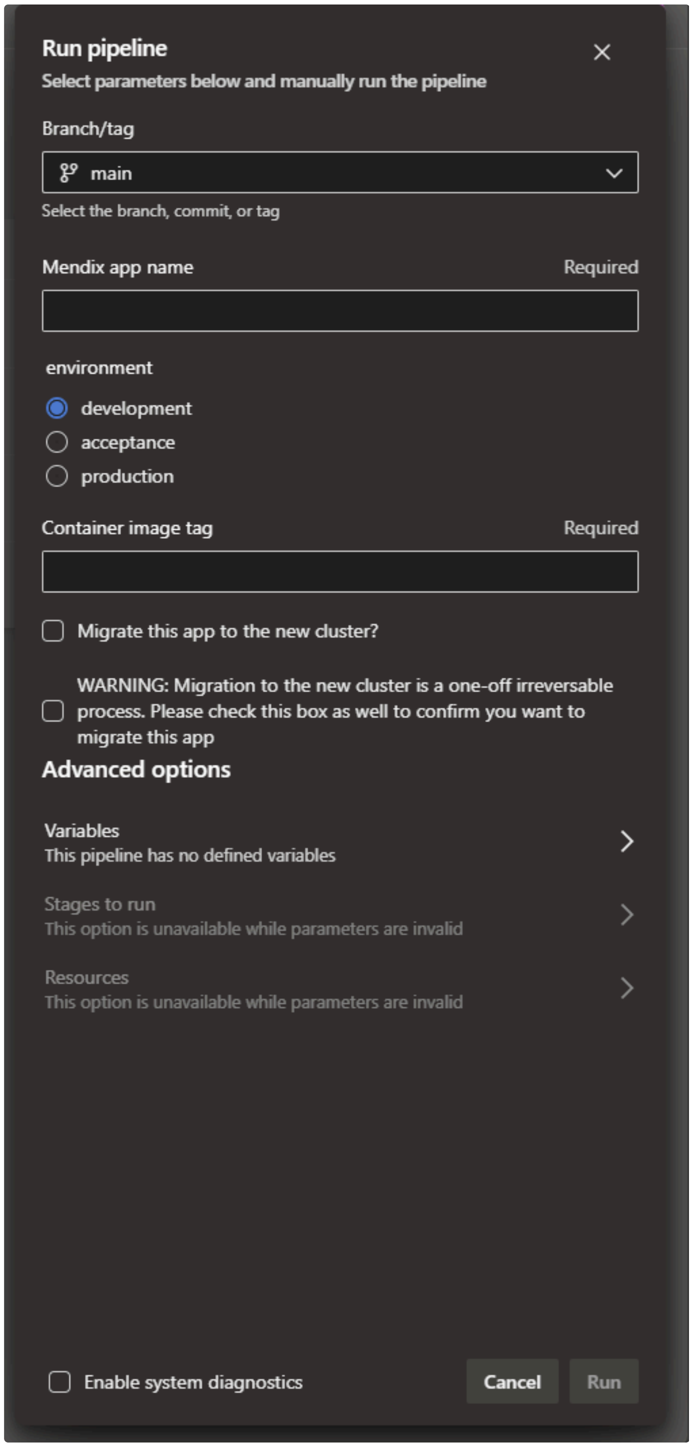
The modes that it can run at are:

- **Normal mode:** Just fill in the mendix application name and the branch (optionally a revision other than ‘latest’) and click run.
    - The Mendix application name **must** match with the naming convention. The easiest way to determine your application name is to go to the SSM Parameter Store in AWS and look for parameters that start with `/dsm/mendix/apps/<app_name>` - the value of <app_name> is what you need to provide as a Mendix application name in this field.
- **Migration mode:** Check the two checkboxes. If you don’t change anything it will build a new container, this is the safest option to choose.

Please refer to the subpages of this Confluence page for more information on the various modes.

Running the release pipeline

To run the release pipeline, find the `Mendix Release pipeline` and click Run Pipeline in the same way as you’d do for the build pipeline as described above. A lot is similar to the build pipeline so read that first before moving on; only the differences for the release pipeline are mentioned here.



Run pipeline

Select parameters below and manually run the pipeline

Branch/tag: main

Select the branch, commit, or tag

Mendix app name — Required

environment

- [✓ development]
- acceptance
- production

Container image tag — Required

☐ Migrate this app to the new cluster?

☐ WARNING: Migration to the new cluster is a one-off irreversable process. Please check this box as well to confirm you want to migrate this app

Advanced options

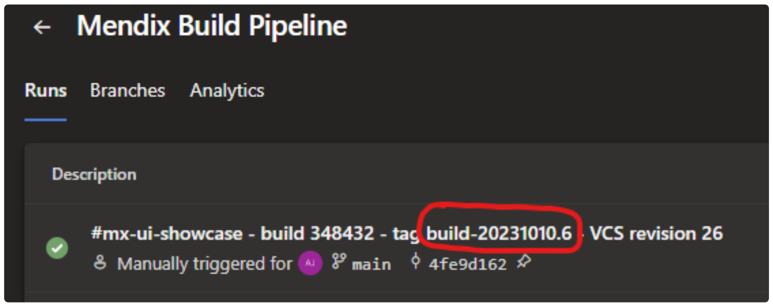
Variables
This pipeline has no defined variables

Stages to run
This option is unavailable while parameters are invalid

Resources
This option is unavailable while parameters are invalid

☐ Enable system diagnostics

In here, you are required to fill in the Mendix application name, which is the same as for the build pipeline, an environment on where to **start** the pipeline deployment (so selecting the acceptance environment will skip the development environment), and the container image tag. The container image tag is what has been produced by the build pipeline and can be found like this:


Mendix Build Pipeline

#mx-ui-showcase - build 348432 - tag build-20231010.6 - VCS revision 26

Manually triggered for main 4fe9d162

The container image tag in this case is **build-20231010.6** and this is what you fill in at the container image tag.