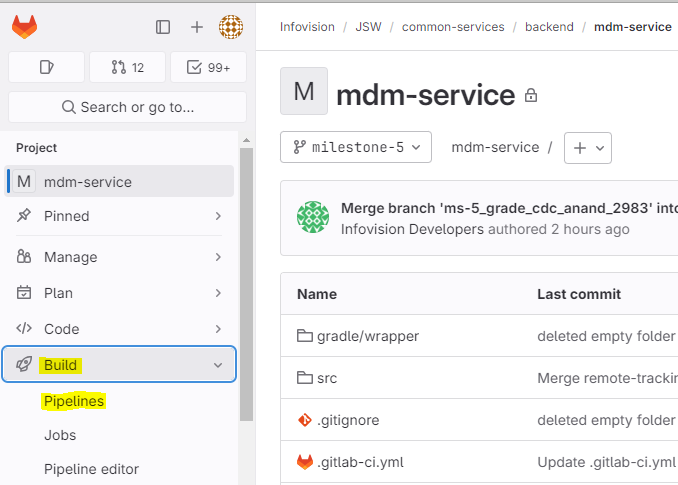
**BACKEND AND FRONTEND DEPLOYMENT PROCESS AND COMMANDS**

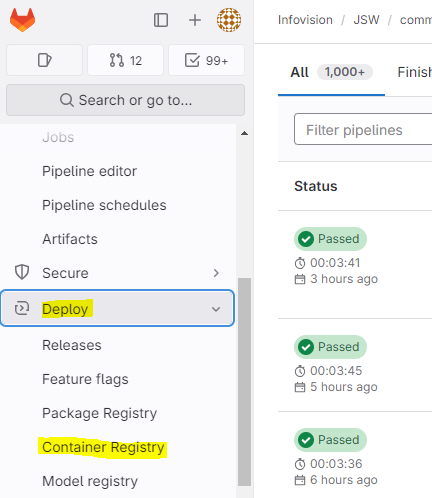
**BACKEND:**

Example scenario for mdm-service:

* After merging the code into the milestone 5 branch, check the pipeline status under Build -> Pipelines.



* If the pipeline is successful, go to Container Registry under Deploy tab and check whether the image is pushed or not.



* Under Container Registry, click on mdm-service and type "latest" in the search bar. You will see something like milestone-5-latest (published just now).
* This confirms that Docker built the image and pushed it to the GitLab Container Registry.
* Now, to deploy the mdm-service, use the command below:

deploy mdm-service:milestone-5-latest

* After deploying, check the list of all running containers using docker ps to verify whether the service is up and running..
* Validate the service through the Swagger URL in your browser.
* If milestone-5-latest doesn’t work or the latest changes are not reflecting, use the latest commit ID, which you can find in the Pipeline tab.

A screenshot of a computer

Description automatically generated

* If the latest tag doesn’t work, execute mdm-service with the commit ID.

deploy mdm-service:milestone-5-commitid

**example**: deploy mdm-service:milestone-5-26170f9d

* If the service is crashed or down, first check docker ps (you won't see the service in the list). To check the logs on the server,

cd /jsw/nextgen-mes/logs

ls

cd <servicename>

ls

cd mdm

ls

cat spring.log

* Here you can find the logs and errors with timestamps in UST.
* After solving the issue, redeploy the service with the same deploy command mentioned above.
* Follow the same process for all other backend services in both dev and QA environments.
* If you want to add variables for services, go to the link below:

[https://gitlab.com/infovision2023/devops/environment\_files/dev.list](https://gitlab.com/infovision2023/devops/-/blob/main/environment_files/dev.list?ref_type=heads)

* Click on the "Edit" button -> "Edit single file" and add the variables, then click "Commit changes."
* Also, add the variables in qa.list by navigating one folder back under "environment\_files."
* Login to the server and execute the following:
* Navigate to cd /root/.devops If you list the contents (ls), you'll see a file named deploy\_envscript.sh. Execute this file to download the updated dev.list file.

. / deploy\_envscript.sh (dot slash deploy\_envscript.sh)

* Redeploy the service to reflect the changes. Repeat the same process for the QA environment as well.

**FRONTEND:**

1. **mdm-mfe** **dev**:

* For the dev environment, follow the same process as backend. To deploy the frontend in the development environment, execute the command below:

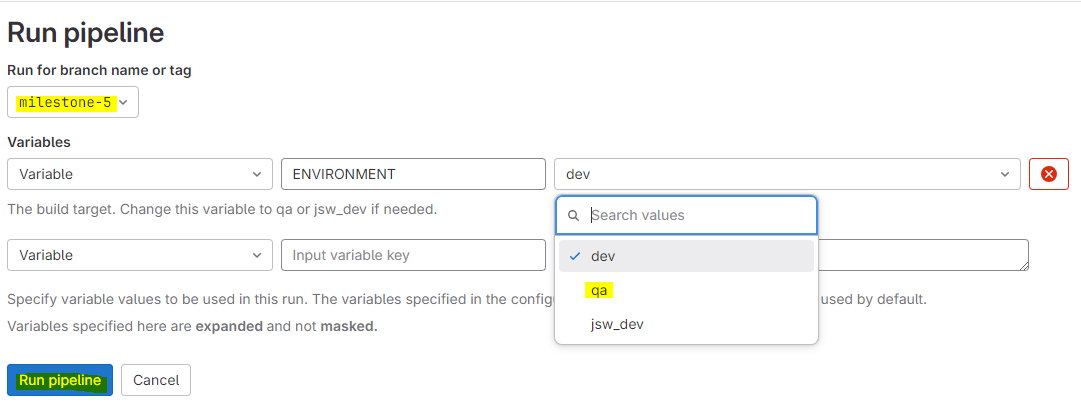
deploy mdm-mfe:milestone-5-dev-latest

**mdm-mfe** **qa**:

* To deploy to the QA environment, the pipeline needs to be manually triggered.

[https://gitlab.com/infovision2023/jsw/common-services/frontend/mdm-mfe/pipelines](https://gitlab.com/infovision2023/jsw/common-services/frontend/mdm-mfe/-/pipelines)

* Click on "Run pipeline", select the branch, and choose the QA environment from the variable dropdown. Then, run the pipeline.



* After the pipeline finishes, deploy the image to the QA server.

deploy mdm-mfe:milestone-5-qa-latest

* Check whether the application is up using docker ps and validate it through the browser.

1. **Rules-engine-mfe dev:**

* Before deployment to the dev and QA environments, first check docker ps

docker stop rules-engine

* Now, deploy the service with the following command:

docker run -d --rm=true --network host --name rules-engine -v /jsw/nextgen-mes/logs:/var/log registry.gitlab.com/infovision2023/jsw/common-services/frontend/rules-engine-mfe:milestone-5-dev-latestcommitid

* Please use the latest commit ID from the pipeline tab as the latest tag is not working currently.
* Validate the changes in the browser.

**QA environment:**

* In the QA environment, follow the same process as for mdm-mfe. Run the pipeline manually by selecting the required branch and QA variable. After the pipeline is successful, deploy the image to the QA server using the latest commit ID

docker run -d --rm=true --network host --name rules-engine -v /jsw/nextgen-mes/logs:/var/log registry.gitlab.com/infovision2023/jsw/common-services/frontend/rules-engine-mfe:milestone-5-qa-latestcommitid

* Validate the changes in the browser.
* Repeat the same process for reports-mfe and ivl-mdm

**Commands for reports-mfe:**

docker run -d --rm=true --network host --name reports -v /jsw/nextgen-mes/logs:/var/log registry.gitlab.com/infovision2023/jsw/common-services/frontend/reports-mfe:milestone-5-dev-latestcommitid

docker run -d --rm=true --network host --name reports -v /jsw/nextgen-mes/logs:/var/log registry.gitlab.com/infovision2023/jsw/common-services/frontend/reports-mfe:milestone-5-qa-latestcommitid

**Commands for ivl-mdm:**

docker run -it --rm=true --network host --name ivl-mdm -v /jsw/nextgen-mes/logs:/var/log registry.gitlab.com/infovision2023/jsw/common-services/frontend/ivl-mdm:phase1-dev-latestcommitid

docker run -it --rm=true --network host --name ivl-mdm -v /jsw/nextgen-mes/logs:/var/log registry.gitlab.com/infovision2023/jsw/common-services/frontend/ivl-mdm:phase1-qa-latestcommitid