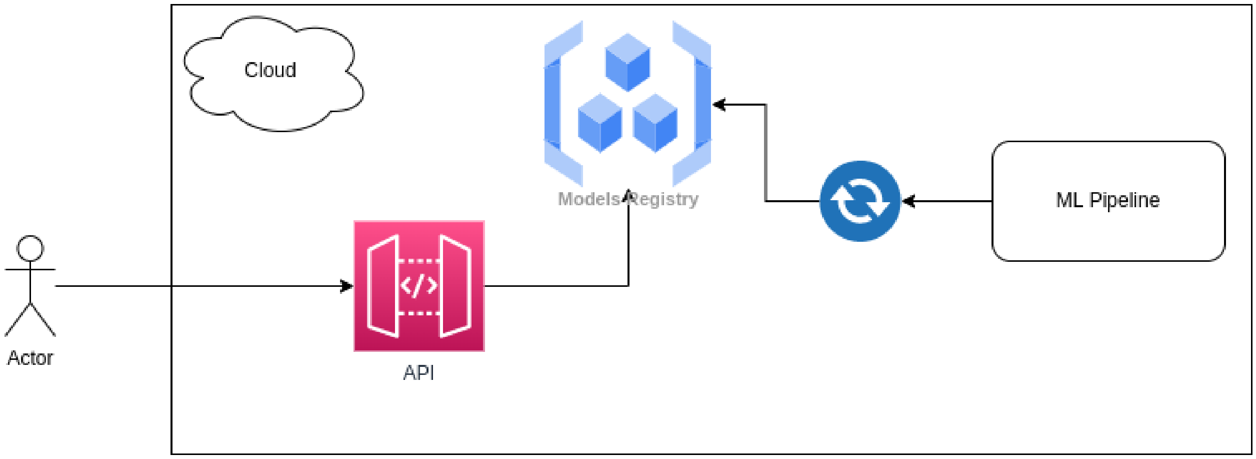
Task 2

How would you design a devops pipeline using e.g. Github Actions for a python package? Which functionalities would you include to ensure code quality and Consistency?

In order to ensure code quality and consistency CI/CD pipelines should be used to ensure the following:

- Unit tests
- Code style checks (pep 8)
- Test coverage
- License check
- Dockerizing the code
- Pushing docker images to the docker registry
- For a feature complete
    - Automatic deployment on development environments
    - Integration tests pipelines should also run
- For releases
    - Integration tests pipelines should also run
    - Automatic deployment on staging environments

Assuming the pipeline you implemented will be deployed as a product. Now the customer also wants to enable real time classification and consume an API that returns the classification results. How would you fit that into the existing architecture?



A simple solution is to include the model with the api. However, such architecture would not scale, or can be maintained really well.

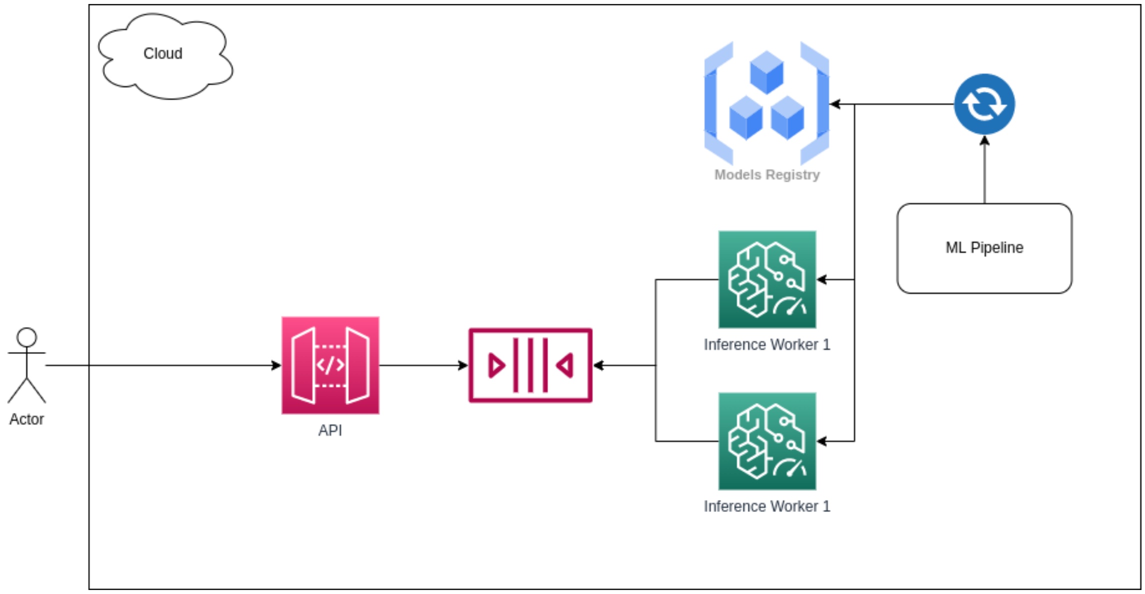
The better solution is to serve the machine learning models in a models registry e.g ML Flow and through that registry models can be used by different services. In this case the API. The benefit of this is we decouple the API from the Models for development and deployment.

The whole system has been a huge success and also other customers want to use it. How would you adapt everything to be able to serve multiple customers with this product? Especially keep in mind scalability and data privacy.

In this scenario, I would adopt a worker/producer architecture. Where the API will serve the requests asynchronously and the inference will be carried out by inference workers that consume requests from the requests queue. These inference workers can scale horizontally and automatically. This will enable the system to scale under load but also save resources when the system is not in use.

Assuming that we are using the same model for all customers, to ensure a higher data privacy regarding requests. We can have a dedicated queue for requests and responses per customer. That will ensure that no customer can access data other than his.

In case, we want to fine tune our models per customer. We can have multiple inference workers that each listens to a specific message queue and can respond only to a specific queue as well. As for training, we will have an automated pipeline that trains these customer specific models



What would you recommend to automatically transfer machine learning models to production by running microservices for inferencing?

By following a microservice architecture and using continuous deployment we can ensure automatically deploying models to the models registry which in turns make the models available to all other services such as the inference workers. These services can in turn use the latest models once they are available.