AutoDevOps

Basic behavior

Let's have a look at this template project: Spring template project

You can see that there is only a `Dockerfile`. This is all you need to be able to use AutoDevOps and deploy your app on a kubernetes cluster.

```
# The application is going to run inside a container loaded with the `maven:3-jdk-8-alpine` image.
FROM maven:3-jdk-8-alpine

# We set the workdir inside the container
WORKDIR /usr/src/app

# We copy the content of the repository into this workdir
COPY . /usr/src/app

# We run the maven package goal
RUN mvn package

# Important! AutoDevOps expect the exposed port of the app to be 5000 and will run an livecheck on this port.
ENV PORT 5000
EXPOSE $PORT

# Launch the app
CMD [ "sh", "-c", "mvn -Dserver.port=${PORT} spring-boot:run" ]
```

As you can see on the pipeline page, a lot of pipelines already ran before Pipelines page

If you open one of them, you will see all the jobs launched automatically for you by AutoDevOps. Example of pipeline

This is nice! But in our case, not really helpful as we want to customize the way the application is deployed to

our cluster.

Customizing AutoDevOps

In order to be able to change the behavior, you will have to create a `.gitlab-ci.yml` file to override the default configuration.

As we still want to use the default behavior of AutoDevOps, we will need to load the AutoDevOps configuration at the beginning of our `.gitlab-ci.yml` file.

```
# Include AutoDevops
include:
  - template: Auto-DevOps.gitlab-ci.yml
  - template: Jobs/Deploy.latest.gitlab-ci.yml #
```

After that you can change any job/stage of the AutoDevOps that are set in Auto-DevOps.gitlab-ci.yml

As we want to customize the way the application is deployed to our cluster, we are going to override some of the jobs set in this file Jobs/Deploy.latest.gitlab-ci.yml

Especially the `review` to be able to customize the URL where we are going to deploy our app.