AWS User Group **Venezia**

Giovedì 19 Settembre

in presenza e in streaming

alle ore 19:00

Graphs are everywhere…
even on AWS !

Marco Falcier
Software Engineer
@ Generali Italia

Data scientist vs Cloud engineer:
who wins ?

Alessandra Bilardi
Data & Automation Specialist
@ Corley Cloud

Promotori di AWS User Group Venezia #4

AWS
User Groups

Venezia

Agenda

Speech
Iniziative
Prossimo incontro
Spritz


Speech

Graphs are everywhere… even on AWS !

Marco Falcier - Software Engineer @ Generali Italia

@AWSUserGroupVenezia #4 #Meetup #AWS

Who am I?

- Software Engineer @ Generali
- Serverless Enthusiast
- Kotlin Lover
- Neo4j Certified Professional

- Twitter: mfalcier
- GitHub: mfalcier
- LinkedIn: marcofalcier

AWS
User Groups

Venezia

Agenda

What is Neptune?
How does Neptune works?
openCypher: (graphs)-[:ARE]->(everywhere)
Demo
Use cases
Some best practices with Neptune and Graph Databases
Questions

What is Neptune?

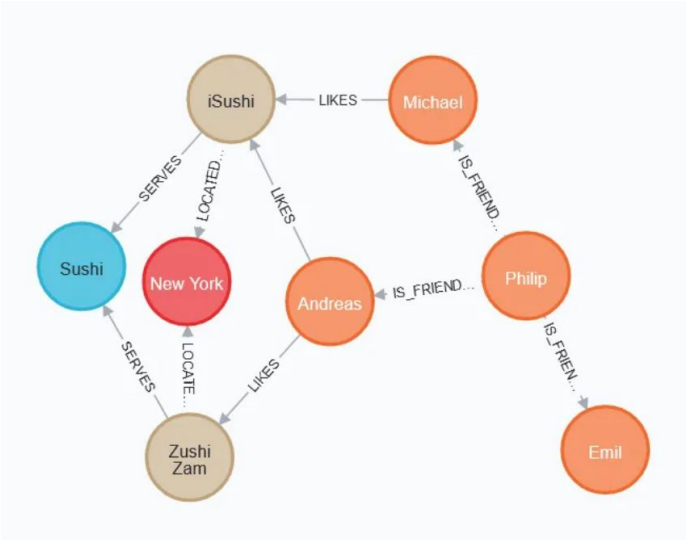
What is Neptune?

- Fully managed AWS' **Graph Database**
- Supports both **RDF** (Resource Description Framework) and **Property Graph**
- Query Languages: Apache TinkerPop **Gremlin**, W3C's **RDS/SPARQL** and **openCypher**
- Provides high availability (3 availability zones per region)
- Provides scalability (up to 128 TiB per cluster and read scales up to 15 replicas per cluster)
- Many integrations with other AWS Services (S3, Lambda, Glue, Athena, DMS, etc.)
- Fully ACID compliant, backups, Snapshots and more

How does Neptune works?

How does Neptune works?

- Property Graph Model:
  - **Nodes** and **relationships** are stored
  - Each node can have **primitive properties**
  - Each relationship can have p**rimitive properties**
- Compared to a Relational database:
  - **Tables** are **Labels**
  - **Rows** are **Nodes**
  - **Joins** are **Relationships**
  - **Many-to-many tables** are **Relationship's properties**
- Graph Database is the **true** and **native** "relational database"
- It works **amazingly** with **highly connected** data!



openCypher: (graphs)-[:ARE]->(everywhere)

openCypher: (graphs)-[:ARE]->(everywhere)

- Intuitive syntax: easy to learn and use, similar to SQL
- Pattern matching: powerful tool for querying complex graph data
- Declarative language: focuses on “what” to query, not “how” to do it
- Broad support: used by multiple platform (Neo4j, AWS Neptune, Tigergraph, Memgraph, etc.)
- Scalability: optimized for large datasets with complex relationships
- Open source: widely supported by the community

DEMO!!1!11!!ELEVEN!

Use cases

Use cases

- **Social Networks**: model relationships and interactions between users
- **Recommendations Engines**: suggest products based on user connections and preferences
- **Fraud Detection**: identify suspicious patterns in transaction networks
- **Knowledge Graphs**: organize and query complex data relationships
- **Supply Chain Management**: track and optimize product flows across networks
- **Sales Networks**: model relationships between customers, sales reps, and distributors to optimize sales strategies

Some best practices with Neptune and Graph Databases

Some best practices

- Avoid using “small” instances if you want to use Neptune; start from R6 instances
- Graph Databases are not cheap: identify a use case that is worth enough
- Don’t max out or minimize your model: design it from what you’re going to ask to your db
- Change your point of view: graphs are very much different from other db (also RDBMS)
- Start from a napkin: your model can be easily designed without complex tools
- Don’t be afraid of future refactoring, it’s pretty much normal and you can do them


Questions ?
@AWSUserGroupVenezia #4 #Meetup #AWS

Data scientist vs Cloud engineer: who wins ?

Alessandra Bilardi - Data & Automation Specialist @ Corley Cloud

@AWSUserGroupVenezia #4 #Meetup #AWS

Alessandra Bilardi

Data & Automation Specialist @ Corley Cloud

- Coderdojo Mentor
- PyData Venice Evangelist
- AWS User Group Venezia Evangelist

alessandra.bilardi@corley.it
@abilardi
bilardi

corley

aws
PARTNER
Advanced Tier
Services

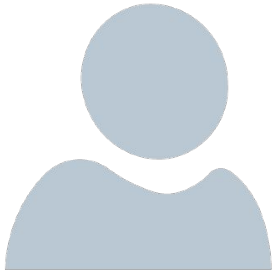
AWS
User Groups

Venezia

Agenda

Use Cases
Methods & actors
Solution
Take Away

Use Case - Start up

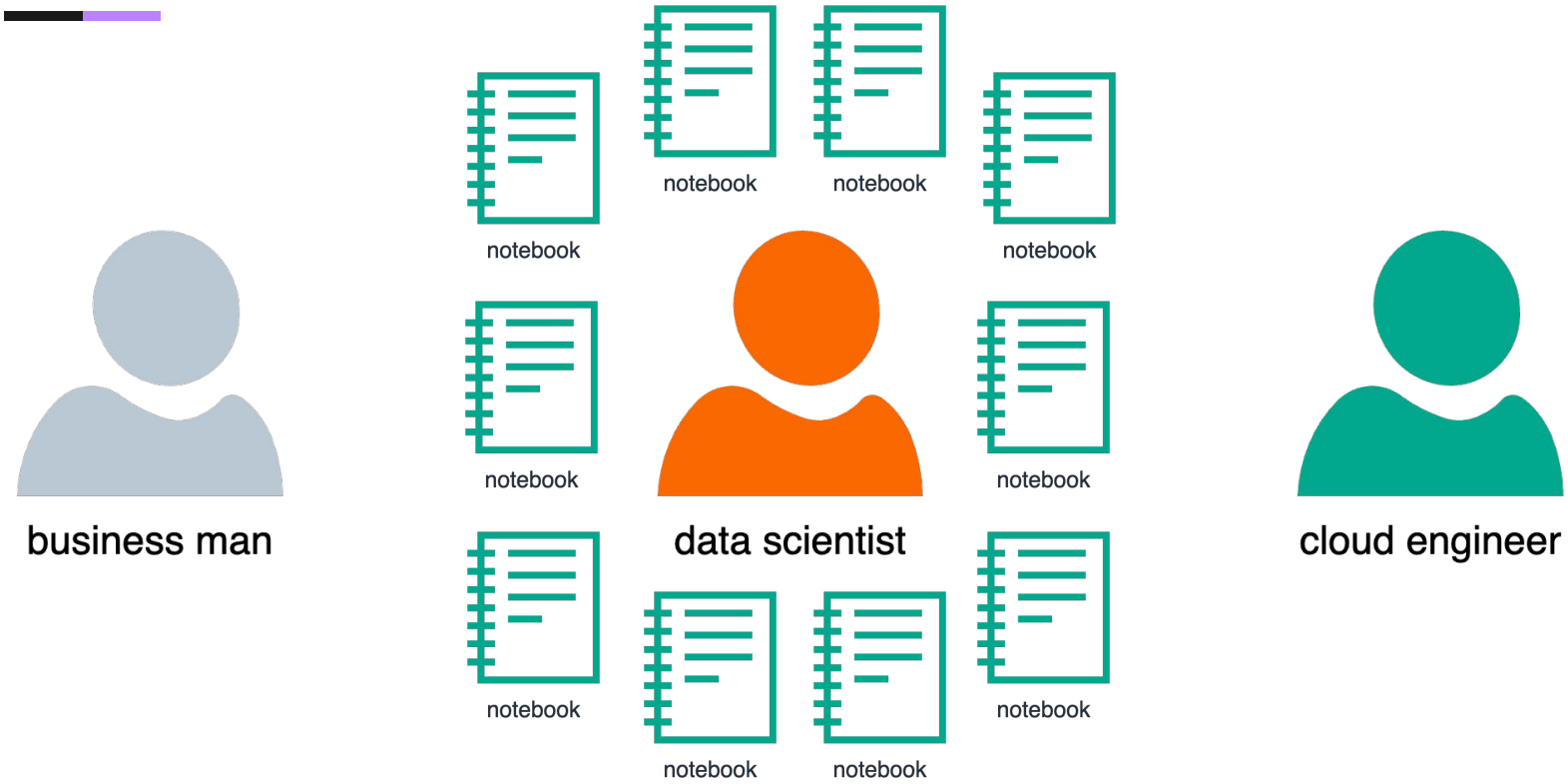


business man

data scientist

cloud engineer

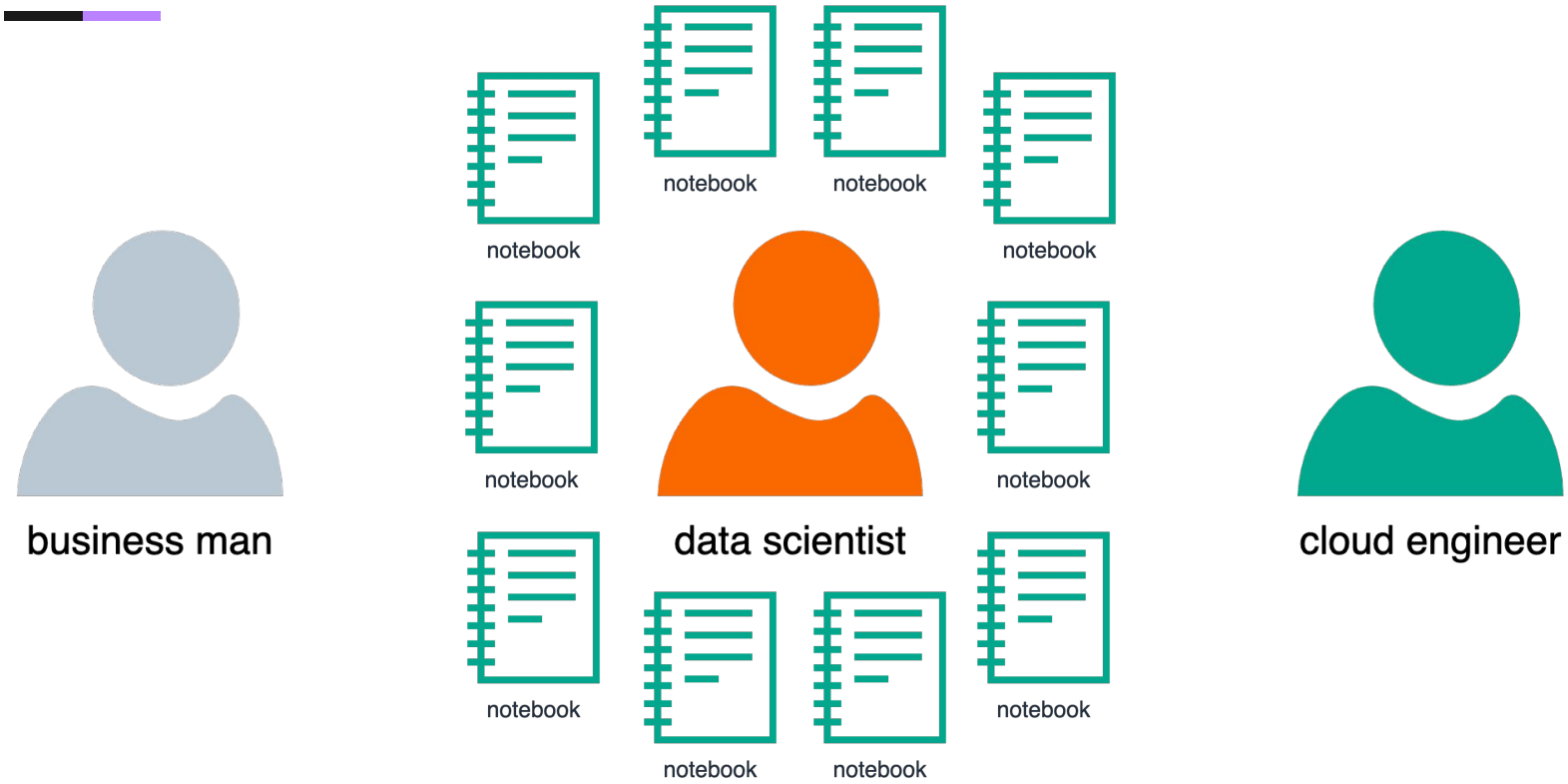
Use Case - Start up



reference

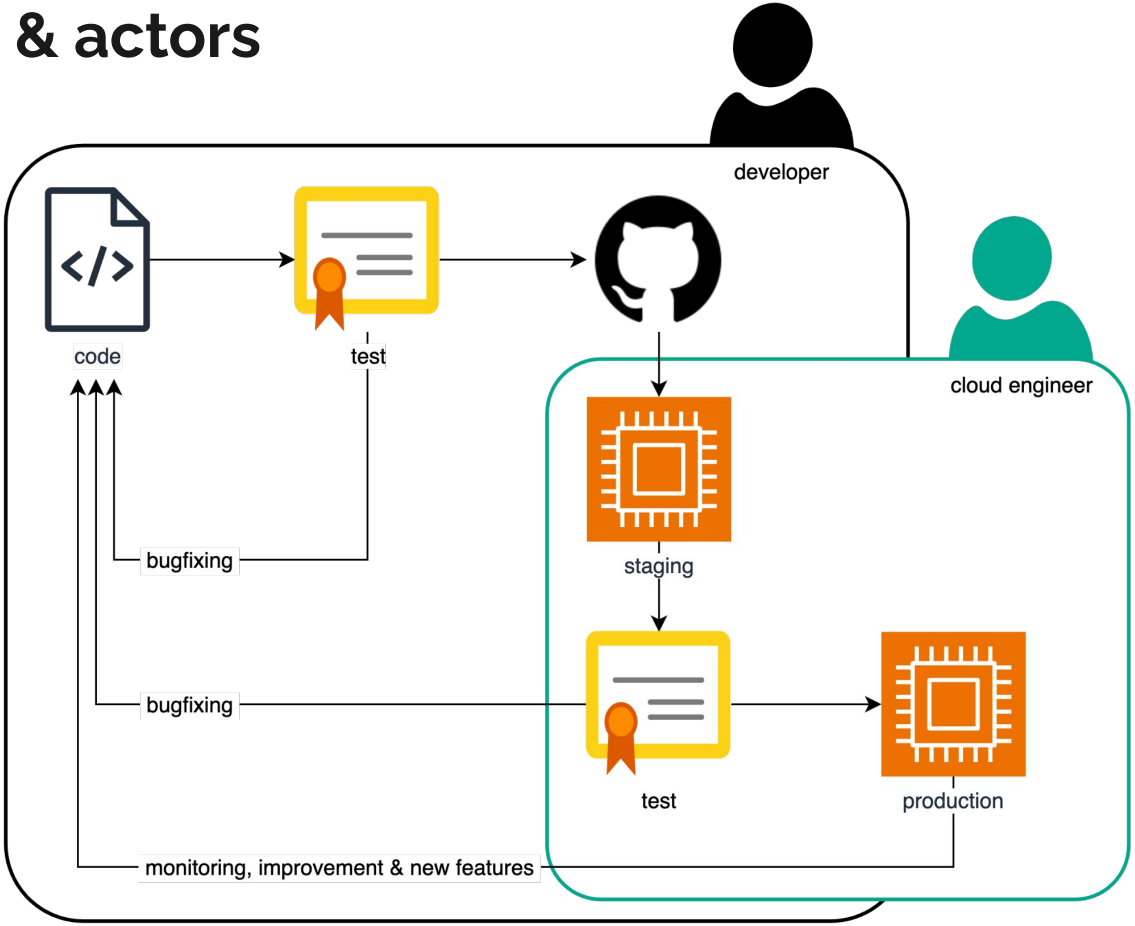
reference

Use Case - Enterprise

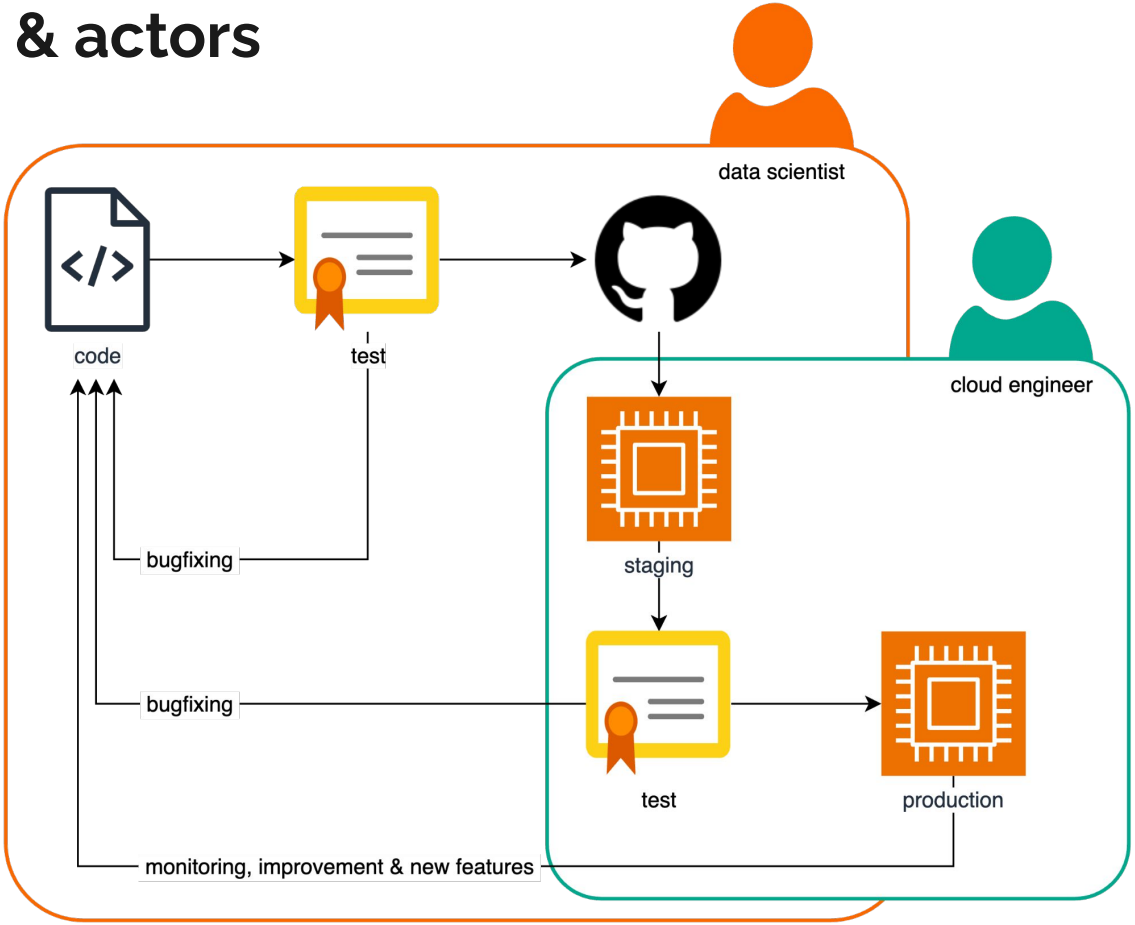


Methods & actors

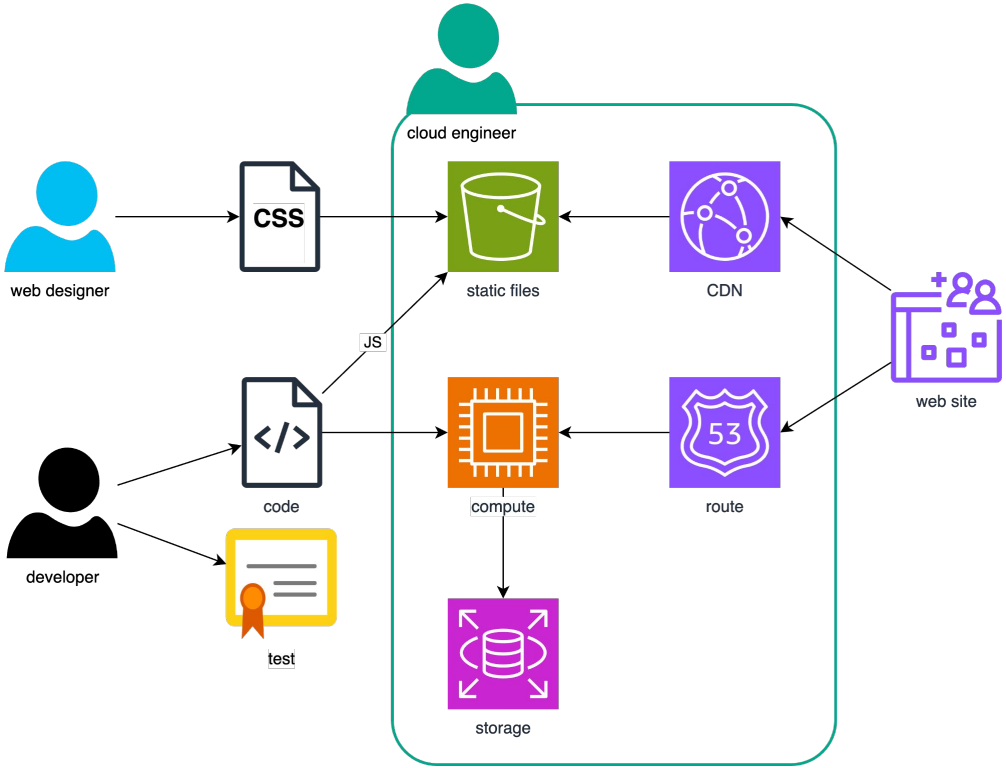
CI / CD & actors

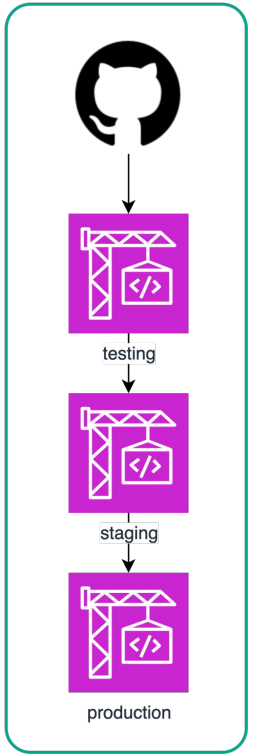

developer
code
test
cloud engineer
staging
bugfixing
bugfixing
test
production
monitoring, improvement & new features

CI / CD & actors

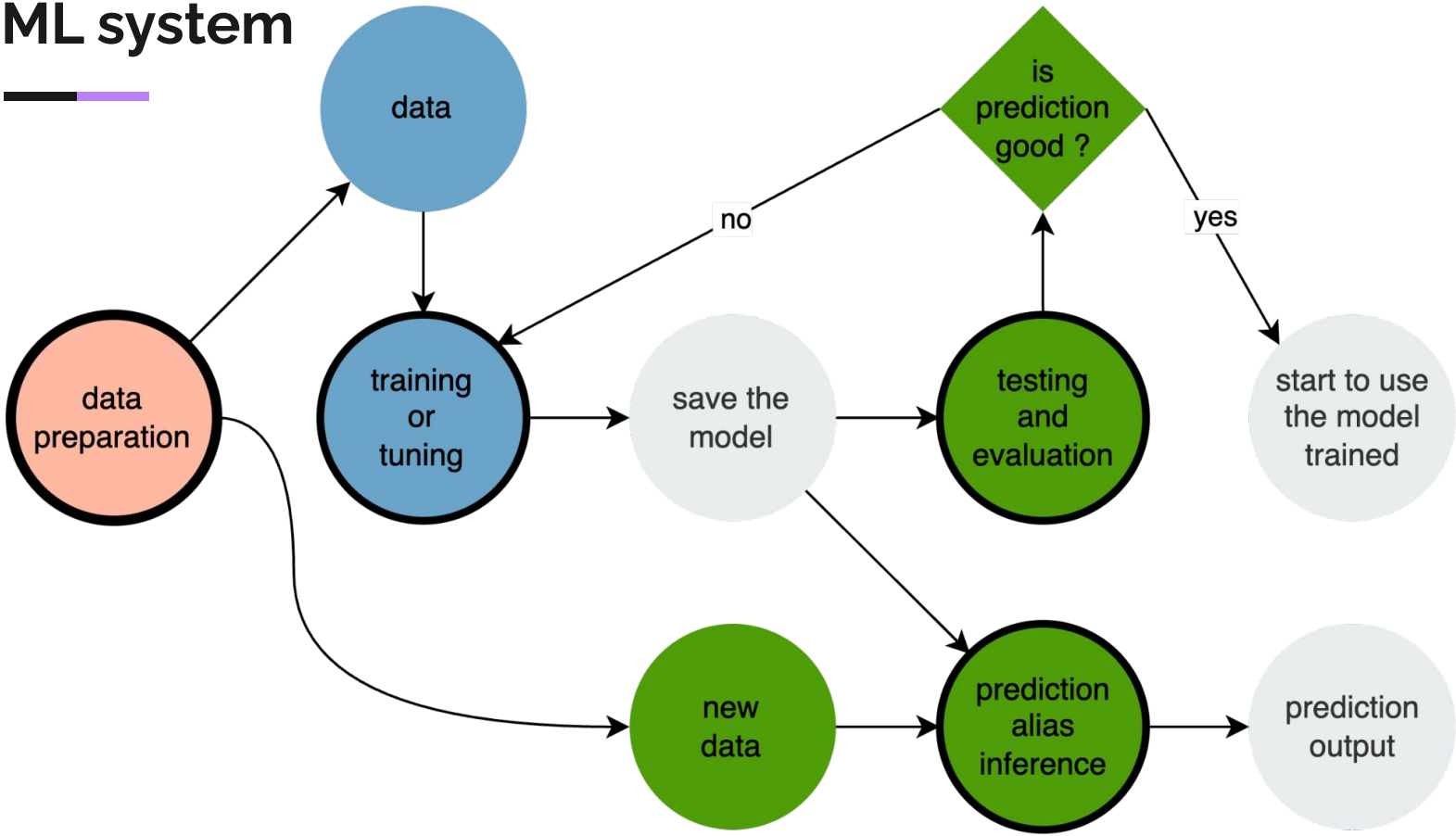

data scientist
code
test
cloud engineer
staging
bugfixing
bugfixing
test
production
monitoring, improvement & new features

DevOps methods & actors

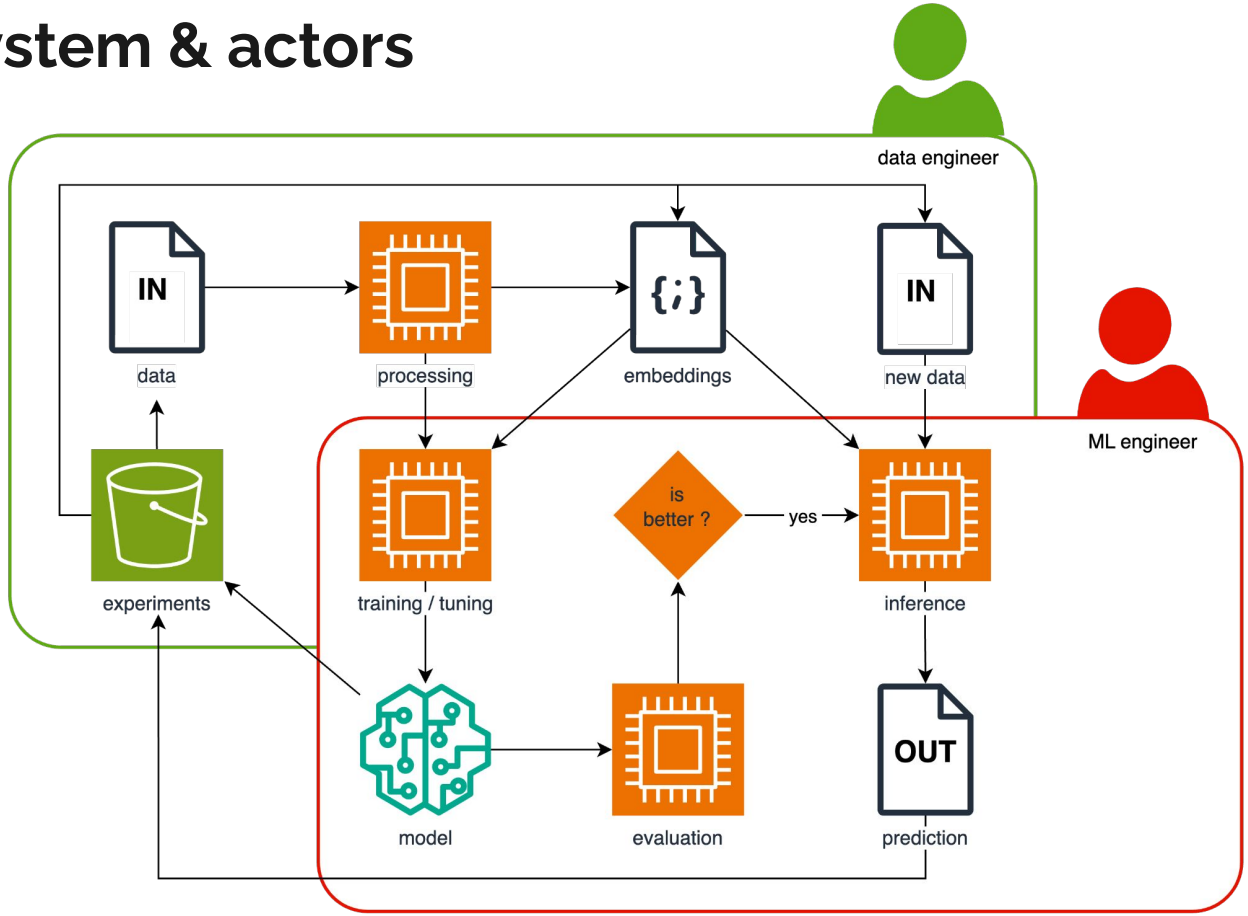




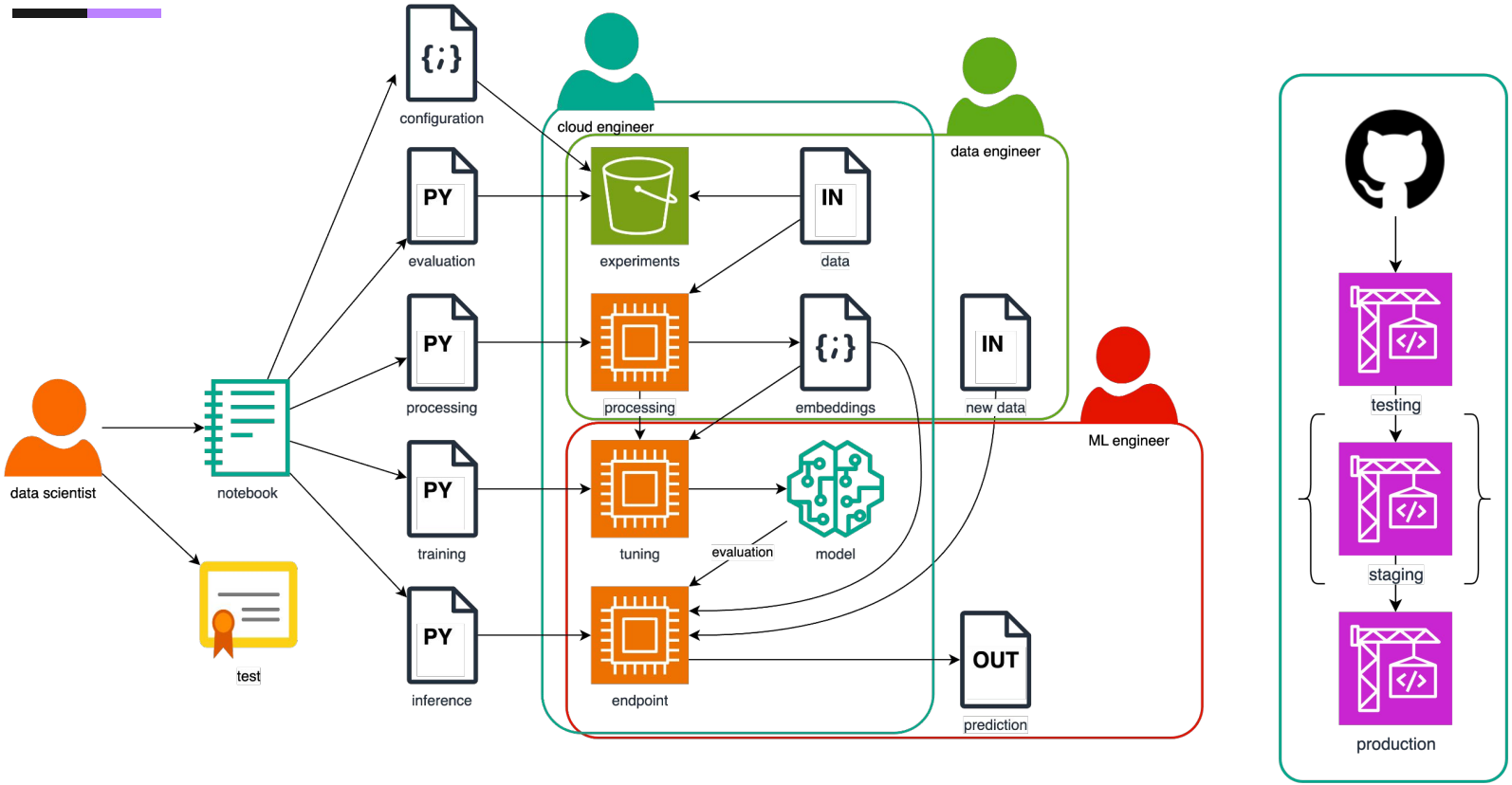
ML system


data
is prediction good ?
no
yes
data preparation
training or tuning
save the model
testing and evaluation
start to use the model trained
new data
prediction alias inference
prediction output

ML system & actors



MLOps methods & actors



reference

Solutions


mlflow


BentoML


METAFLOW


Pachyderm

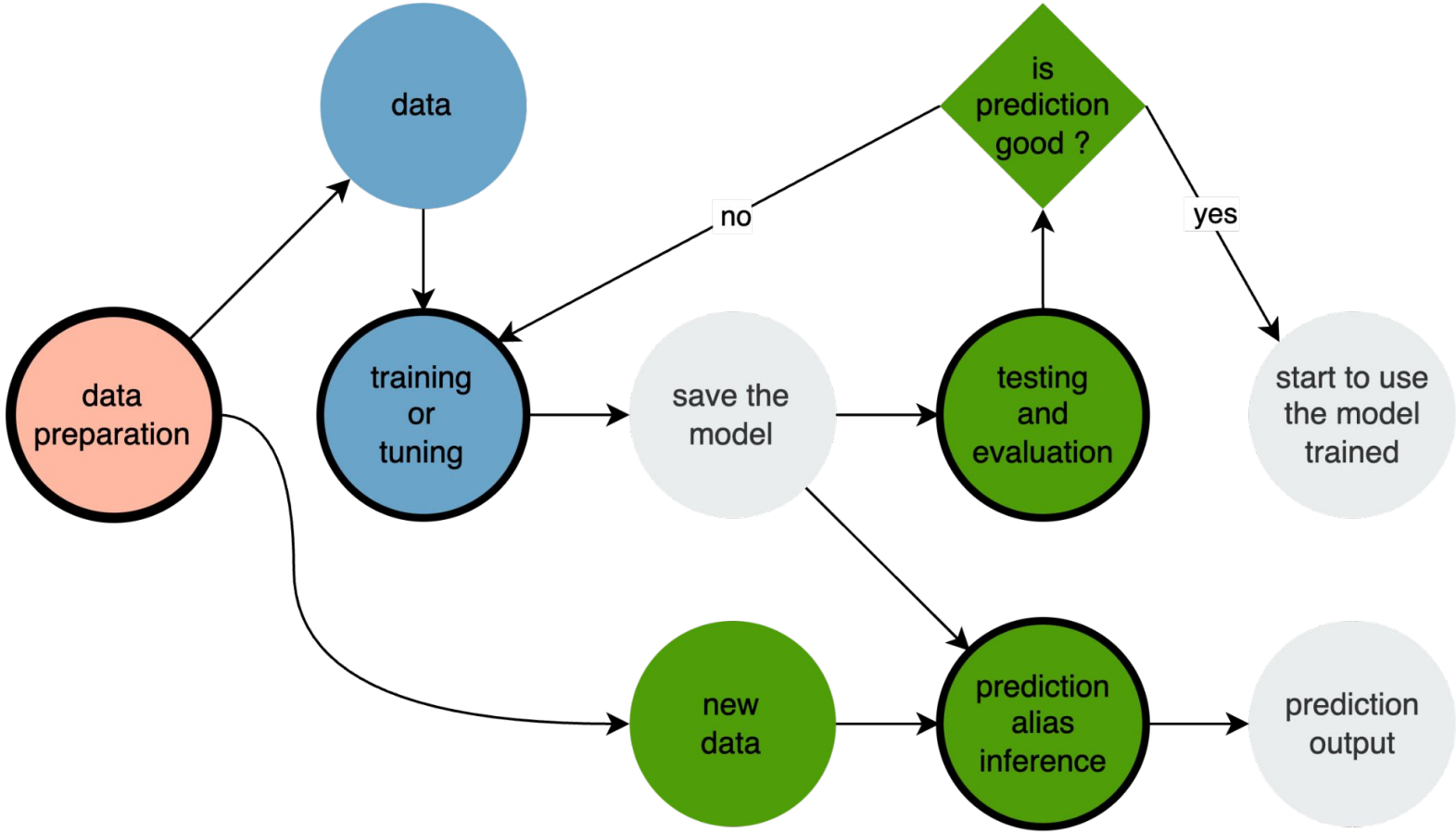

kedro

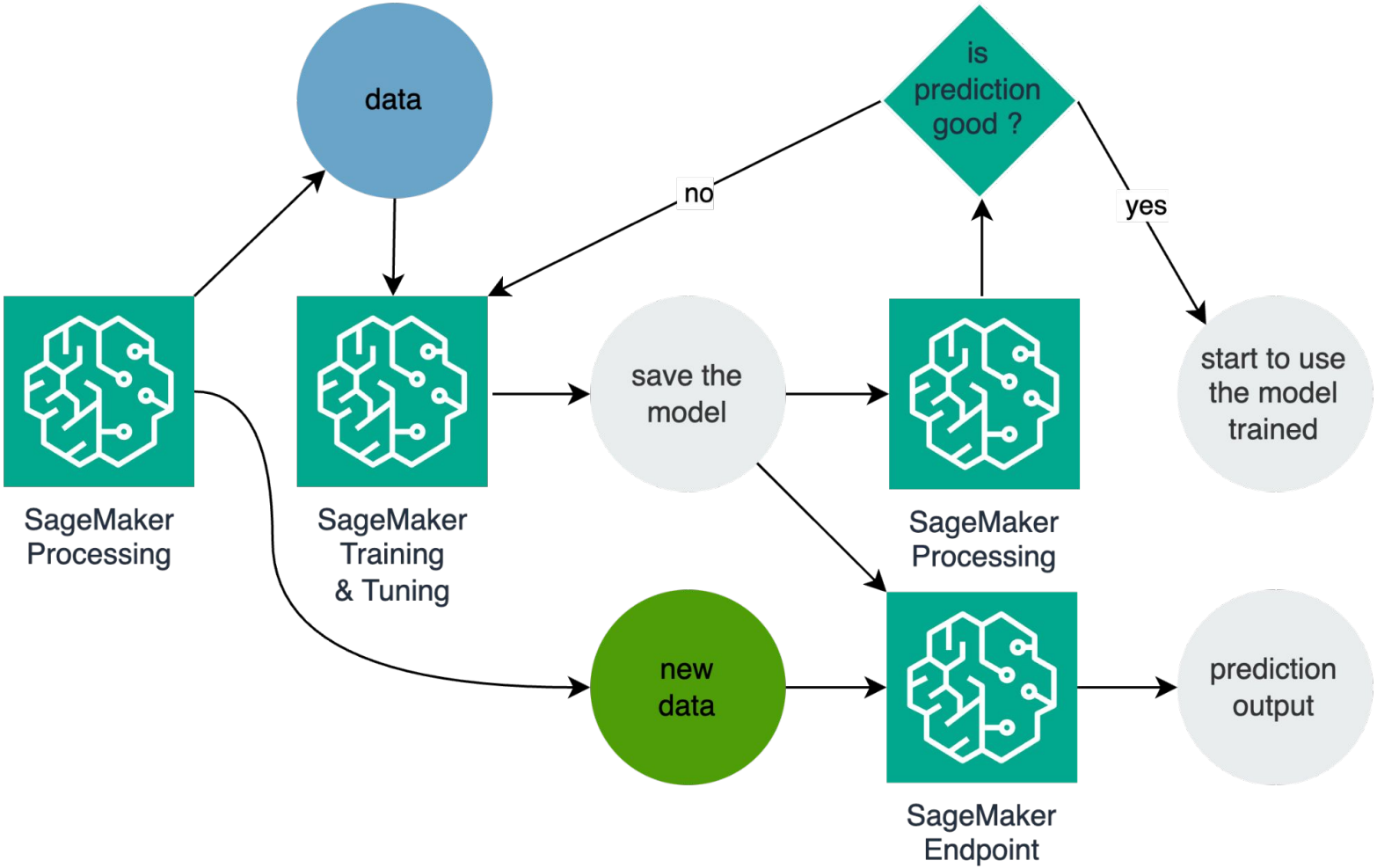

DVC

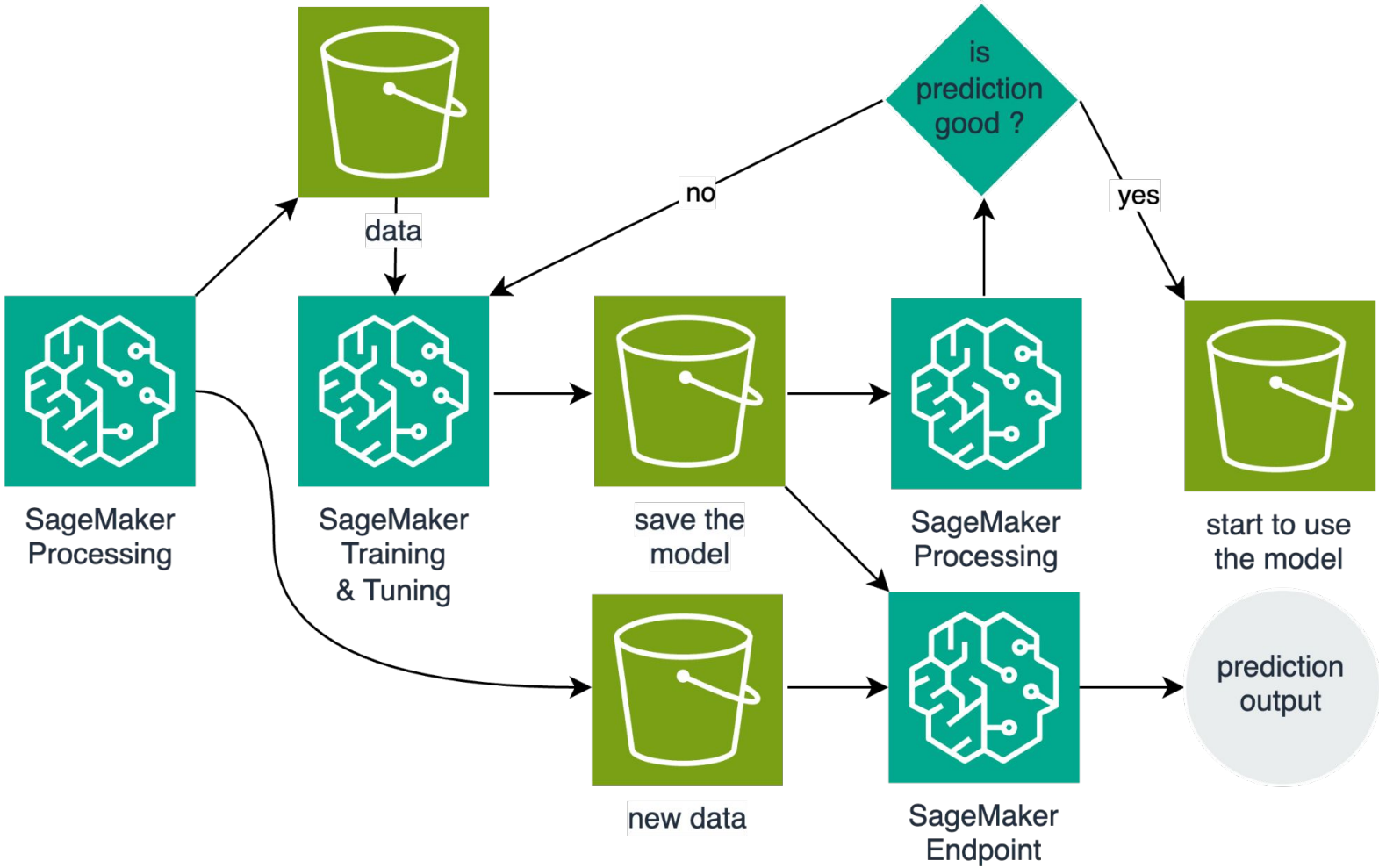

Kubeflow


deepchecks.

[reference]


data
is prediction good ?
no
yes
data preparation
training or tuning
save the model
testing and evaluation
start to use the model trained
new data
prediction alias inference
prediction output


data
is prediction good ?
no
yes
SageMaker Processing
SageMaker Training & Tuning
save the model
SageMaker Processing
start to use the model trained
new data
SageMaker Endpoint
prediction output


is prediction good ?
no
yes
data
SageMaker Processing
SageMaker Training & Tuning
save the model
SageMaker Processing
start to use the model
new data
SageMaker Endpoint
prediction output

```
import boto3
import sagemaker
from sagemaker.image_uris import retrieve

sess = sagemaker.Session()

region = boto3.Session().region_name

s3_output_path = \
    f"s3://{data_bucket}/{subfolder}/output"

container = retrieve(framework="pca", region=region)

pca = sagemaker.estimator.Estimator(
    container,
    role,
    instance_count=1,
    instance_type="ml.c4.xlarge",
    output_path=s3_output_path,
    sagemaker_session=sess,
)
pca.set_hyperparameters(
    feature_dim=np.shape(matrx_train)[1],
    num_components=10,
    subtract_mean=True,
    algorithm_mode="randomized",
    mini_batch_size=200,
)

pca.fit({"train": train_input, "test": test_input})
```

AWS Step Functions

```
from stepfunctions import steps
from stepfunctions.steps import TrainingStep, ModelStep, TransformStep
from stepfunctions.inputs import ExecutionInput
from stepfunctions.workflow import Workflow
from stepfunctions.template import TrainingPipeline
```

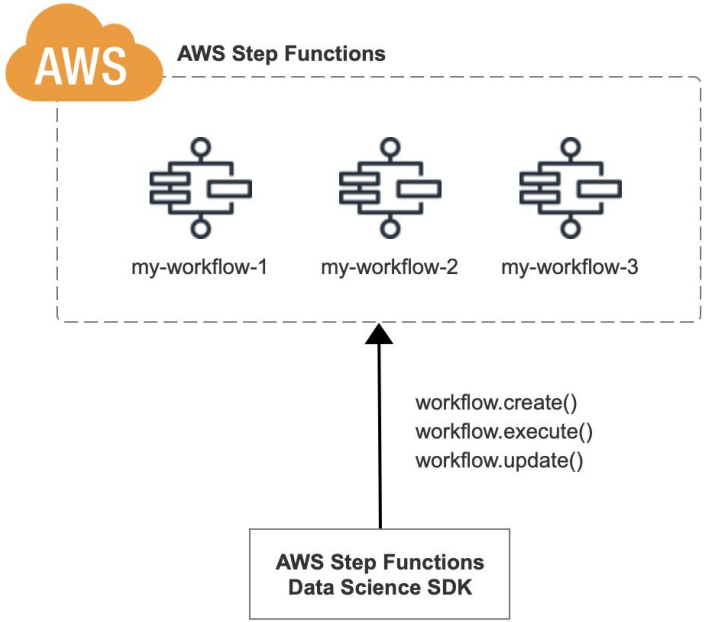
```
model_step = steps.ModelStep(
    "Save model", model=training_step.get_expected_model(), model_name=execution_input["ModelName"]
)
```

```
workflow_definition = steps.Chain(
    [transform_step, training_step, model_step, endpoint_config_step, endpoint_step]
)
```

```
workflow = Workflow(
    name="MyTrainTransformDeploy_v1",
    definition=workflow_definition,
    role=workflow_execution_role,
    execution_input=execution_input,
)
```

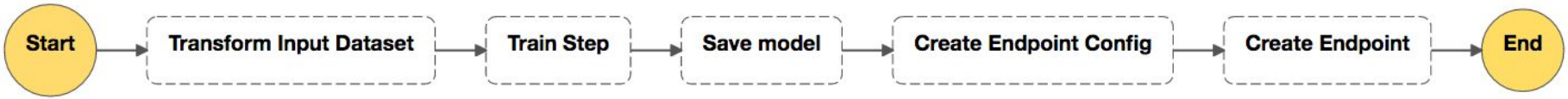
```
workflow.create()
```

AWS Step Functions

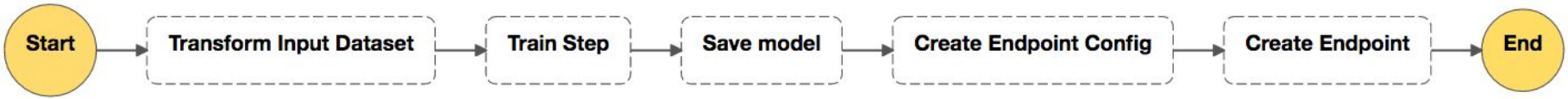


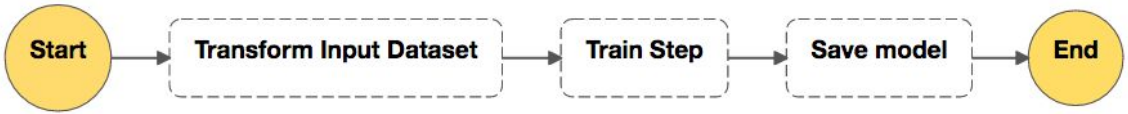
https://github.com/aws/aws-step-functions-data-science-sdk-python

Solutions


Start
Transform Input Dataset
Train Step
Save model
Create Endpoint Config
Create Endpoint
End

Solutions

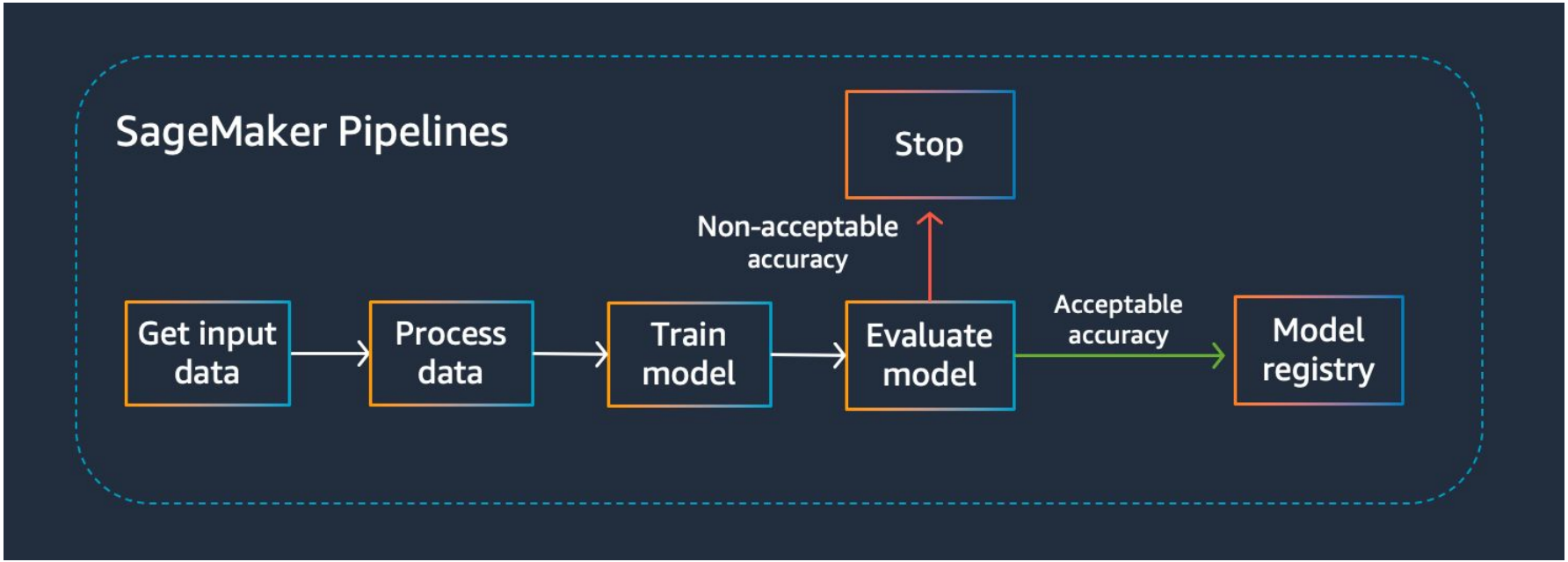

Start
Transform Input Dataset
Train Step
Save model
Create Endpoint Config
Create Endpoint
End


Start
Transform Input Dataset
Train Step
Save model
End

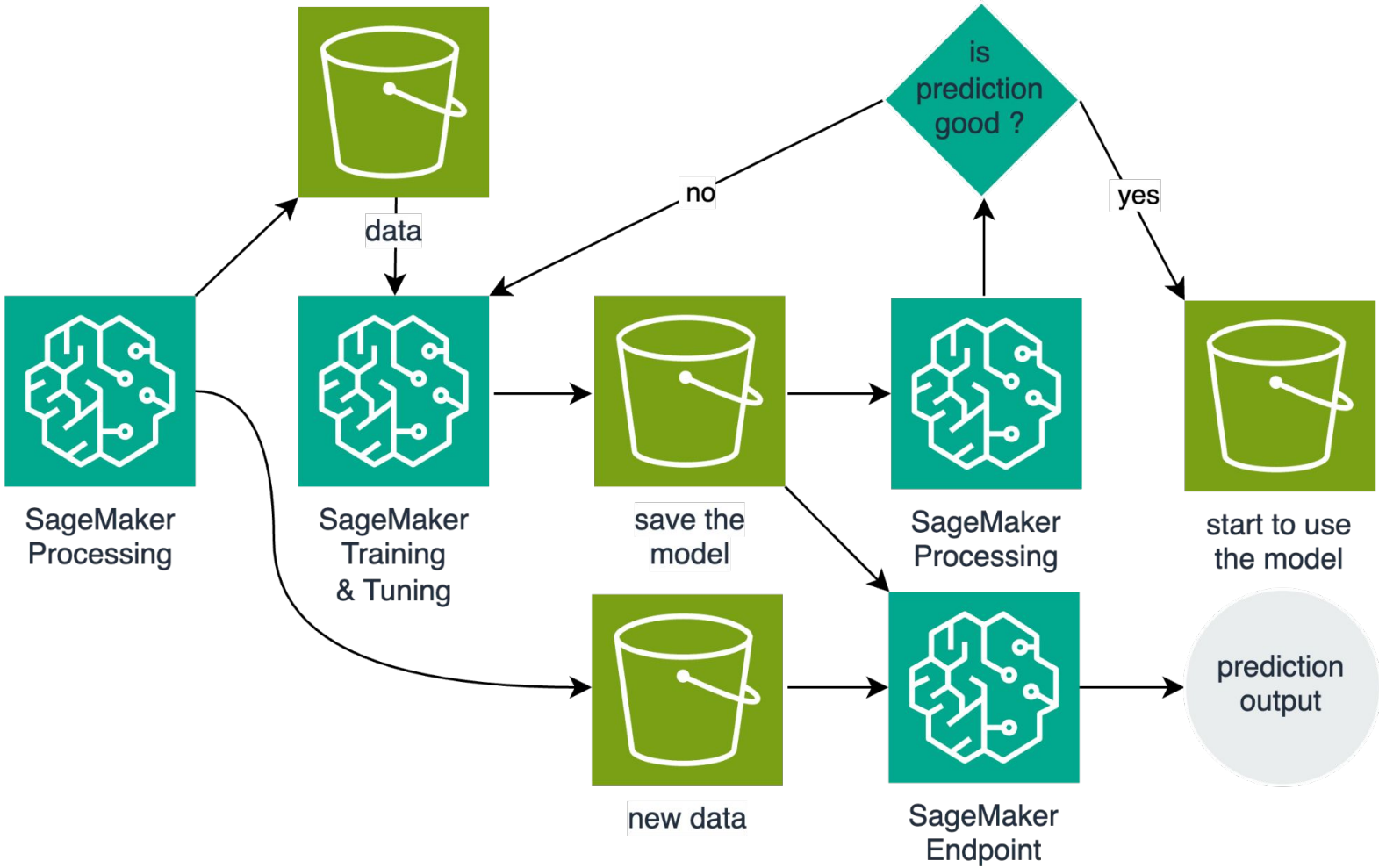
Amazon SageMaker Pipeline

```
from sagemaker.workflow.step_collections import RegisterModel
from sagemaker.workflow.condition_step import ConditionStep
from sagemaker.workflow.pipeline import Pipeline
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[
        processing_instance_type,
        training_instance_type,
        input_data,
        preprocess_script,
        evaluate_script,
        accuracy_condition_threshold,
        model_registry_package,
        max_parallel_training_jobs,
        max_training_jobs,
    ],
    steps=[step_preprocess_data, step_tuning, step_evaluate_model, step_cond],
)
pipeline.upsert(role_arn=role)
```

Amazon SageMaker Pipeline

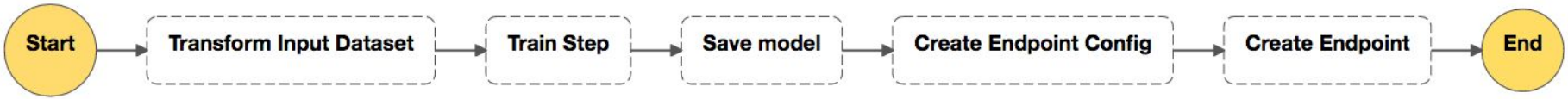


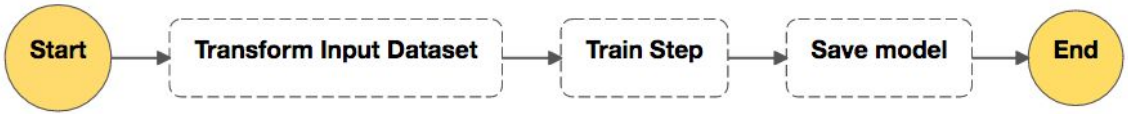
https://github.com/aws/amazon-sagemaker-examples/tree/main/sagemaker-pipeline-parameterization


is prediction good ?
no
yes
data
SageMaker Processing
SageMaker Training & Tuning
save the model
SageMaker Processing
start to use the model
new data
SageMaker Endpoint
prediction output

Take Away

Solutions


Start
Transform Input Dataset
Train Step
Save model
Create Endpoint Config
Create Endpoint
End


Start
Transform Input Dataset
Train Step
Save model
End

Step Functions vs SageMaker Pipeline

```
workflow_definition = steps.Chain(
    [transform_step, training_step, model_step, endpoint_config_step, endpoint_step]
)
```

```
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[
        processing_instance_type,
        training_instance_type,
        input_data,
        preprocess_script,
        evaluate_script,
        accuracy_condition_threshold,
        model_registry_package,
        max_parallel_training_jobs,
        max_training_jobs,
    ],
    steps=[step_preprocess_data, step_tuning, step_evaluate_model, step_cond],
)
```

AWS services

Which AWS services for which role, when we want to automate a solution from local to production: depends on the life cycle of the model but above all by the actors involved.

Data Scientists (DS)	Cloud Engineer (CE)	DS AWS Services	CE AWS Services
many	one	AWS Step Functions	AWS CloudFormation
one	many	Amazon SageMaker Pipeline	AWS CloudFormation & Step Functions

Data Scientist & Cloud Engineer

How to survive with colleagues with different skills, without hindering the table football game during the lunch break.

- Promoting resources awareness to be used
- Versioning of configuration, training and inference data statistics, models, ..
- Define tools and services: libraries, images, storages, ..
- Define guidelines for who does what: review, support, deploying the model, ..


Questions ?
@AWSUserGroupVenezia #4 #Meetup #AWS

Iniziative - AWS Community Day - Italy

https://www.awscommunityday.it/

- venerdì 27 settembre 2024
- location: Roma

Iniziative - ufficializzazione gruppo

- https://aws.amazon.com/developer/community/usergroups/
- https://www.meetup.com/pro/global-aws-user-group-community/

- https://aws-experience.com/emea/italia/external-event/d52d3f8f-523d-491b-9020-30475736223b
- missing - https://awsusergroup.it/

Prossimo incontro

- giovedì 28 novembre ore 19:00

Prossimo incontro

- giovedì 28 novembre ore 19:00

AWS User Group Venezia compie 1 anno ad ottobre 🎉

Proposte


Spritz


Thanks for participating.
@AWSUserGroupVenezia #4 #Meetup #AWS