Alvenir

alvenir.ai

The Journey from ML Model to ML Product

rasmus@alvenir.ai

Agenda

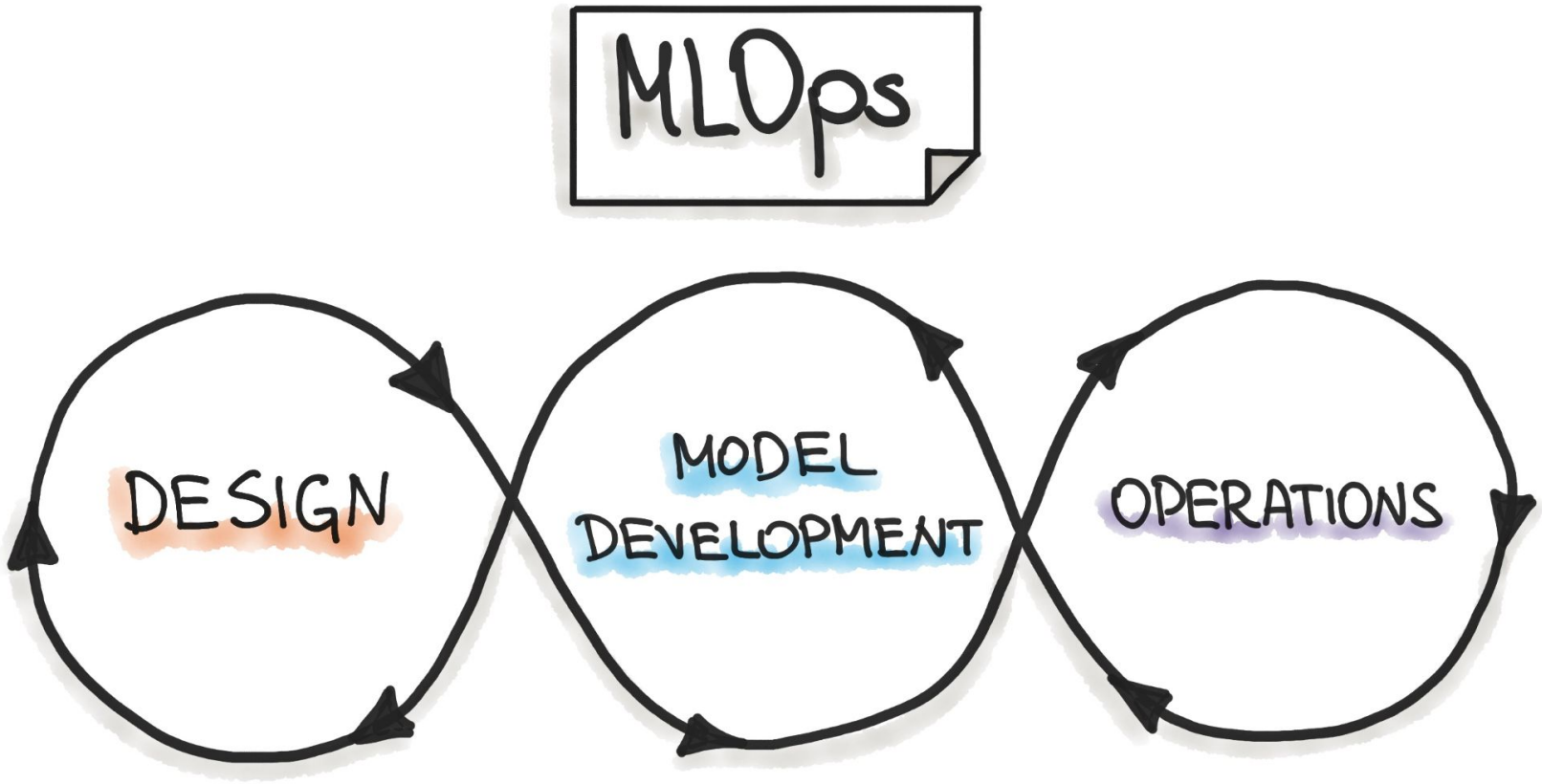
- What is ML Ops?
- Model wrapping
- Containerization
- Break
- Cloud providers
- Cloud Deployment
- Scaling choices
- Navigating ML Ops in a startup environment
- Case discussions

rasmus@alvenir.ai

What is ML Ops?

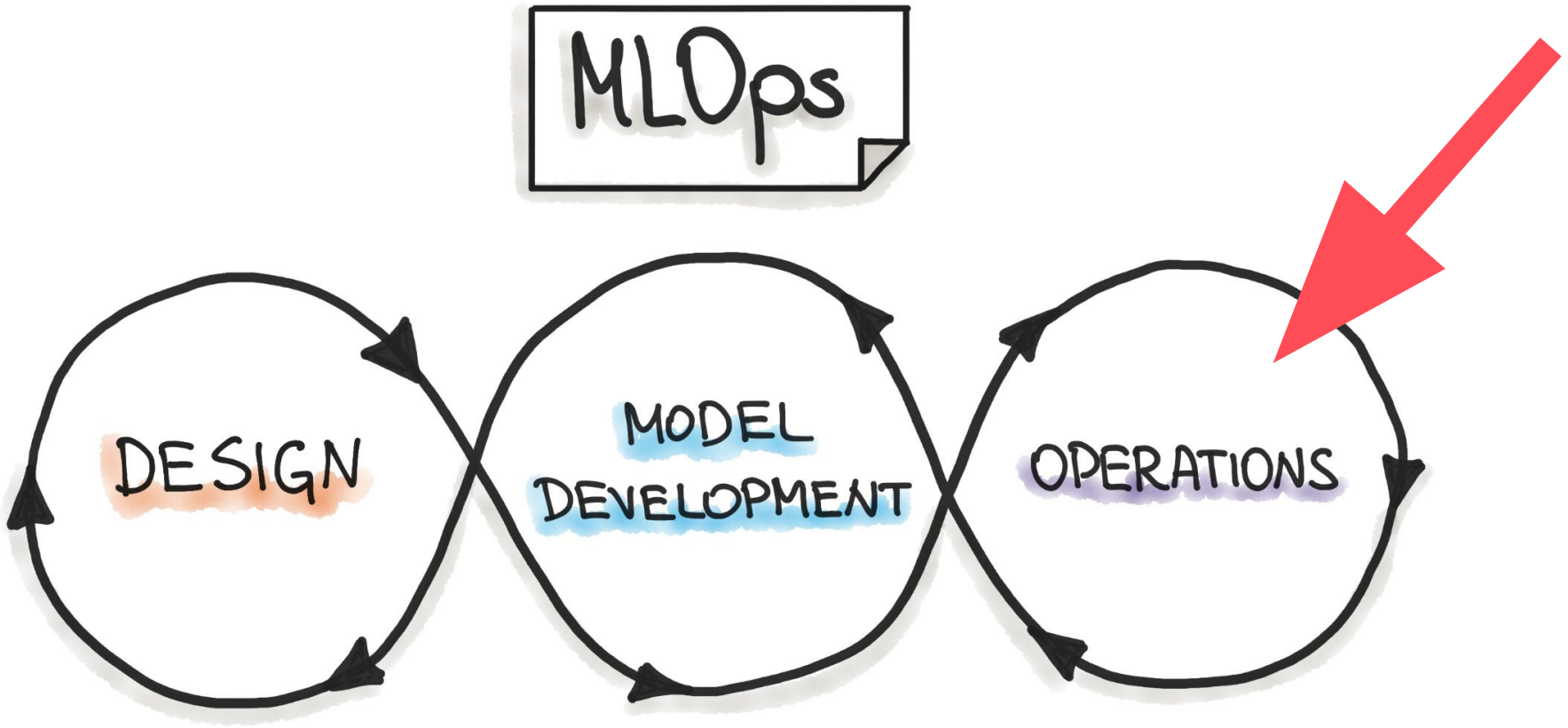
- Machine Learning + Operations = ML Ops
- The management of the production ML lifecycle

rasmus@alvenir.ai



Source: https://ml-ops.org/content/mlops-principles

rasmus@alvenir.ai



Source: https://ml-ops.org/content/mlops-principles

MLOps at a high level

1. Optimizing workflows
   - Getting organized cost time initially but will save you time down the line
2. Versioning
   - Keep track of code changes, trained models etc. so everything can be backtracked
3. Automatization and Continuous X
   - Make sure that new changes automatically gets tested, deployed etc.
4. Reusability
   - Why rewrite the same code for a new project if you can reuse
5. Reproducibility
   - Make sure that your results can be redon by others

Source: https://github.com/SkafteNicki/dtu_mlops

MLOps at a high level

1. Optimizing workflows
   - Getting organized cost time initially but will save you time down the line
2. Versioning
   - Keep track of code changes, trained models etc. so everything can be backtracked
3. Automatization and Continuous X
   - Make sure that new changes automatically gets tested, deployed etc.
4. Reusability
   - Why rewrite the same code for a new project if you can reuse
5. Reproducibility
   - Make sure that your results can be redon by others

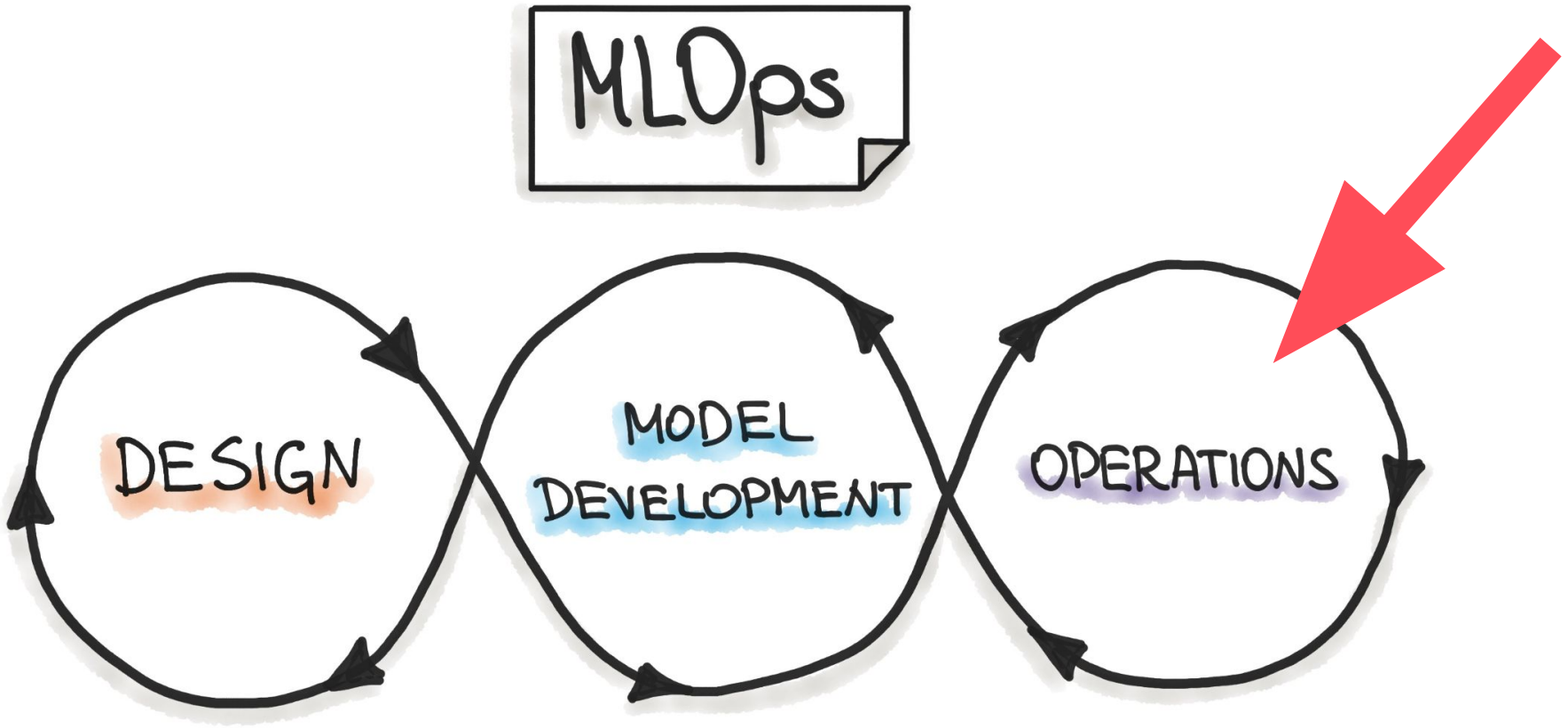
Source: https://github.com/SkafteNicki/dtu_mlops

General important things

- Make sure your code is organized.
  - Very important when multiple people are working on the same project

- Be consistent with your code, and make sure everyone complies to standard.
  - A linter can help with syntax and complying to standard coding practices
    - pylint (https://pylint.pycqa.org/en/latest/)
    - Flake 8 (https://flake8.pycqa.org/en/latest/)
  - Examples include line-length, naming conventions etc.

- Use a version control system such as git(https://git-scm.com/) to track changes
  - Who made what changes?
  - When did the change happen?
  - What was changed?



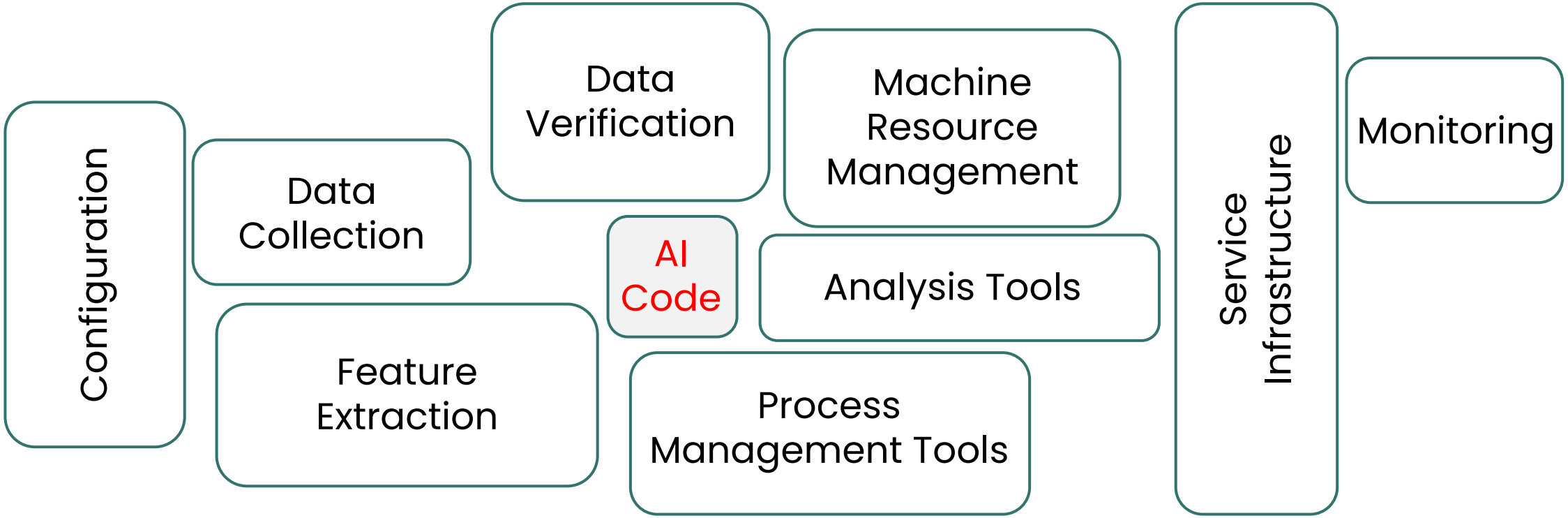
Source: https://ml-ops.org/content/mlops-principles

Why is it important?

AI
Code

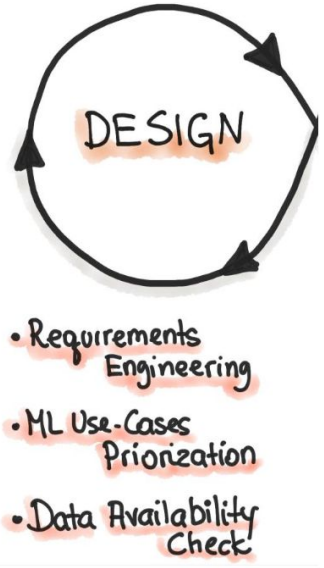
rasmus@alvenir.ai

Why is it important?



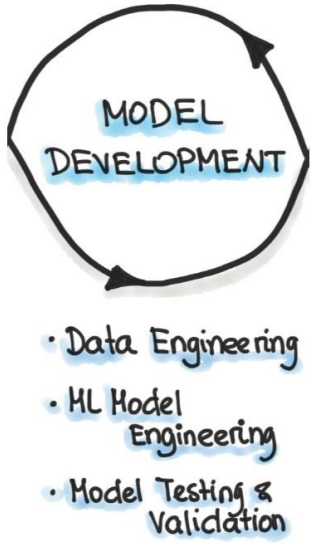
Source: D. Scully et al. “Hidden Technical Debt in Machine Learning Systems

- Analyze the problem
- Look in literature for references
- What data do we have access to in order to investigate / solve the problem?

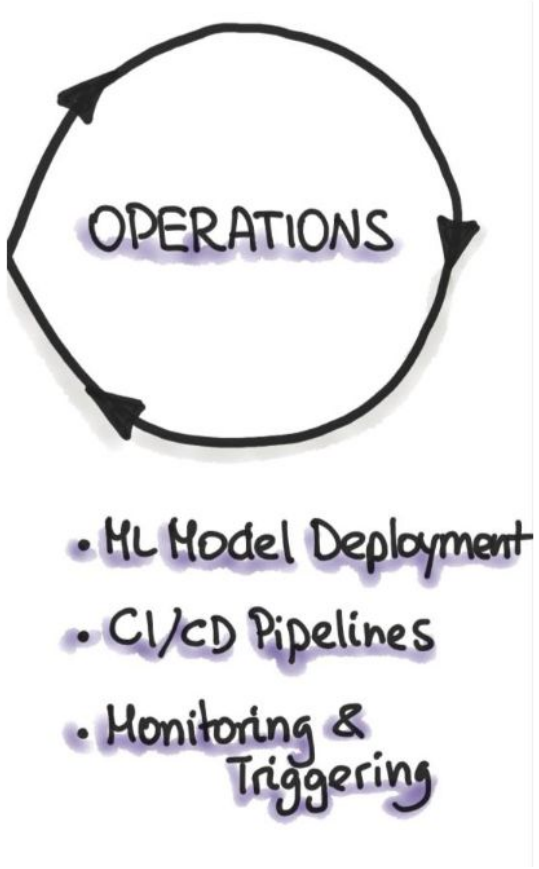


Source: https://ml-ops.org/content/mlops-principles

- Practical machine learning implementation.
- How should data be formatted?
- What are the important features?
- How do we validate and test performance?



Source: https://ml-ops.org/content/mlops-principles



Source: https://ml-ops.org/content/mlops-principles

- **The operations part is running an ML model in production.**
  - **Model deployment**
    - Deployment to a server perhaps in the cloud exposing it to the end-users.
  - **Continuous integration / continuous delivery (CI/CD)**
    - Automatic pipeline so that codebase changes are incorporated directly into the model and the model gets deployed automatically.
    - Requires a lot of automated tests!
      - Unit tests, integration tests, model evaluation tests
  - **Monitoring of the model**
    - How much data is it processing?
    - Is it running as we expect?

rasmus@alvenir.ai

Model deployment

- On demand predictions
- Batch predictions
- On device (i.e. model is running on a mobile phone, embedded)

rasmus@alvenir.ai

Model deployment

- On demand predictions
- Batch predictions
- On device (i.e. model is running on a mobile phone, embedded)

rasmus@alvenir.ai

Model deployment

- On demand predictions
- Batch predictions
- On device (i.e. model is running on a mobile phone, embedded)

rasmus@alvenir.ai

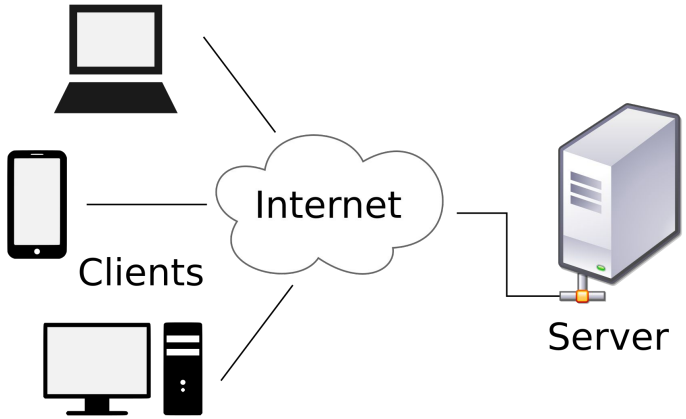
On Demand Predictions

- Spelling correction
- Live Transcription
- Sentiment of what you have written

rasmus@alvenir.ai

On Demand Predictions

- What is a REST API?
  - Client / server model
  - Stateless
  - Uses JSON documents for representation of data
  - Different type of request methods
    - get, put, post, delete, patch
  - post contains a json body of data
    - Let's focus on that



- Other protocols exist, however, REST is very popular especially for machine learning models

rasmus@alvenir.ai

On Demand Predictions

- Wrap the model in a REST API framework to create the server such as
  - Flask (https://flask.palletsprojects.com/en/2.1.x/)
  - Fast API (https://fastapi.tiangolo.com/)

- Let’s go through af quick Fast API example!

rasmus@alvenir.ai

On Demand Predictions

- **A file called api.py**

```python
from fastapi import FastAPI

from api_models import PredictInput
from model import MyModel, convert_api_input_to_model_input

# Instantiating FastAPI
api = FastAPI()

lr_model = MyModel()


# Defining the post prediction endpoint
@api.post('/predict')
async def predict(prediction_input: PredictInput):
    # Convert api input to model input
    model_input = convert_api_input_to_model_input(prediction_input)

    # Get a prediction
    pred = lr_model.predict(model_input)[0]

    return pred
```

rasmus@alvenir.ai

On Demand Predictions

- Now, you can deploy the app locally using the command

```
uvicorn api:api --host 0.0.0.0 --port 8080
```

- Now, the model is running on your machine on 0.0.0.0 using port 8080 and supports the endpoint predict
- url: 0.0.0.0:8080/predict

rasmus@alvenir.ai

On Demand Predictions

- **You can now call the model from other processes / scripts**

```
import json

import requests

url = 'http://0.0.0.0:8080/predict'
data = {'sentence': 'i am very bored and this api is bad}
response = requests.post(url, data=json.dumps(data))
print(response.json())


{'label': 'NEGATIVE', 'score': 0.9998231530189514}
```

Other options for ML

- Uses a model object and then takes care of exposing it with REST
  - torchserve (https://pytorch.org/serve/)
  - tensorflow-serving (https://www.tensorflow.org/tfx/guide/serving)

Batch predictions

- Use batch predictions if:
  - The response is not needed right away
  - The amount of data is big

- Can be a very simple i.e. a script that reads a file, performs predictions and outputs a file.

- Use a framework to schedule tasks that are run based on a trigger event
  - Trigger event might be every night at 12:00
  - Trigger event might be data exceeds x amount of samples

rasmus@alvenir.ai

Batch predictions

- Common tools to schedule batch jobs
  - Airflow (https://airflow.apache.org/)
  - Prefect (https://www.prefect.io/)

rasmus@alvenir.ai

Recap

- Now we know how to wrap machine learning code into a reachable service

- So how do we go from running the service on our own machine to running it somewhere where we can expose it to everyone?

What is important?

- Reproducibility
- Work seamlessly on different machines
- Easy to deploy anywhere

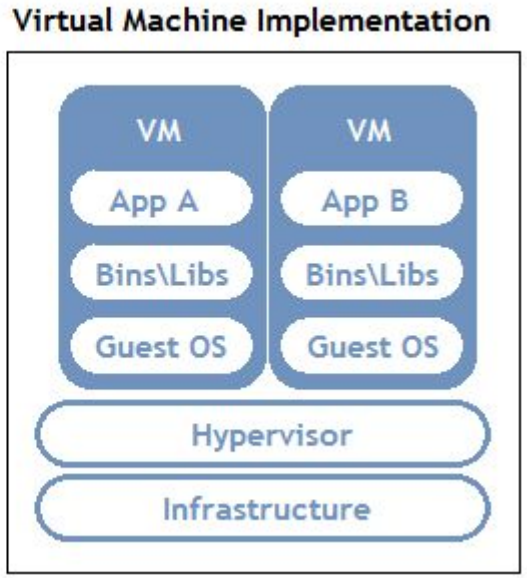
rasmus@alvenir.ai

Solution = Docker

rasmus@alvenir.ai

Virtual Machines

- **What is a virtual machine?**
  - **Copy of the machine from operating system and up!**
  - **Each virtual machine runs its own operating system kernel and functions separately from the other VMs, even when they are all running on the same host i.e. same physical machine.**



Source: https://www.aquasec.com/cloud-native-academy/docker-container/

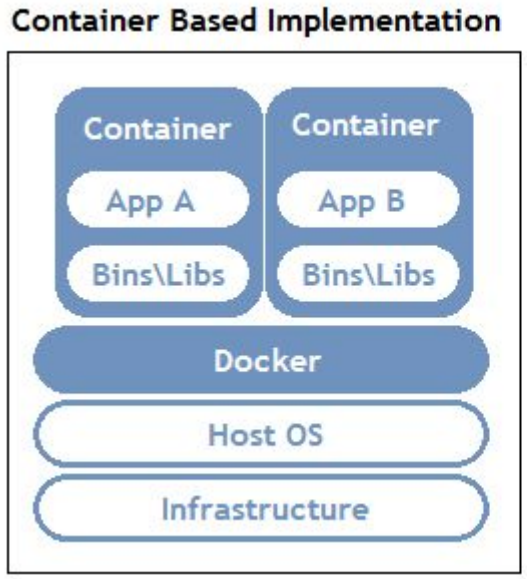
- **If all applications run on the same operating system, do we then really need a host operating system for each machine?**

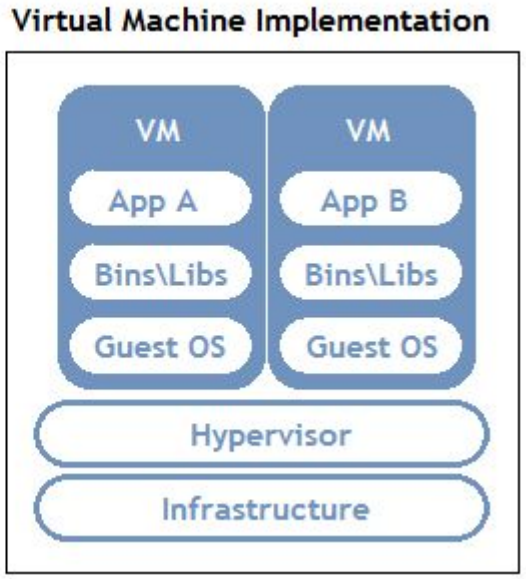
Source: https://www.aquasec.com/cloud-native-academy/docker-container/

rasmus@alvenir.ai

Containerization

- **Enter Docker (https://www.docker.com/)**
  - **Software framework for building, running, and managing containers on servers and the cloud.**

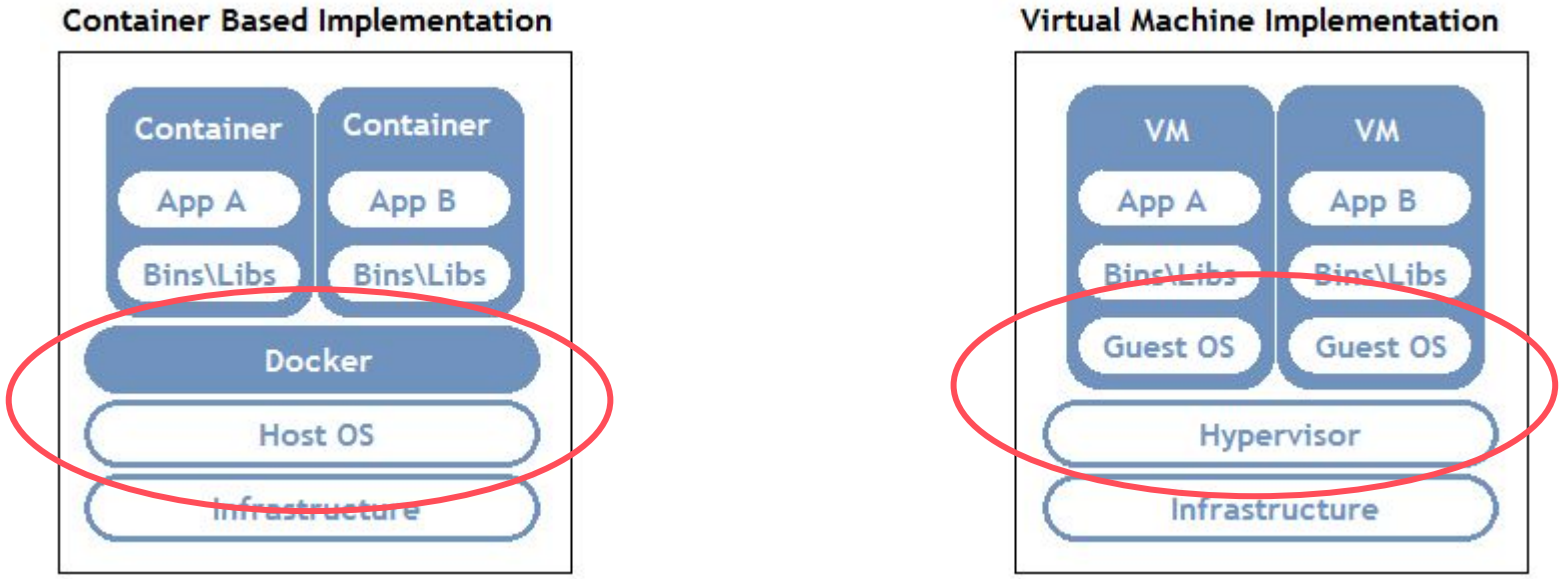




Source: https://www.aquasec.com/cloud-native-academy/docker-container/

Containerization

- **Enter Docker (https://www.docker.com/)**
  - **Software framework for building, running, and managing containers on servers and the cloud.**



Source: https://www.aquasec.com/cloud-native-academy/docker-container/

Containerization

- Using containers is more lightweight than pure virtual machines
  - Since containers share the same operating system kernel
  - Containers are smaller than virtual machines
  - Faster boot time
  - Can run more applications on the same server using containers i.e. maximizing utilization of resources.
- Containers have become the standard way to package and deploy code and its dependencies.

rasmus@alvenir.ai

Containerization

“An analogy we can use here is that of homes and apartments. Virtual Machines are like homes: stand-alone buildings with their own infrastructure including plumbing and heating, as well as a kitchen, bathrooms, bedrooms, and so on. Docker containers are like apartments: they share common infrastructure like plumbing and heating, but come in various sizes that match the exact needs of an owner.”

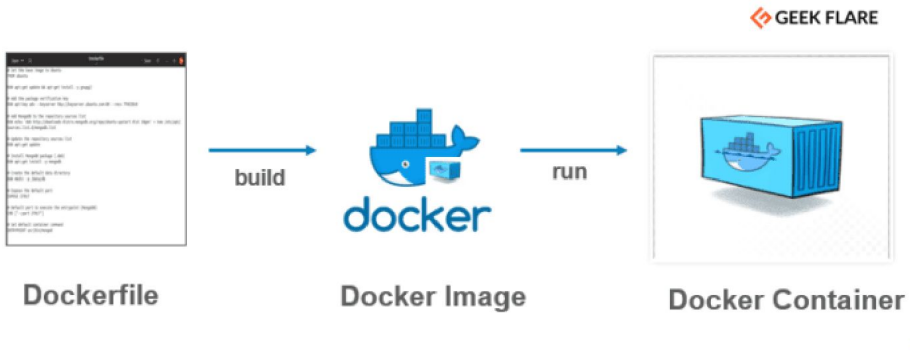
Quote taken from: https://wsvincent.com/beginners-guide-to-docker/

Containerization

Docker

MLOps

- A way to create containerize applications = specialized VMs



Source:
https://github.com/SkafteNicki/dtu_mlops/blob/main/s3_reproduceability/lecture/Reproducibility.pdf

Dockerfile

- A file that contains the relevant commands a user could call on the command line to assemble an image

rasmus@alvenir.ai

Dockerfile

```
# Start from the official Python base image.
FROM python:3.8

# Set the current working directory to /code.
# This is where we'll put the requirements.txt file and the app directory.
WORKDIR /code

# Copy the file with the requirements to the /code directory.
COPY ./requirements.txt /code/requirements.txt

# Install dependencies through the requirements.txt file
RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt

# Copy the relevant files inside the /code directory.
COPY ./api.py /code/api.py
COPY ./model.py /code/model.py
COPY ./api_models.py /code/api_models.py

# Make sure that port 8080 is exposed
EXPOSE 8080

# CMD takes a list of strings, each of these strings is what you would type in the command line separated by spaces.
CMD ["uvicorn", "api:api", "--host", "0.0.0.0", "--port", "8080"]
```

rasmus@alvenir.ai

Building image

- **Stand in the same directory as where your Dockerfile is.**
- **Build**

```
docker build . -t alvenir/my_api:0.0.1
```

Running container

- **Run**

```
docker run -p 8080:8080 --name "my_machine_learning_api" --rm alvenir/my_api:0.0.1
```

rasmus@alvenir.ai

Break

rasmus@alvenir.ai

Cloud solutions

- **Now you have your machine learning model as a docker image than can be run as a docker container.**

- **Where should we run the service?**

Source: https://cloudcomputinggate.com/cloud-provider-definition/

Cloud solutions

- **How to choose cloud provider?**
  - **Most of the bigger cloud solutions support the core required features, they just name them slightly differently.**
  - **Do you have any experience with any cloud providers? Or friends / colleagues who might be able to help you?**
    - **Then use the one you/your friends have experience with.**
  - **Where can you get the most free credits for experimentation?**
    - **It is usually possible to get some free cloud credits to start experimenting with their platform.**

rasmus@alvenir.ai

We picked google cloud

- Different deployment options.
  - Google Functions
  - Google Run
  - Google Kubernetes Engine
  - And many more options

rasmus@alvenir.ai

Google cloud

- Google cloud generally needs to fetch the docker image from somewhere
  - Push the docker image to a container registry
    - You can use google clouds own Google Container Registry (GCR) or Docker registry (https://docs.docker.com/registry/)
  - If you use GCR, then google can automatically fetch the image without additional authentication (assuming you're using the same project for gcr and deploy).

```
docker tag alvenir/my_api:0.0.1 eu.gcr.io/mlopspres/alvenir/my_api:0.0.1 && docker push eu.gcr.io/mlopspres/alvenir/my_api:0.0.1
```

Google Run

- see more at https://cloud.google.com/run/docs

```
gcloud run deploy my-api-model-service --memory 2G --region europe-west4 --allow-unauthenticated \
--image eu.gcr.io/mlopspres/alvenir/my_api:0.0.1
```

Google Functions

- **see more at https://cloud.google.com/functions/docs/tutorials**

```python
import functions_framework

@functions_framework.http
def hello_get(request):
    """HTTP Cloud Function.
    Args:
        request (flask.Request): The request object.
        <https://flask.palletsprojects.com/en/1.1.x/api/#incoming-request-data>
    Returns:
        The response text, or any set of values that can be turned into a
        Response object using `make_response`
        <https://flask.palletsprojects.com/en/1.1.x/api/#flask.make_response>.
    Note:
        For more information on how Flask integrates with Cloud
        Functions, see the `Writing HTTP functions` page.
        <https://cloud.google.com/functions/docs/writing/http#http_frameworks>
    """
    return 'Hello World!'
```

```
gcloud functions deploy hello_get \
    --runtime python39 --trigger-http --allow-unauthenticated
```

rasmus@alvenir.ai

Monitoring tools

- Cloud providers provide you with a few monitoring tools that allows you to see the number of resources being used, latency etc. and set alerts based on the outputted logs and other metrics.

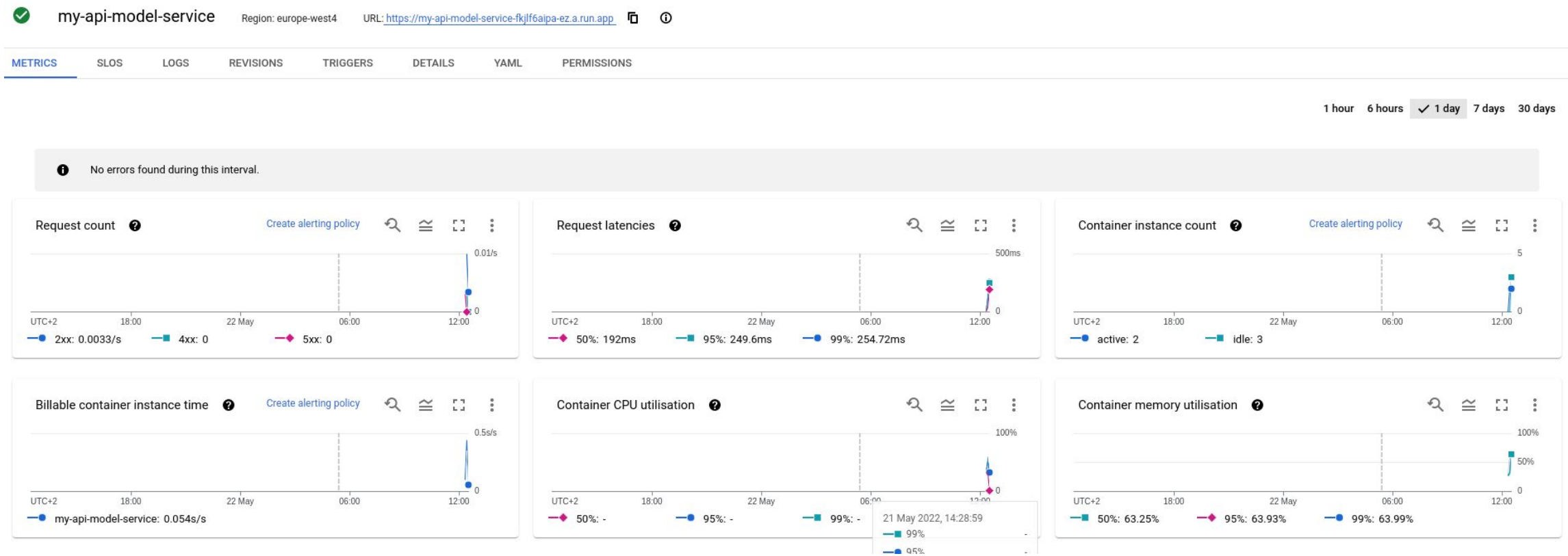
- If you need more advanced monitoring tools i.e. how many requests is this specific customer doing, what are the models predicting right now, then you need to implement it yourself.
  - It could be as a dashboard e.g. Grafana (https://grafana.com/)

rasmus@alvenir.ai

Alvenir

alvenir.ai

Monitoring tools



rasmus@alvenir.ai

Need to be able run your solution anywhere?

- Kubernetes (https://kubernetes.io/) is a container orchestration tool
- Most cloud providers support kubernetes clusters.
- A kubernetes cluster allows you have compute nodes where you can deploy your containers.
- Each node has a certain amount of resources

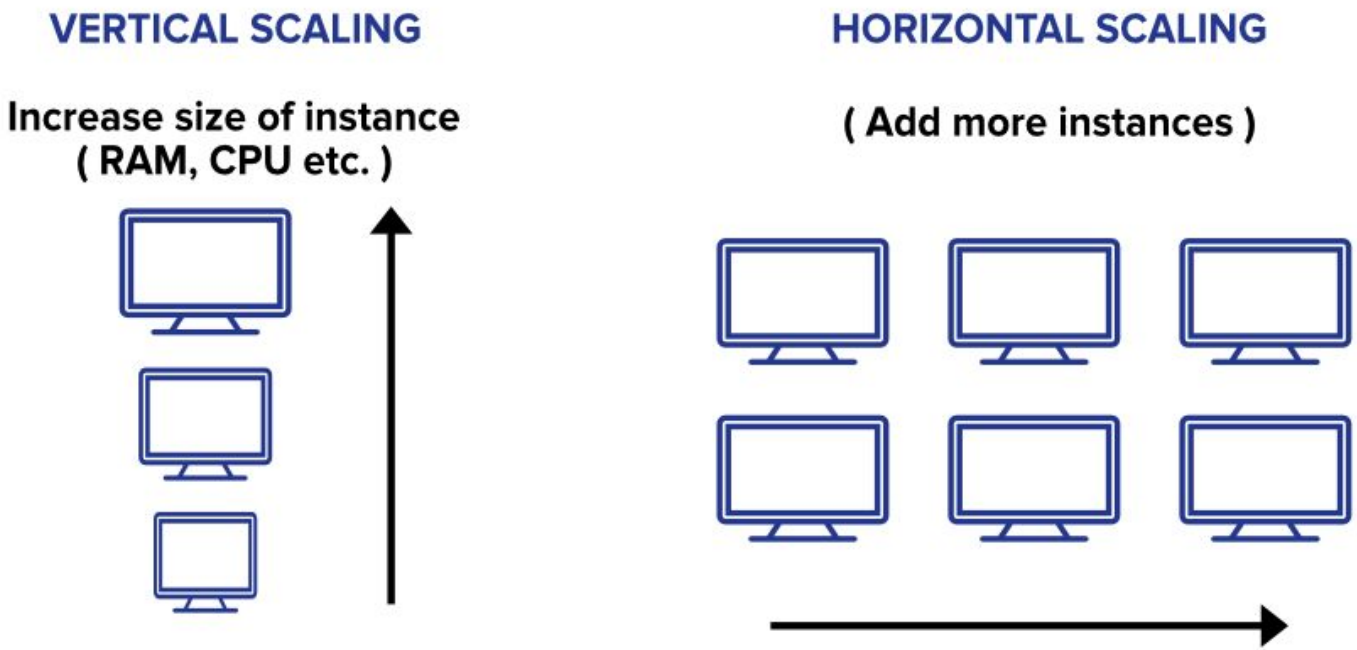
rasmus@alvenir.ai

Need to be able run your solution anywhere?

- Many companies who manage their own servers are also able to run a kubernetes cluster.
- It works quite well with scaling your resources up and down.
- Drawbacks:
  - It requires more work to setup and run a kubernetes cluster
  - If not careful, then you might get entangled up with cloud specific options and deploying your model / system elsewhere is then not as easy as it should be

rasmus@alvenir.ai

Scaling



Source: https://www.geeksforgeeks.org/system-design-horizontal-and-vertical-scaling/

Horizontal Scaling

- How do you scale if you need to handle for example 10000 concurrent requests? One GPU instance is not enough.
- Horizontal scaling refers to adding more containers running your service to meet the demands.
- A load balancer is typically then put in front of your service which distributes requests across all of your running nodes.

rasmus@alvenir.ai

Autoscaling

- You can even autoscale based on number on specific events.
  - Number of requests
  - Cpu usage
  - Basically, any event

- When a given event is reached, automatically spin up one more instance of your service / machine learning model.

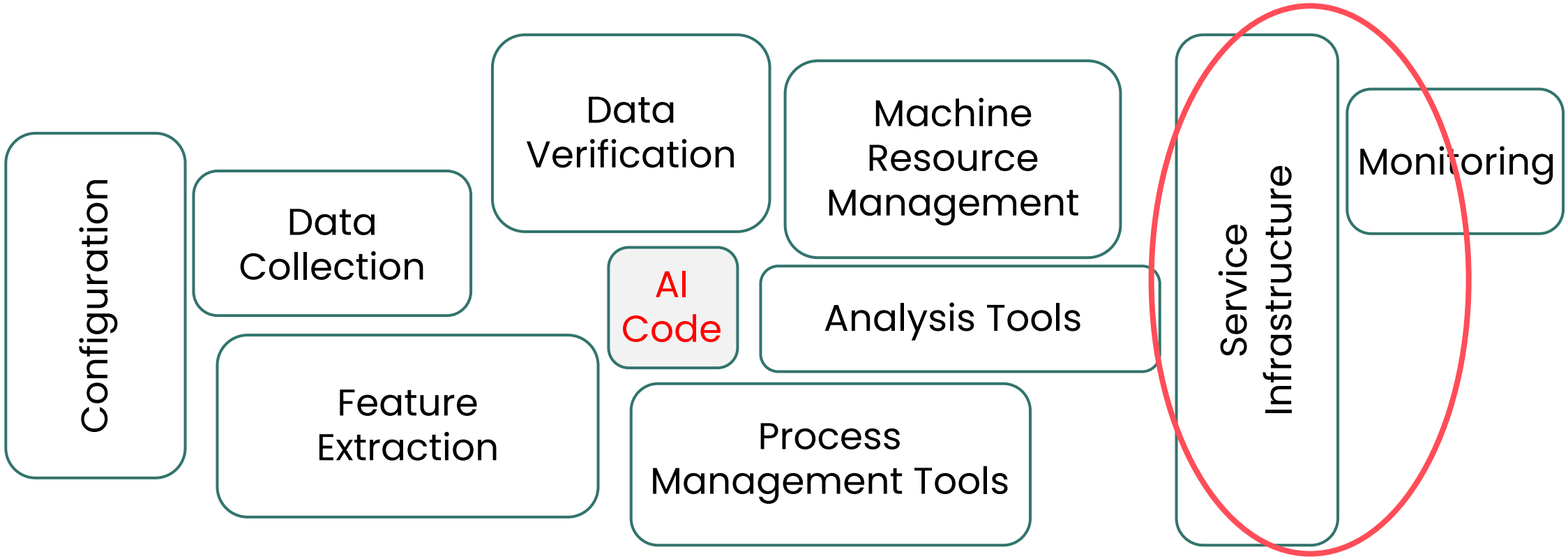
rasmus@alvenir.ai

GPU vs CPU

- It all comes down to the specific / expected load
- Generally, GPUs are more expensive than CPUs
- It is a tradeoff between inference time vs costs
- If you have 10 requests an hour for your machine learning model, and you need to return an instant response, then consider using a CPU since it is the cheaper option.
    - Consider horizontal scaling to handle peaks.

rasmus@alvenir.ai

Why is it important?



rasmus@alvenir.ai

Alvenir

alvenir.ai

Navigating ML Ops in a Startup company

rasmus@alvenir.ai

ML Ops in a startup

- Ml Ops is very nice! But configuring / implementing a full ML Ops cycle is very time consuming.

- We usually want to get things into production fast
  - Testing with users, getting feedback, fail fast and iterate etc.

- As the company grows, the more important it becomes to utilize ML Ops correctly!

rasmus@alvenir.ai

So what can you do?

- Cloud providers have an easy to use user interface for deploying / managing your services.
  - Be careful with these features
- Make sure that everything you do can be performed with scripts and configuration files!
  - Then eventually these scripts can be used in automatic pipelines.

rasmus@alvenir.ai

So what can you do?

- Aim for storing relevant data in the cloud
    - Eventually you can setup training / data validation / model experimentation in the cloud.

- Make sure you have consistent, shareable and easy maintainable code
    - Use a linter!
    - Use git!

rasmus@alvenir.ai

So what can you do?

- Everyone wants to be cloud independent
  - Why?
  - Really consider whether your business case really requires you to be cloud independent.

rasmus@alvenir.ai

So what can you do?

- Essentially, don’t take too many shortcuts
  - Technical debt will pile up!
- Try to incorporate more and more parts of the ML Ops cycle while you move forward.

rasmus@alvenir.ai

Alvenir setup

- Docker
- Kubernetes in google cloud to run everything
  - We are actually migrating to AWS, so glad we chose the kubernetes option.
- Local GPU server for experiments due to the computational requirements (hint very expensive) for training speech recognition systems.
- Batch jobs
- Horizontal scaling
- Scale-to-zero
- Autoscaling
- CPU and GPU instances (depending on workload)
- Kafka
- Custom monitoring tools
- REST API and gRPC for different kinds of integrations

rasmus@alvenir.ai

Questions ?

rasmus@alvenir.ai

Discussion session

- 3 cases
  - Each case has a machine learning model.
  - Go through all cases.
  - Discuss how you would take the machine learning model to production. What tools would you use?
  - Then a member from each group explains the decisions / thought process for a single case.

- Think about;
  - How to perform the predictions?
  - How to scale
  - What cloud provider?
  - CPU / GPU?
- You can make your own assumptions about the potential customers / use case if needed to pick the right tools.

rasmus@alvenir.ai

Case 1

- You have designed and implemented a model that is able to predict whether a patient is likely going to get diabetes based on very private data.
- Multiple companies want to buy the service and apply the model on their own data. However, the customers needs to deploy the model on their own infrastructure as they cannot transmit data directly to you.
- Your job is to take the machine learning model to production! Which tools are you going to use?

- Think about;
  - How to perform the predictions?
  - How to scale?
  - What cloud provider?
  - CPU / GPU?

rasmus@alvenir.ai

Case 2

- You have implemented a machine learning model that is able to predict how many listens a given podcast will have based only on the audio! The accuracy is surprisingly very high.
- Multiple companies are interested in buying the service. They want to send a lot of audio files at once a few days before releasing the podcasts, so that they can verify how well their podcasts will perform before exposing them to the public, allowing them to make last minute changes.
- Your job is to take the machine learning model to production! Which tools are you going to use?

- Think about;
  - How to perform the predictions?
  - How to scale?
  - What cloud provider?
  - CPU / GPU?

rasmus@alvenir.ai

Case 3

- You have implemented a machine learning model that is able to predict how the weather is going to be for the next 10 minutes just by knowing your location, however, it requires quite a lot of computing resources.
- A lot of companies want to integrate this functionality directly into their apps.
- Your job is to take the machine learning model to production! Which tools are you going to use?

- Think about;
    - How to perform the predictions?
    - How to scale?
    - What cloud provider?
    - CPU / GPU?

rasmus@alvenir.ai

How to continue with MLOps?

- Check out the amazing DTU MLOps course by Nicki Skafte Detlefsen
  - https://github.com/SkafteNicki/dtu_mlops

```
@misc{skafte_mlops,
    author       = {Nicki Skafte Detlefsen},
    title        = {Machine Learning Operations},
    howpublished = {\url{https://github.com/SkafteNicki/dtu_mlops}},
    year         = {2021}
  }
```

- Get practical experience by deploying your own models in the cloud using the tools / learnings explained in this presentation.

rasmus@alvenir.ai