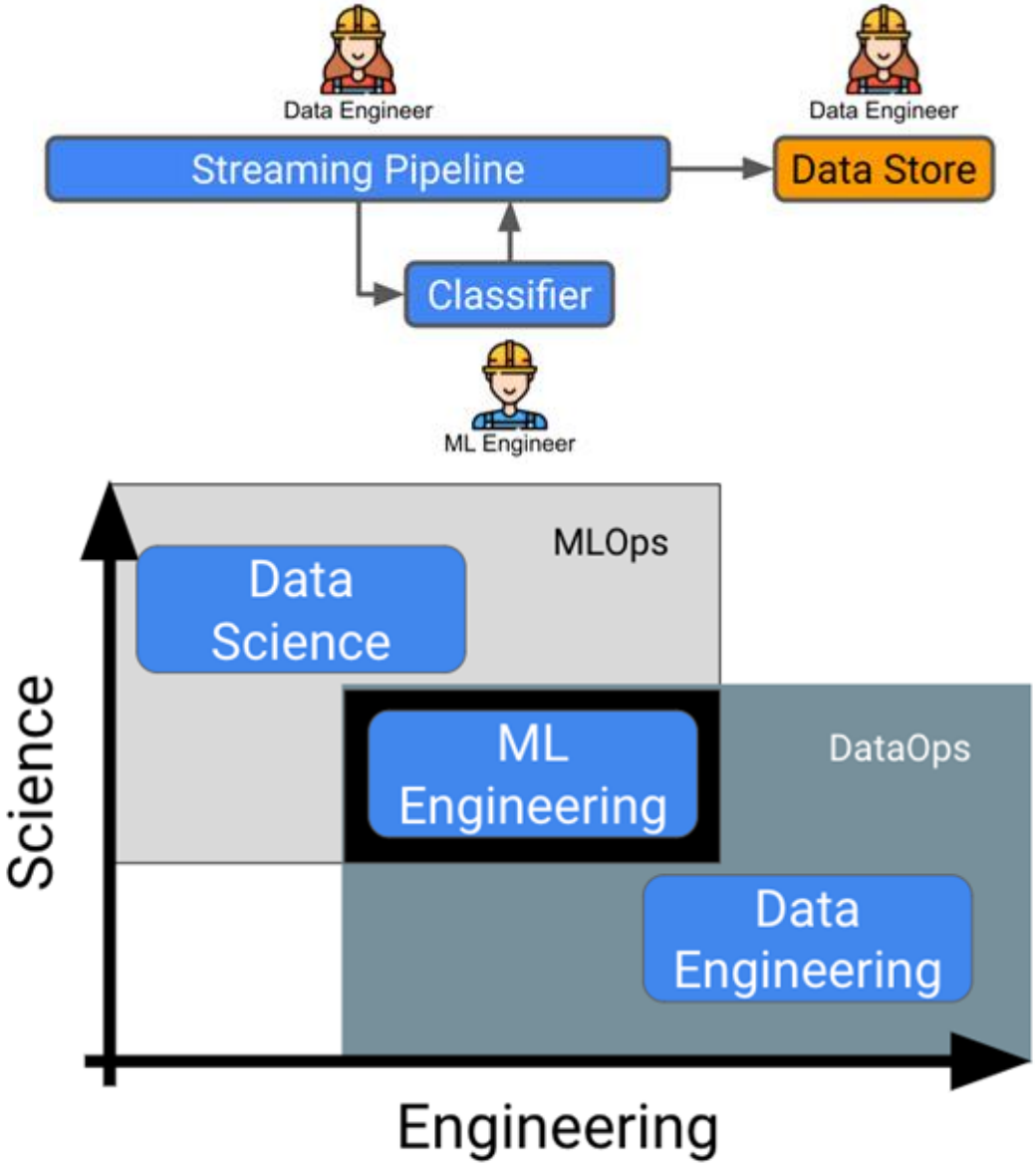
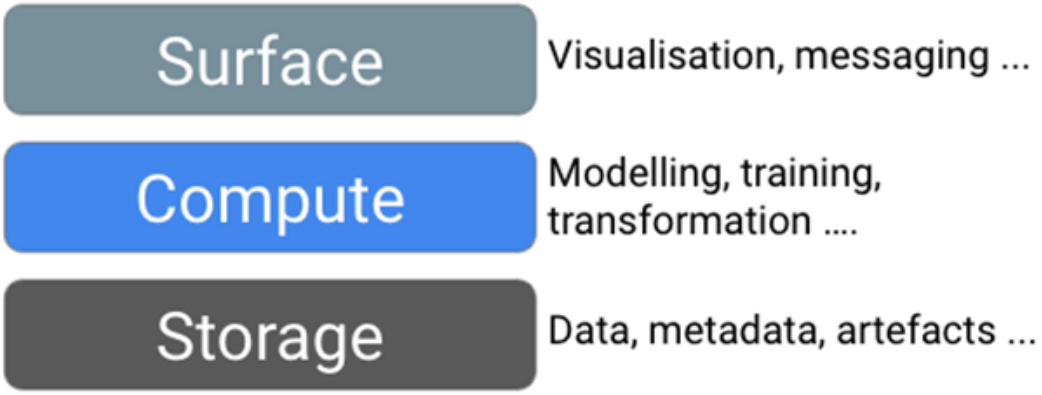
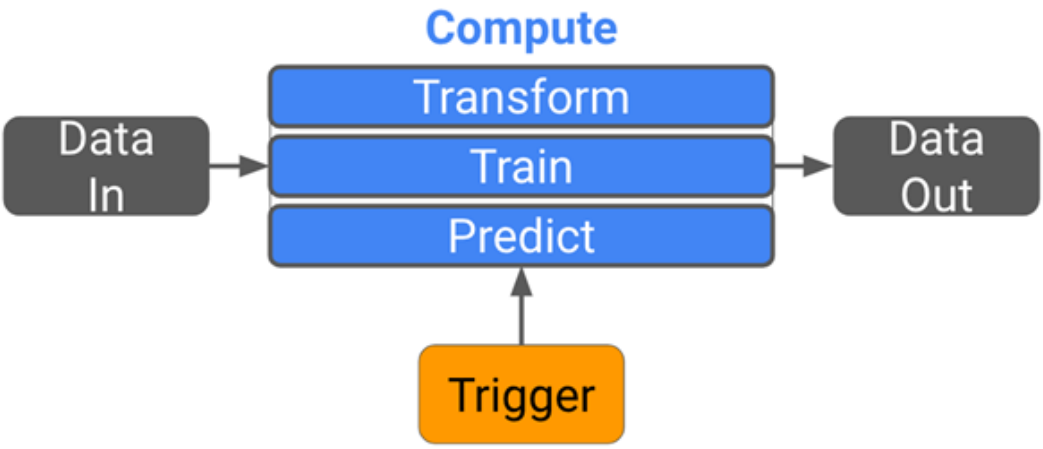
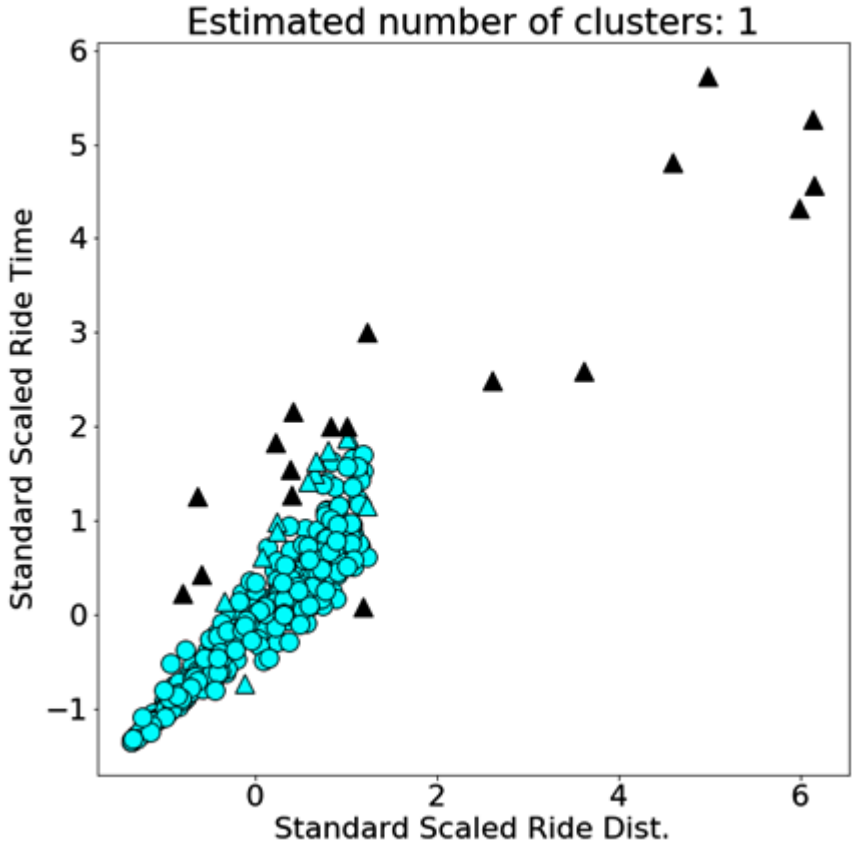
Chapter 1: Introduction to ML Engineering

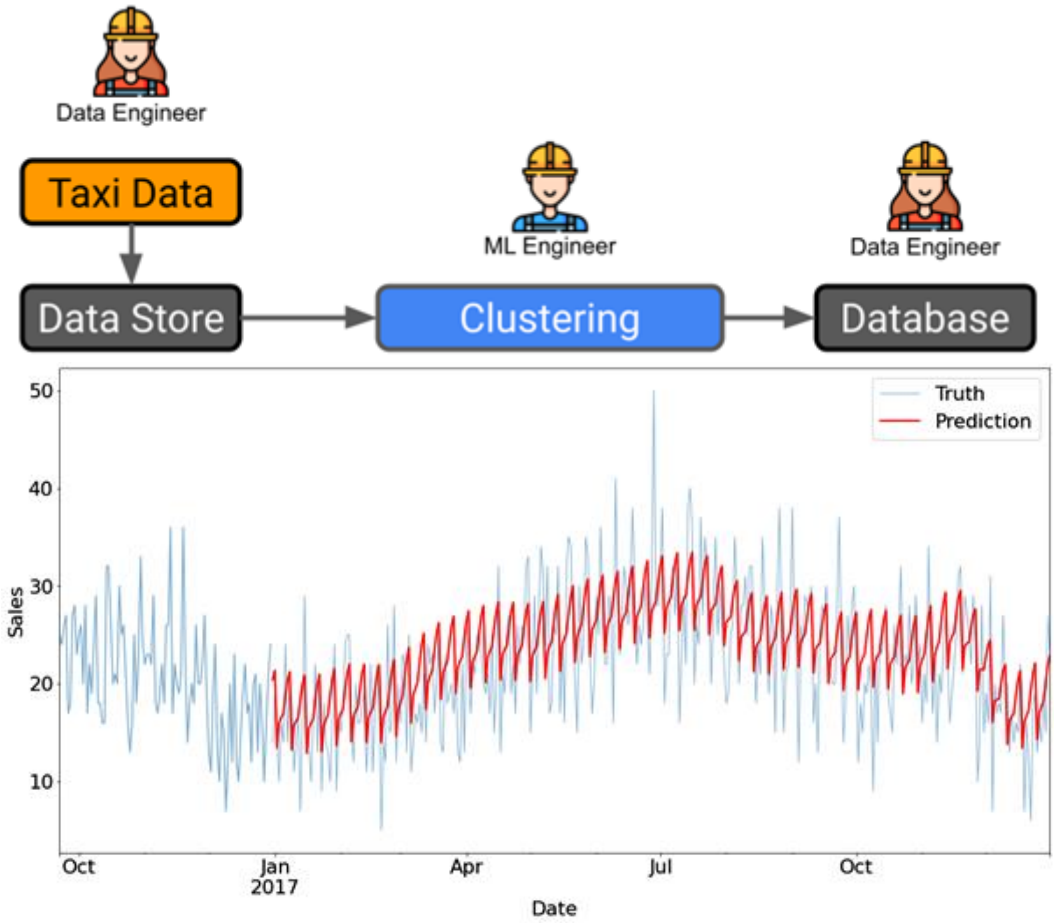


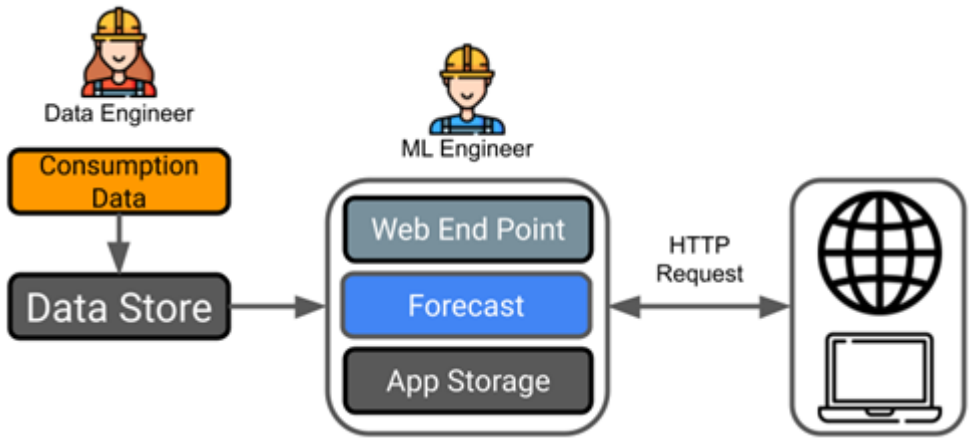
Requirement	Is ML Appropriate?	Details
Anomaly detection of energy pricing signals	Yes	You will want to do this on large numbers of points on potentially varying time signals.
Improving data quality in an ERP system	No	This sounds more like a process problem. You can try and apply ML to this but often it is better to make the data entry process more automated or the process more robust.
Forecasting item consumption for a warehouse	Yes	ML will be able to do this more accurately than a human can, so this is a good area of application.
Summarizing data for business reviews	Maybe	This can be required at scale but is not an ML problem - simple queries against your data will do.



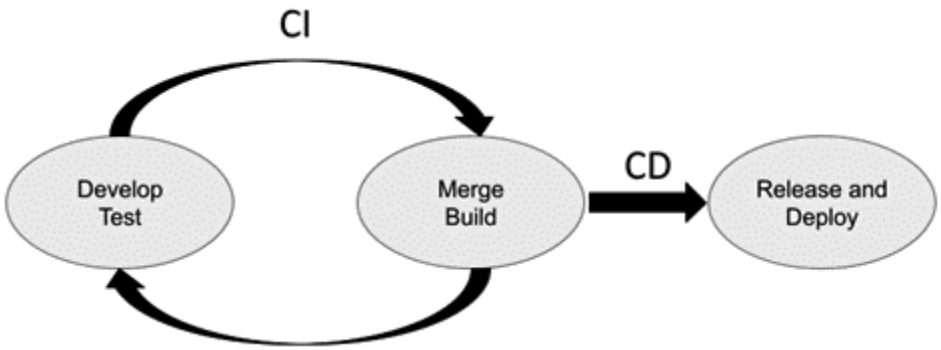

Compute
Transform
Train
Predict
Data In
Data Out
Trigger

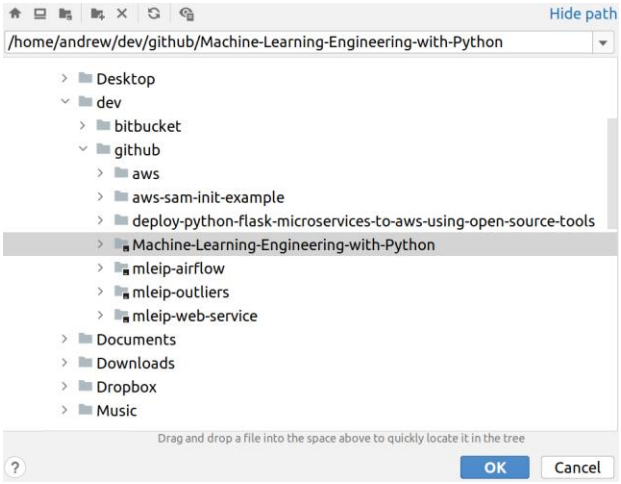

Estimated number of clusters: 1
Standard Scaled Ride Time
Standard Scaled Ride Dist.
6
5
4
3
2
1
0
−1
0
2
4
6


Data Engineer
Taxi Data
Data Store
ML Engineer
Clustering
Data Engineer
Database
Truth
Prediction
Sales
50
40
30
20
10
Oct
Jan
2017
Apr
Jul
Oct
Date


Data Engineer
Consumption Data
Data Store
ML Engineer
Web End Point
Forecast
App Storage
HTTP Request

Chapter 2: The Machine Learning Development Process





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * **Repository name** *

AndyMc629 /

Great repository names are short and memorable. Need inspiration? How about **bookish-goggles**?

Description (optional)

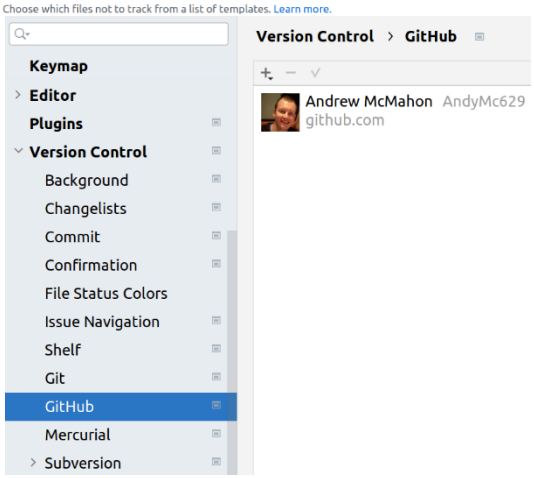
Public
Anyone on the internet can see this repository. You choose who can commit.

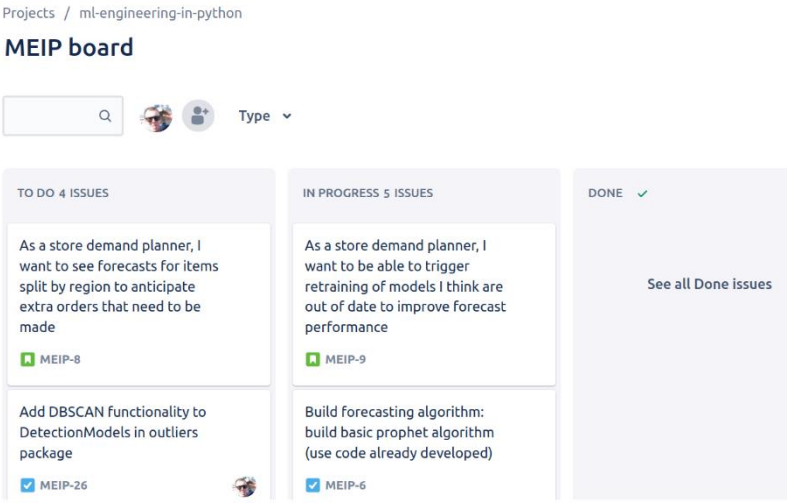
Private
You choose who can see and commit to this repository.

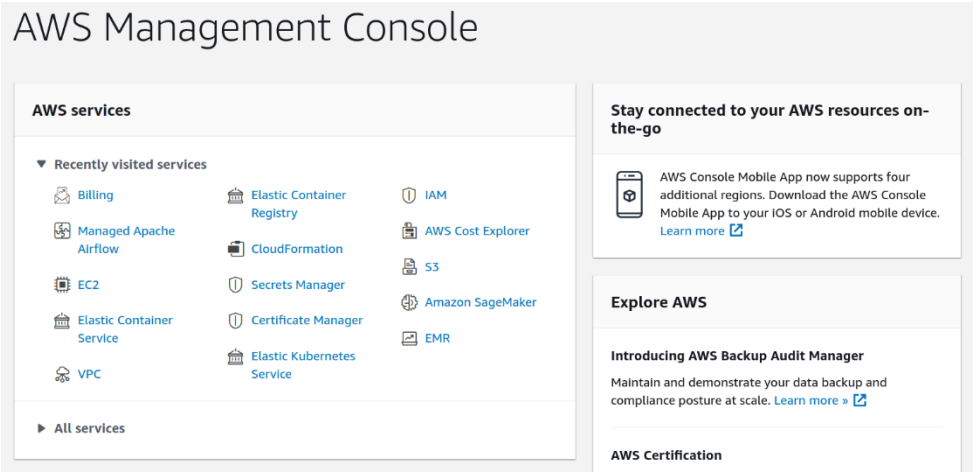
Initialize this repository with:
Skip this step if you're importing an existing repository.

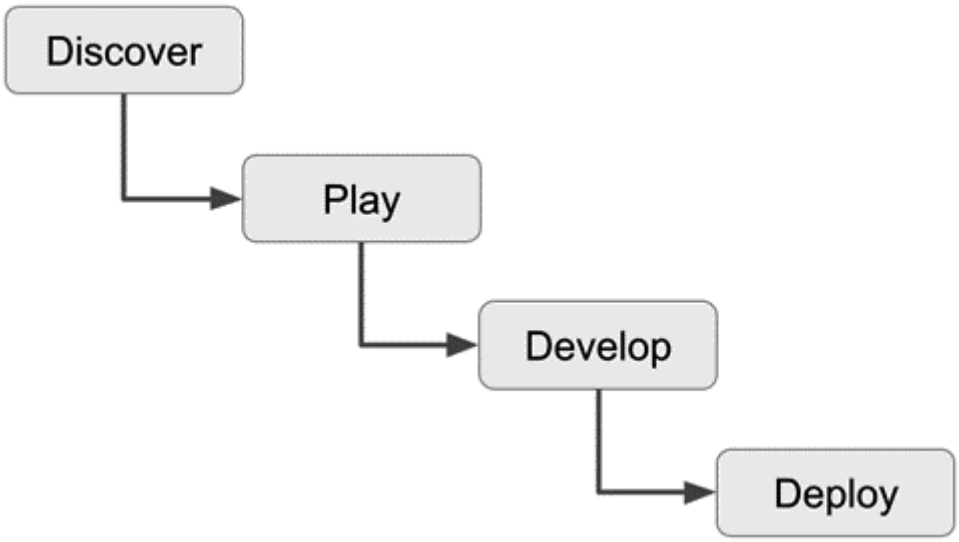
Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

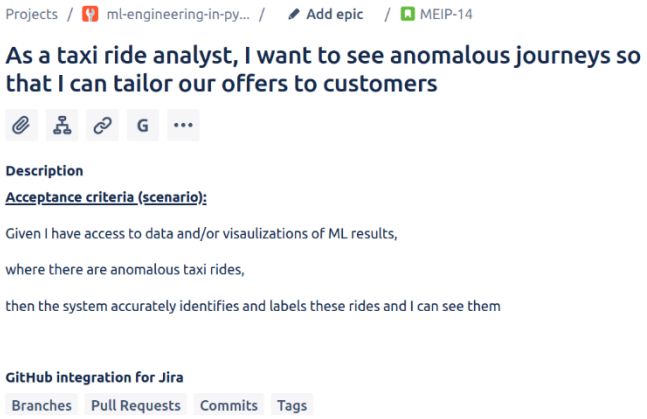



Projects / ml-engineering-in-python
MEIP board
Type
TO DO 4 ISSUES
As a store demand planner, I want to see forecasts for items split by region to anticipate extra orders that need to be made
MEIP-8
Add DBSCAN functionality to DetectionModels in outliers package
MEIP-26
IN PROGRESS 5 ISSUES
As a store demand planner, I want to be able to trigger retraining of models I think are out of date to improve forecast performance
MEIP-9
Build forecasting algorithm: build basic prophet algorithm (use code already developed)
MEIP-6
DONE
See all Done issues


AWS Management Console
AWS services
Recently visited services
Billing
Managed Apache Airflow
EC2
Elastic Container Service
VPC
Elastic Container Registry
CloudFormation
Secrets Manager
Certificate Manager
Elastic Kubernetes Service
IAM
AWS Cost Explorer
S3
Amazon SageMaker
EMR
All services
Stay connected to your AWS resources on-the-go
AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your IOS or Android mobile device.
Learn more
Explore AWS
Introducing AWS Backup Audit Manager
Maintain and demonstrate your data backup and compliance posture at scale. Learn more »
AWS Certification



Stage	Outputs
Discover	Clarity on the business question Clear arguments for ML over another approach Definition of the KPIs and metrics you want to optimize A sketch of the route to value
Play	Detailed understanding of the data Working proof of concept Agreement on the model/algorithm/logic that will solve the problem Evidence that a solution is doable within realistic resource scenarios Evidence that good ROI can be achieved
Develop	A working solution that can be hosted on appropriate and available infrastructure Thorough test results and performance metrics (for algorithms and software) An agreed retraining and model deployment strategy Unit tests, integration tests, and regression tests Solution packaging and pipelines
Deploy	A working and tested deployment process Provisioned infrastructure with appropriate security and performance characteristics Mode retraining and management processes An end-to-end working solution!



Prep for Prophet

```
df.rename(columns= {'Datetime': 'ds', 'AEP_MW': 'y'}, inplace=True)
```

```
df['ds']=df['ds'].astype('datetime64[ns]')
```

```
df.dtypes
```

```
#Initialize Split Class, we'll split our data 5 times for cv
ts_splits = TimeSeriesSplit(n_splits=5)
```

Train and Forecast

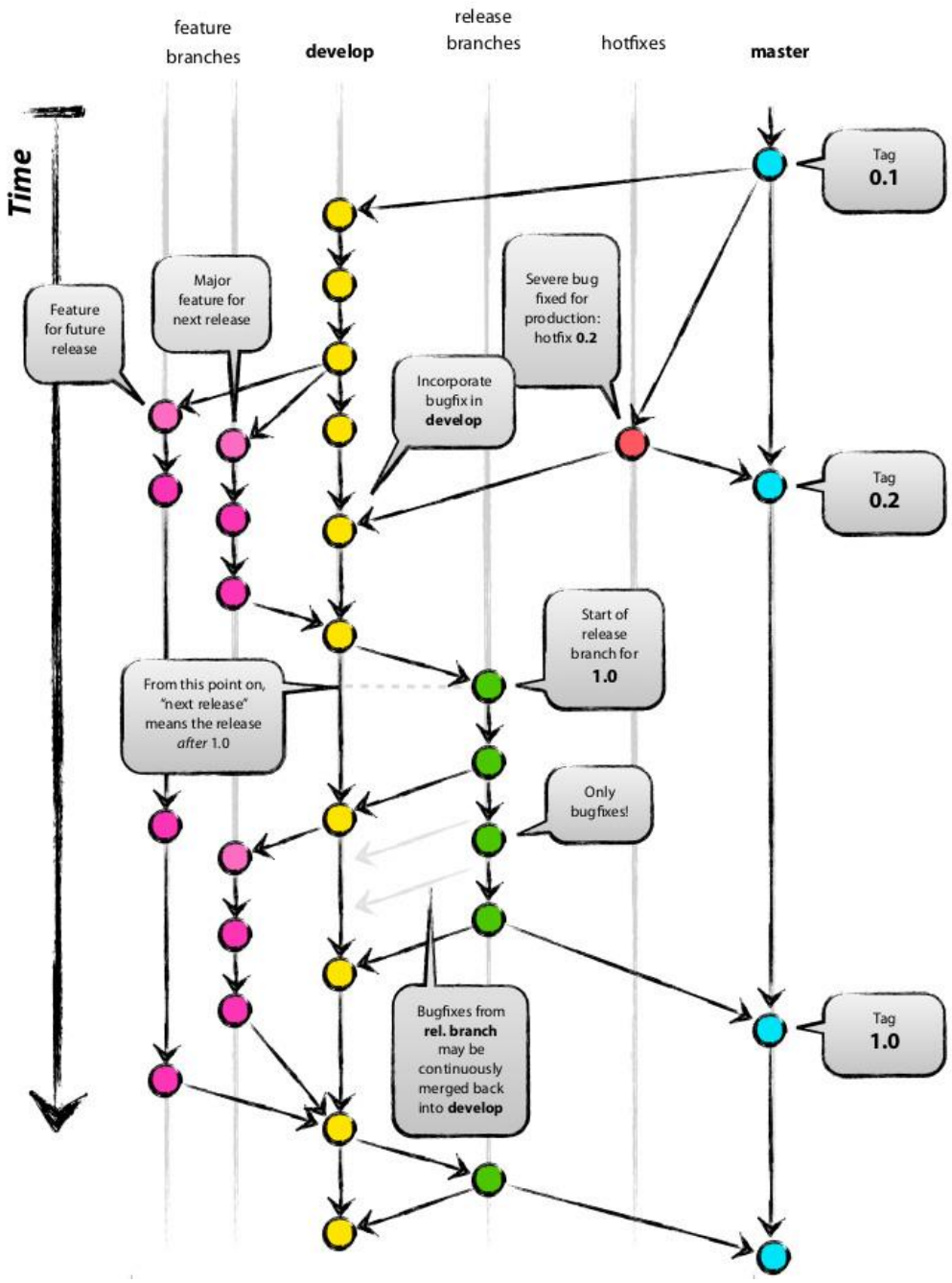
```
tmp = time_split_train_test(df.sort_values('ds', ascending=True).iloc[-1000:], ts_splits)
```

```
tmp.head()
```

Plot

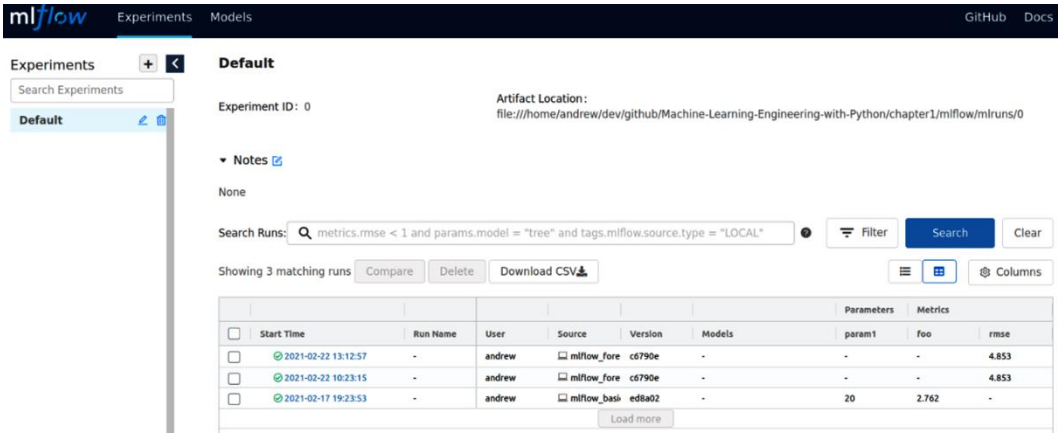
```
nrow = 5; ncol = 1;
fig, axs = plt.subplots(nrows=nrow, ncols=ncol, figsize=(20,30))
fig.subplots_adjust(hspace=0.4, wspace=0.4)
for i, ax in enumerate(fig.axes):
    split_rmse = tmp[(tmp['split']==i) & (tmp['train']==False)]['rmse'].iloc[0]

    ax.set_title('Split '+str(i)+' - RMSE: '+"{:.2f}".format(split_rmse))

    tmp[(tmp['split']==i) & (tmp['train']==True)].plot(x='ds', y='y', ax=ax, color='blue', marker='o')
    tmp[(tmp['split']==i) & (tmp['train']==False)].plot(x='ds', y='y', ax=ax, color='red', marker='o')
    tmp[(tmp['split']==i) & (tmp['train']==False)].plot(x='ds', y='yhat', ax=ax, color='orange', marker='^')
```

Methodology	Pros	Cons
Agile	Flexibility is expected. Faster dev to deploy cycles.	If not well managed, can easily have scope drift. Kanban or Sprints may not work well for some projects.
Waterfall	Clearer path to deployment. Clear staging and ownership of tasks.	Lack of flexibility. Higher admin overheads.


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

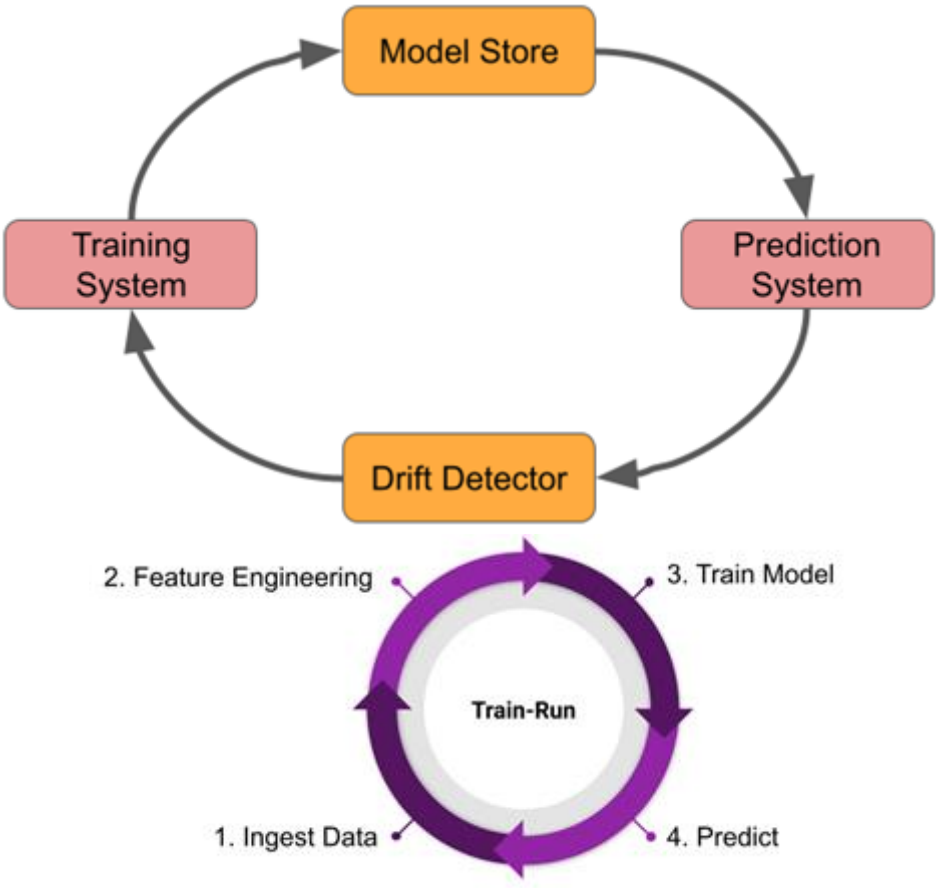
9 chapter1/bad-git/pipeline.py

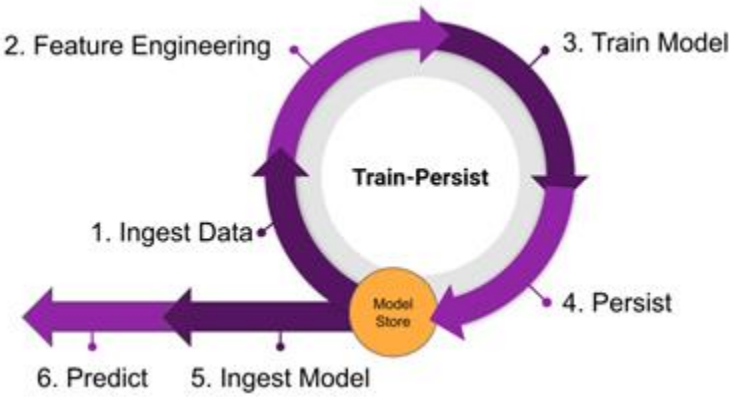
```
@@ -1,9 +1,14 @@
- # EXAMPLE BELOW TAKEN FROM THE SPARK API DOCS, BEFORE BEING UPDATE FOR THE BOOK
+ # EXAMPLE BELOW ADAPTED FROM THE SPARK DOCS
  # https://spark.apache.org/docs/latest/ml-pipeline.html#pipeline
  from pyspark.ml import Pipeline
  from pyspark.ml.classification import LogisticRegression
  from pyspark.ml.feature import HashingTF, Tokenizer

+ import json
+
+ with open("model_config.json") as f:
+     model_config = json.load(f)
+
  # Prepare training documents from a list of (id, text, label) tuples.
  training = spark.createDataFrame([
      (0, "a b c d e spark", 1.0),
@@ -15,7 +20,7 @@
  # Configure an ML pipeline, which consists of three stages: tokenizer, hashingTF, and lr.
  tokenizer = Tokenizer(inputCol="text", outputCol="words")
  hashingTF = HashingTF(inputCol=tokenizer.getOutputCol(), outputCol="features")
- lr = LogisticRegression(maxIter=10, regParam=0.001)
+ lr = LogisticRegression(maxIter=model_config['maxIter'], regParam=model_config['regParam'])
  pipeline = Pipeline(stages=[tokenizer, hashingTF, lr])
```

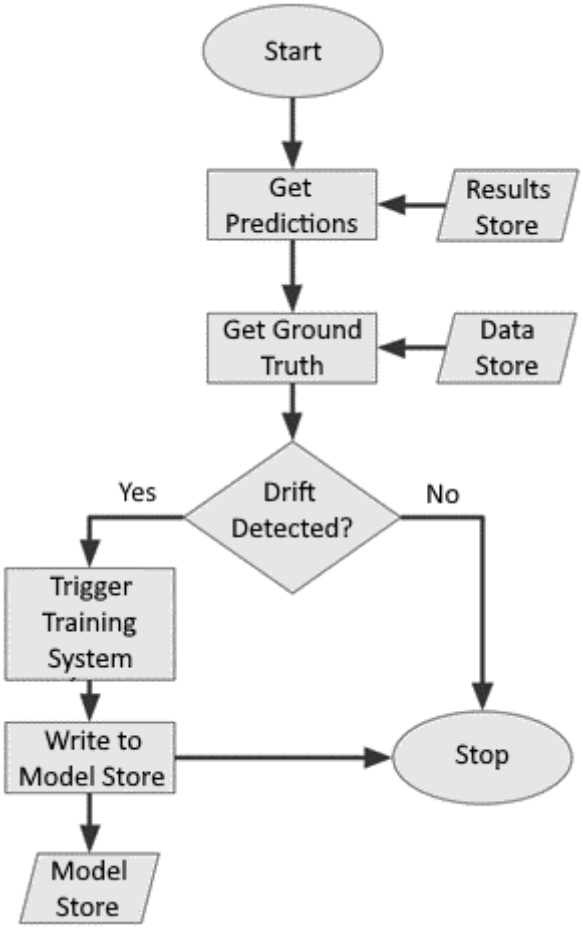


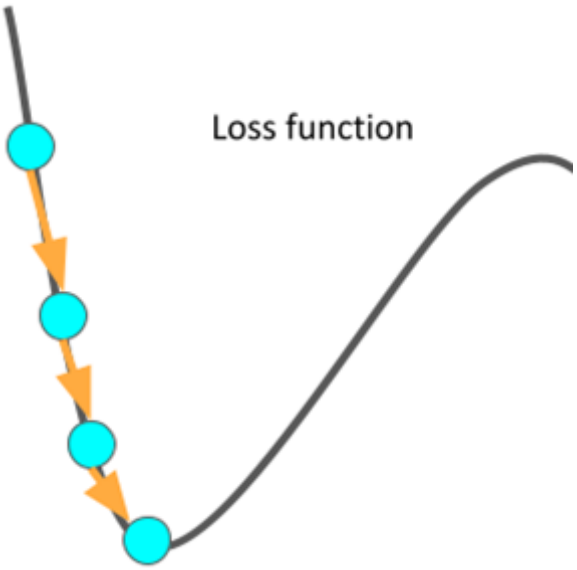
Chapter 3: From Model to Model Factory

Algorithm	Hyperparameters	What This Controls
K-Nearest Neighbors	• **K** • Distance metric	• The number of clusters • How to define the distance between points
DBSCAN	• **Epsilon** • Minimum number of samples • Distance metric	• The max distance to be considered **neighbors** • How many neighbors are required to be considered **core** • How to define the distance between points




2. Feature Engineering
3. Train Model
Train-Persist
1. Ingest Data
Model Store
4. Persist
6. Predict
5. Ingest Model


Start
Get Predictions
Results Store
Get Ground Truth
Data Store
Drift Detected?
Yes
No
Trigger Training System
Write to Model Store
Stop
Model Store



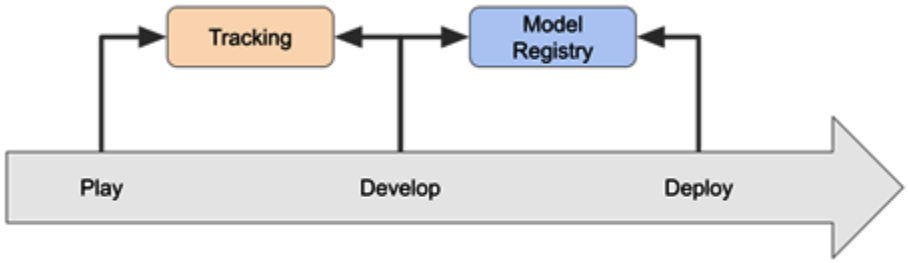
AutoML
Perform automated selection of the best hyperparameters and the best ML algorithm. Provide only a high level steer of this process.

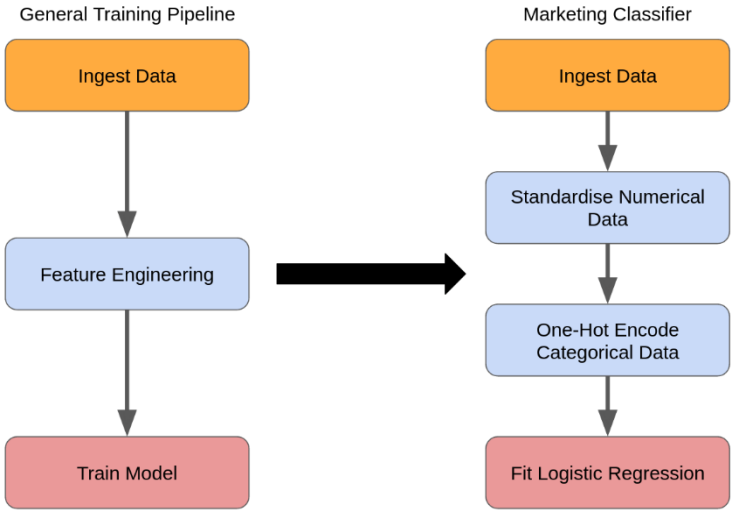
Hyperparameter Optimization
Tune the parameters that tell your ML algorithm how to learn. Use tools that allow for efficient search over hyperparameter spaces.

Hand Cranking
Define the hyperparameters for learning explicitly and select one ML algorithm to try at a time.

Algorithm	Hyperparameters	What This Controls
Decision Trees & Random Forests	• Tree depth • Min/max leaves	• How many levels in your trees • How much 'branching' can occur at each level
Support Vector Machines	• 'C' • 'Gamma'	• Penalty for misclassification • Radius of influence of training points for Radial Basis Function (RBF) kernels
Neural Networks (numerous architectures)	• Learning rate • Number of hidden layers • Activation function • Many more	• Update step sizes • How 'deep' your net is • The 'firing' conditions of your neurons
Logistic Regression	• Solver • Regularization type • Regularization prefactor	• How to minimise the loss • How to prevent overfitting/make problem well behaved • The amount to do this

Algorithm	Hyperparameters	What This Controls
K-Nearest Neighbors	• "K" • Distance metric	• The number of clusters • How to define distance between points
DBSCAN	• "Epsilon" • Minimum number of samples • Distance metric	• The max distance to be considered 'neighbors' • How many neighbors required to be considered 'core' • How to define distance between points

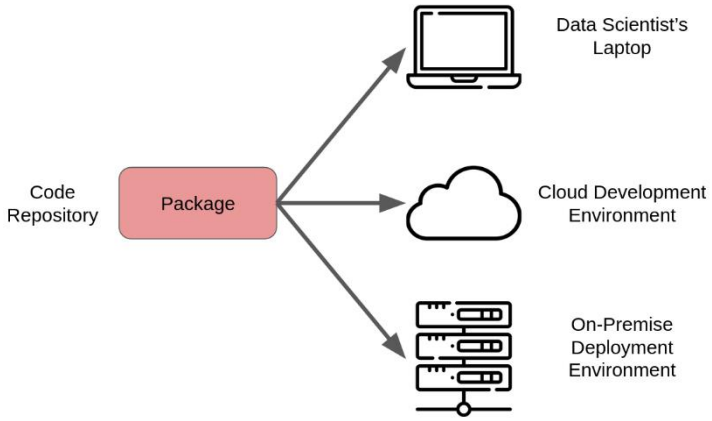




Algorithm	Hyperparameters	What This Controls
Decision Trees and Random Forests	• Tree depth • Min/max leaves	• How many levels are in your trees • How much **branching** can occur at each level
Support Vector Machines	• **C** • **Gamma**	• Penalty for misclassification • The radius of influence of the training points for **Radial Basis Function** (**RBF**) kernels
Neural Networks (numerous architectures)	• Learning rate • Number of hidden layers • Activation function • Many more	• Update step sizes • How **deep** your network is • The *firing* conditions of your neurons
Logistic Regression	• Solver • Regularization type • Regularization prefactor	• How to minimize the loss • How to prevent overfitting/make the problem well behaved • The *strength* of the regularization type

Chapter 4: Packaging Up

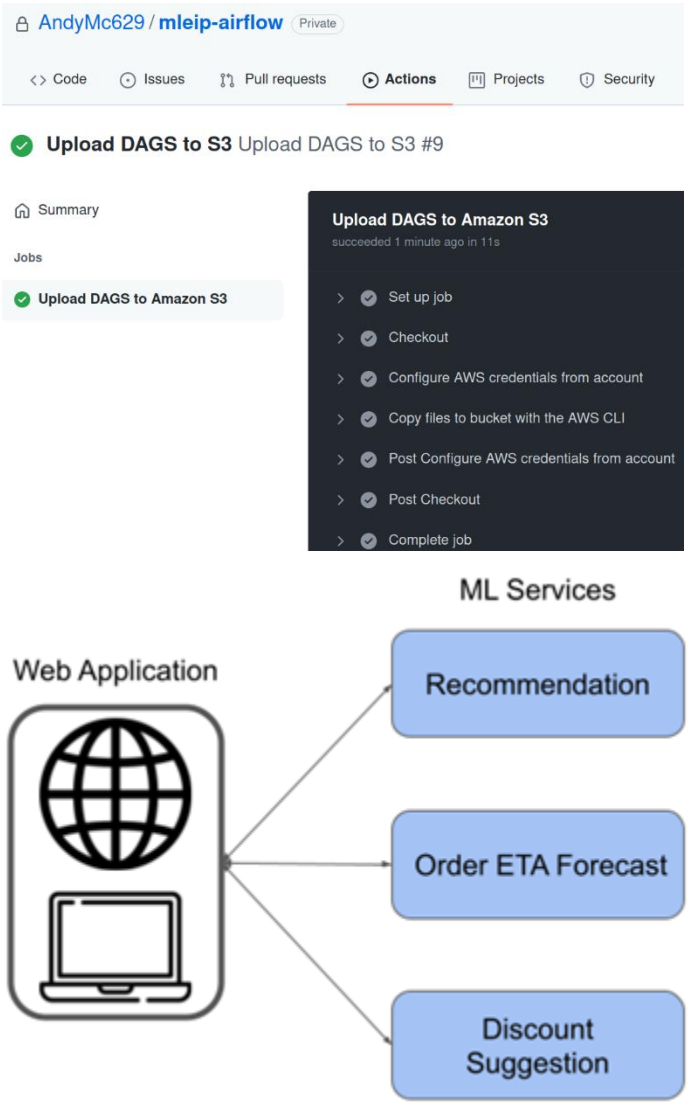
Container	Description
`deque`	This is a double-ended queue and allows you to add and remove elements to either end of the object in a scalable way. It's useful if you want to add to the beginning or end of large data lists or if you want to search for the last occurrences of X in your data.
`Counter`	Counters take in iterables such as dicts or lists and return the count of each of the elements. They're really useful to get quick summaries of the content of these objects.
`OrderedDict`	The standard `dict` object does not maintain order, so `OrderedDict` introduces this functionality. This can be really useful if you need to loop back over a dictionary you have created in the same order as it was created for new processing.

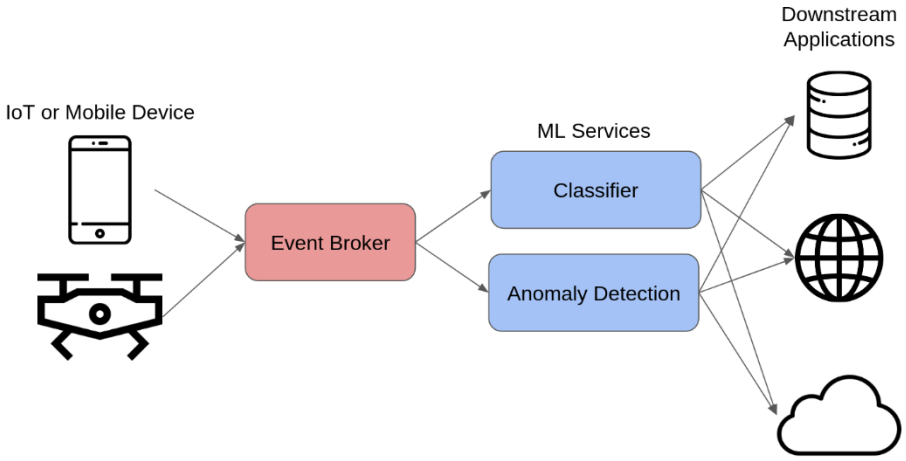



```
============================================================================================ test session starts ======
platform linux -- Python 3.8.5, pytest-6.1.1, py-1.9.0, pluggy-0.13.1 -- /home/andrew/anaconda3/envs/mleng/bin/python
cachedir: .pytest_cache
rootdir: /home/andrew/dev/github/Machine-Learning-Engineering-with-Python/chapter4/outlier_package
collected 2 items

outliers/tests/test_create_data.py::test_data_is_numpy PASSED
outliers/tests/test_create_data.py::test_data_is_large PASSED

============================================================================================= 2 passed in 0.45s =======
```

```
============================================================================================ test session starts ======
platform linux -- Python 3.8.5, pytest-6.1.1, py-1.9.0, pluggy-0.13.1 -- /home/andrew/anaconda3/envs/mleng/bin/python
cachedir: .pytest_cache
rootdir: /home/andrew/dev/github/Machine-Learning-Engineering-with-Python/chapter4/outlier_package
collected 5 items

outliers/tests/test_create_data.py::test_data_is_numpy PASSED
outliers/tests/test_create_data.py::test_data_is_large PASSED
outliers/tests/test_detectors.py::test_model_creation PASSED
outliers/tests/test_detectors.py::test_model_get_models PASSED
outliers/tests/test_detectors.py::test_model_evaluation PASSED
```

Chapter 5: Deployment Patterns and Tools





MLEIP / **MLEIP-web-service-basic**

POST http://localhost:5000/forecast

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
"store_number": 55,
"forecast_start_date": "2021-10-10T00:00:00"
}
```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "result": [
        0.5119601362414734,
        0.048732590944116194,
        0.6843997801791688,
        0.8694554790764257,
        0.6549083843755388,
        0.7591360074029415,
        0.02741600455310511,
        0.9855584281616281,
        0.054214771538483975,
        0.3851831190248214
    ],
    "store_number": 55
}
```

```
IMAGE ID       CREATED AT                      REPOSITORY
cfc60d3d8055   2021-08-22 21:28:21 +0100 BST   basic-ml-webservice
cfc60d3d8055   2021-08-22 21:28:21 +0100 BST   508972911348.dkr.ecr.eu-west-1.amazonaws.com/basic-ml-microservice
cfc60d3d8055   2021-08-22 21:28:21 +0100 BST   basic-ml-microservice
c369d19a3e8f   2021-08-22 21:28:03 +0100 BST   <none>
4e8f4d160116   2021-08-07 19:40:11 +0100 BST   ml-microservice
06f5a53dccfb   2021-08-07 19:39:52 +0100 BST   <none>
9d753150d71e   2021-08-07 19:32:29 +0100 BST   <none>
f68dc7fd2a66   2021-08-07 19:32:16 +0100 BST   <none>
dd8edfcc5a84   2021-08-07 19:09:58 +0100 BST   <none>
0589078a401c   2021-08-07 19:09:45 +0100 BST   <none>
aa27d063bdae   2021-07-13 16:09:49 +0100 BST   inferencefunction
b5712d3d8d03   2021-07-13 16:05:55 +0100 BST   508972911348.dkr.ecr.eu-west-2.amazonaws.com/mleip-lambda-example-repo
b5712d3d8d03   2021-07-13 16:05:55 +0100 BST   inferencefunction
e6f9b123f9a0   2021-07-13 13:52:37 +0100 BST   public.ecr.aws/lambda/python
```

Linux x86 (64-bit) | Linux ARM

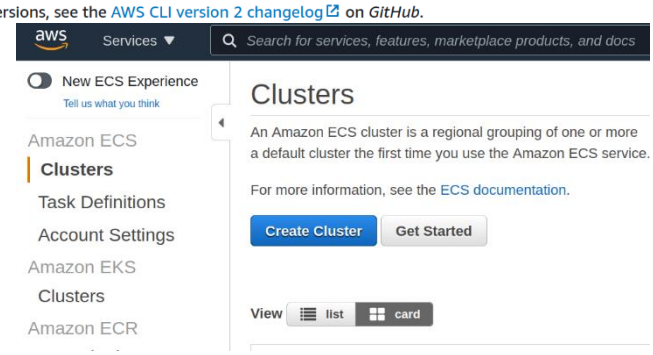
For the latest version of the AWS CLI, use the following command block:

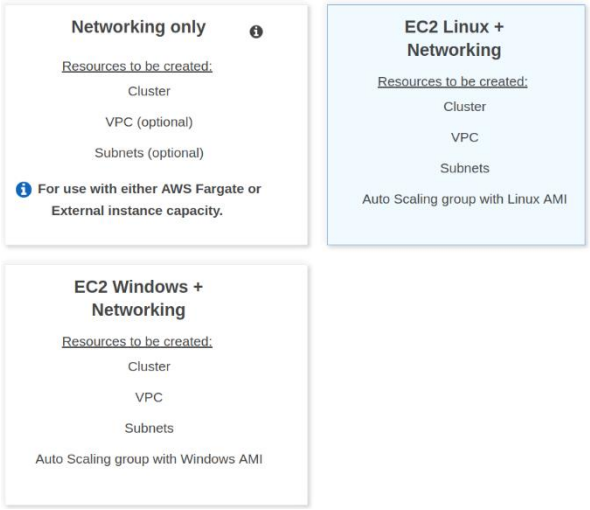
```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

For a specific version of the AWS CLI, append a hyphen and the version number to the filename. For this example the filename for version *2.0.30* would be `awscli-exe-linux-x86_64-2.0.30.zip` resulting in the following command:

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

For a list of versions, see the AWS CLI version 2 changelog on *GitHub*.





Instance configuration

Provisioning Model (•) On-Demand Instance

With On-Demand Instances, you pay for compute capacity by the hour, with no long-term commitments or upfront payments.

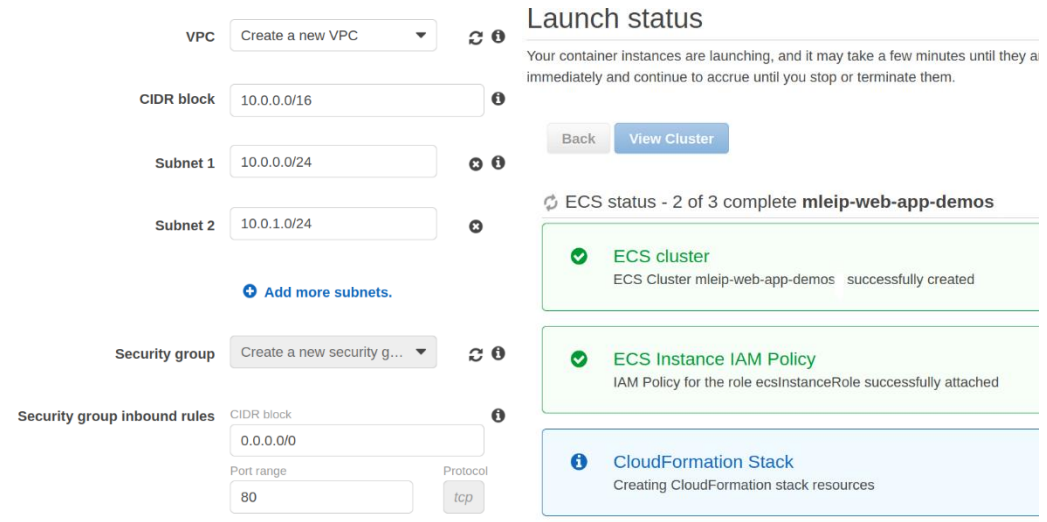
( ) Spot

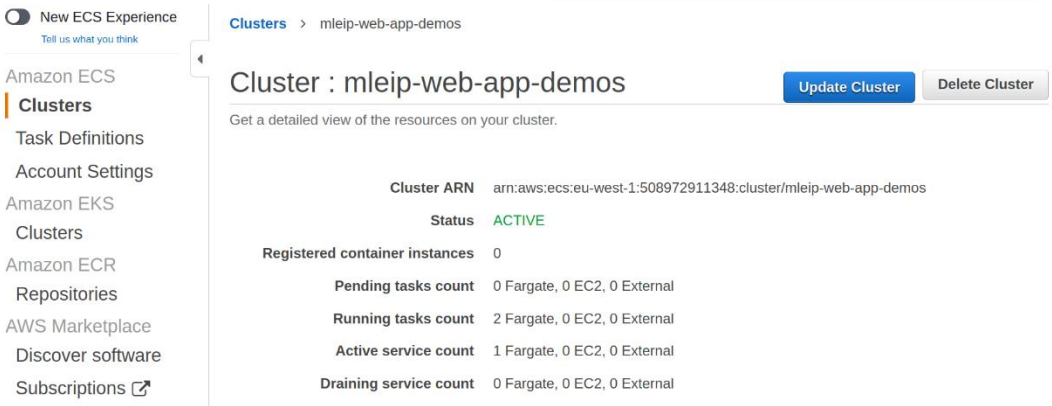
Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud. Spot Instances are available at up to a 90% discount compared to On-Demand prices. Learn more

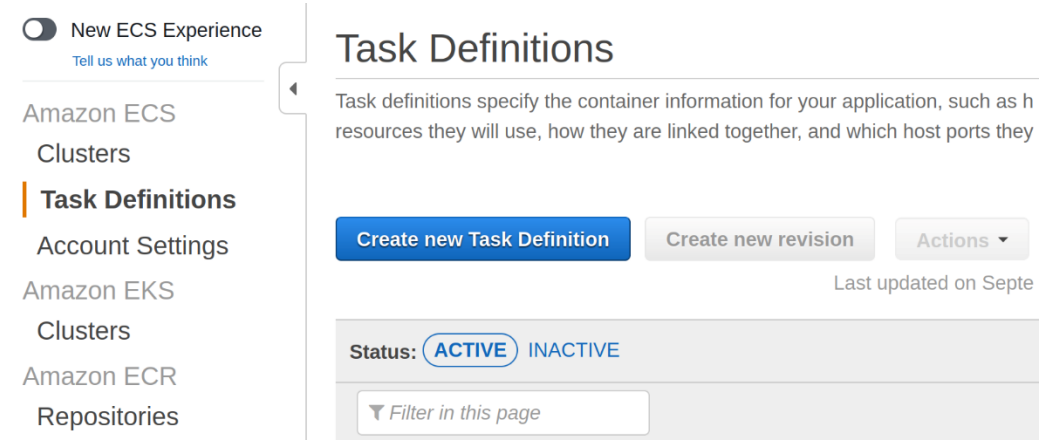
EC2 instance type* t2.micro

[ ] Manually enter desired instance type

Number of instances* 1


VPC
Create a new VPC
CIDR block
10.0.0.0/16
Subnet 1
10.0.0.0/24
Subnet 2
10.0.1.0/24
Add more subnets.
Security group
Create a new security g…
Security group inbound rules
CIDR block
0.0.0.0/0
Port range
80
Protocol
tcp
Launch status
Your container instances are launching, and it may take a few minutes until they a
immediately and continue to accrue until you stop or terminate them.
Back
View Cluster
ECS status - 2 of 3 complete mleip-web-app-demos
ECS cluster
ECS Cluster mleip-web-app-demos successfully created
ECS Instance IAM Policy
IAM Policy for the role ecsInstanceRole successfully attached
CloudFormation Stack
Creating CloudFormation stack resources


New ECS Experience
Tell us what you think
Amazon ECS
Clusters
Task Definitions
Account Settings
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions
Clusters › mleip-web-app-demos
Cluster : mleip-web-app-demos
Update Cluster
Delete Cluster
Get a detailed view of the resources on your cluster.
Cluster ARN arn:aws:ecs:eu-west-1:508972911348:cluster/mleip-web-app-demos
Status ACTIVE
Registered container instances 0
Pending tasks count 0 Fargate, 0 EC2, 0 External
Running tasks count 2 Fargate, 0 EC2, 0 External
Active service count 1 Fargate, 0 EC2, 0 External
Draining service count 0 Fargate, 0 EC2, 0 External


New ECS Experience
Tell us what you think
Amazon ECS
Clusters
Task Definitions
Account Settings
Amazon EKS
Clusters
Amazon ECR
Repositories
Task Definitions
Task definitions specify the container information for your application, such as h
resources they will use, how they are linked together, and which host ports they
Create new Task Definition
Create new revision
Actions
Last updated on Septe
Status: ACTIVE INACTIVE
Filter in this page

Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

FARGATE

Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances to manage

EC2

Price based on resource usage

Multiple network modes available

Self-managed infrastructure using Amazon EC2 instances

EXTERNAL

Price based on instance-hours and additional charges for other AWS services used

Self-managed on-premise infrastructure with ECS Anywhere

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. Learn more

Task Definition Name* microservice-forecast-task

Requires Compatibilities* FARGATE

Task Role ecsTaskExecutionRole

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the IAM Console

Network Mode awsvpc

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

Task size

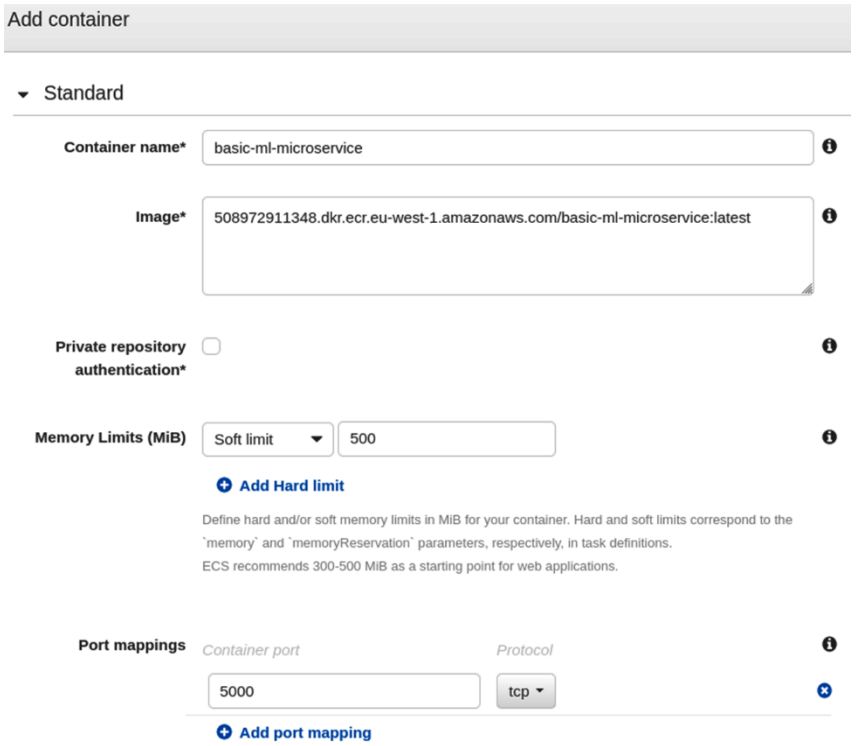
The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 or External launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (GB) 0.5GB

The valid memory range for 0.25 vCPU is: 0.5GB - 2GB.

Task CPU (vCPU) 0.25 vCPU

The valid CPU for 0.5 GB memory is: 0.25 vCPU



Task Definitions

Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they
more

Create new Task Definition | Create new revision | Actions

Status: ACTIVE INACTIVE

Filter in this page

Task Definition	Latest revision status
microservice-forecast-task	ACTIVE

Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

Launch type
- (•) FARGATE
- ( ) EC2
- ( ) EXTERNAL

Switch to capacity provider strategy

Task Definition
Family: microservice-forecast-task
Revision: 1 (latest)

Platform version: LATEST

Cluster: mleip-web-app-demos

Service name: micro-forecast-service

Service type*: REPLICA

Number of tasks: 2

Minimum healthy percent: 50

Maximum percent: 200

Deployment circuit breaker: Disabled

Deployments

Choose a deployment option for the service.

Deployment type*
- (•) Rolling update
- ( ) Blue/green deployment (powered by AWS CodeDeploy)

This sets AWS CodeDeploy as the deployment controller for the service. A CodeDeploy application and deployment group are created automatically with default settings for the service. To change to the rolling update deployment type after the service has been created, you must re-create the service and select the "rolling update" deployment type.

Service IAM role Task definitions that use the awsvpc network mode use the AWSServiceRoleForECS service-linked role, which is created for you automatically. Learn more.

Load balancer name mleip-app-lb

Container to load balance

basic-ml-microservice : 5000

Remove

Production listener port* 80:HTTP

Production listener protocol* HTTP

Target group name create new micro-forecast-service

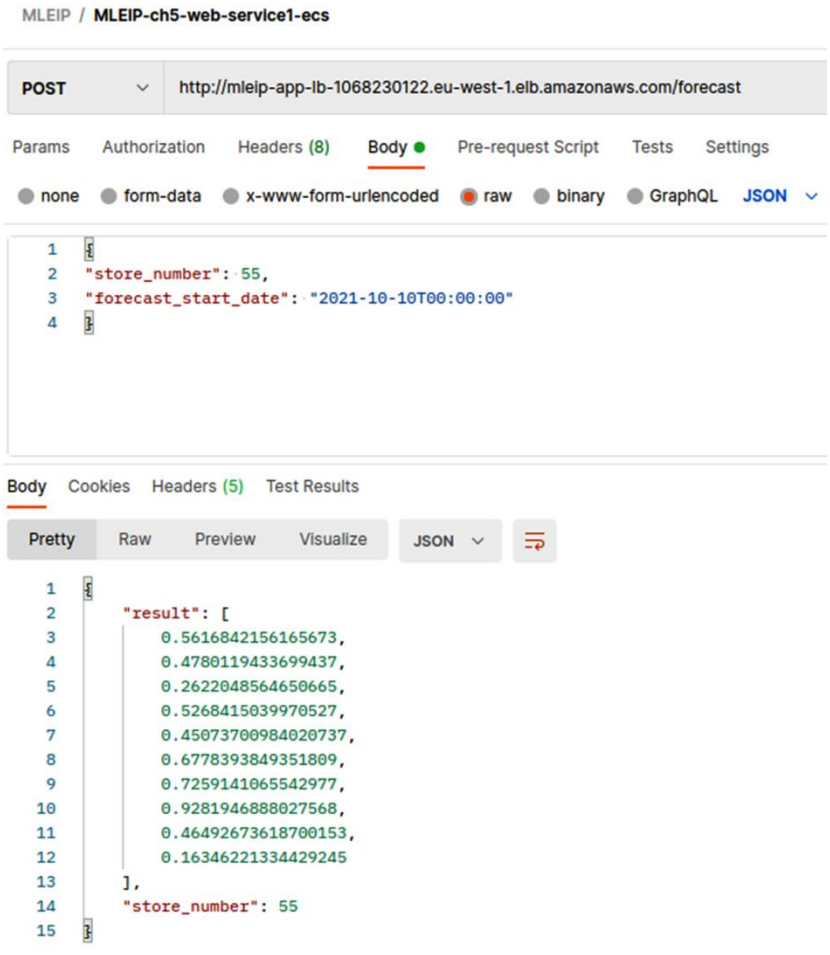
Target group protocol HTTP

Target type ip

Path pattern /micro-forecast-service/forecast **Evaluation order** 1

Path pattern: The first path pattern for a listener is the default path (/), which accepts all traffic that does not match another rule. You can later add additional patterns and priority values to this listener for other services.

Evaluation order: Rules are evaluated in priority order, from the lowest value to the highest value. Once a path pattern rule is matched, all other rules are ignored. You can route traffic from this listener to multiple services by creating a path for each service.


MLEIP / MLEIP-ch5-web-service1-ecs
POST
http://mleip-app-lb-1068230122.eu-west-1.elb.amazonaws.com/forecast
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"store_number": 55,
"forecast_start_date": "2021-10-10T00:00:00"
}
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"result": [
0.5616842156165673,
0.4780119433699437,
0.2622048564650665,
0.5268415039970527,
0.45073700984020737,
0.6778393849351809,
0.7259141065542977,
0.9281946888027568,
0.46492673618700153,
0.16346221334429245
],
"store_number": 55
}

Select load balancer type

Elastic Load Balancing supports four types of load balancers: Appli

Learn more about which load balancer is right for you

Application Load Balancer

Create

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Learn more >

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facin

Name mleip-app-lb

Scheme (•) internet-facing ( ) internal

IP address type ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Add listener

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only on increase the availability of your load balancer.

VPC vpc-422d8f3b (172.31.0.0/16) (default)

Availability Zones [x] eu-west-1a subnet-80223de6

IPv4 address Assigned by AWS

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group (•) Create a **new** security group ( ) Select an **existing** security group

Security group name mleip-app-lb-sg-wizard

Description mleip-app-lb-sg-wizard-1 created on 2021-08-22T20:03:31.275+01:00

Type	Protocol	Port Range	Source
Custom TCP	TCP	80	Custom 0.0.0.0/0, ::/0

Add Rule

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify here. It also performs health checks on this load balancer. You can edit or add listeners after the load balancer is created.

Target group

Target group (i): New target group

Name (i): mleip-app-lb-tg

Target type:
- (•) Instance
- ( ) IP
- ( ) Lambda function

Protocol (i): HTTP

Port (i): 5000

Protocol version (i):
- (•) HTTP1
  Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
- ( ) HTTP2
  Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
- ( ) gRPC
  Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

Protocol (i): HTTP

Path (i): /

Step 6: Review

Please review the load balancer details before continuing

Load balancer

Name	mleip-app-lb
Scheme	internet-facing
Listeners	Port:80 - Protocol:HTTP
IP address type	ipv4
VPC	vpc-422d8f3b
Subnets	subnet-80223de6, subnet-23806e68, subnet-f70159ad
Tags	

Security groups

Security groups	mleip-app-lb-sg-wizard

Routing

Target group	New target group
Target group name	mleip-app-lb-tg
Port	5000
Target type	instance
Protocol	HTTP
Protocol version	HTTP1
Health check protocol	HTTP
Path	/
Health check port	traffic port
Healthy threshold	5
Unhealthy threshold	2
Timeout	5
Interval	30
Success codes	200

Load Balancer Creation Status

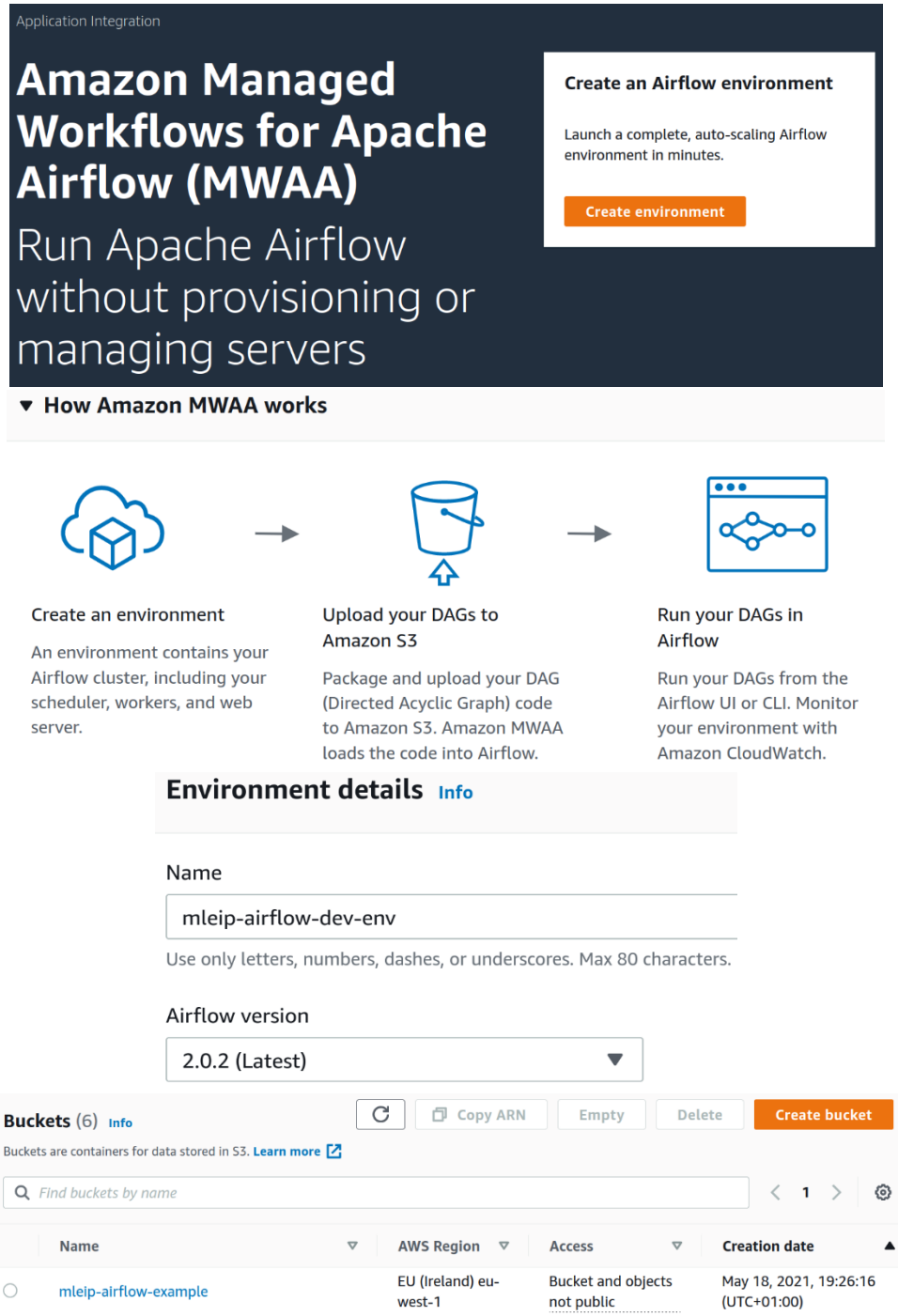
Successfully created load balancer

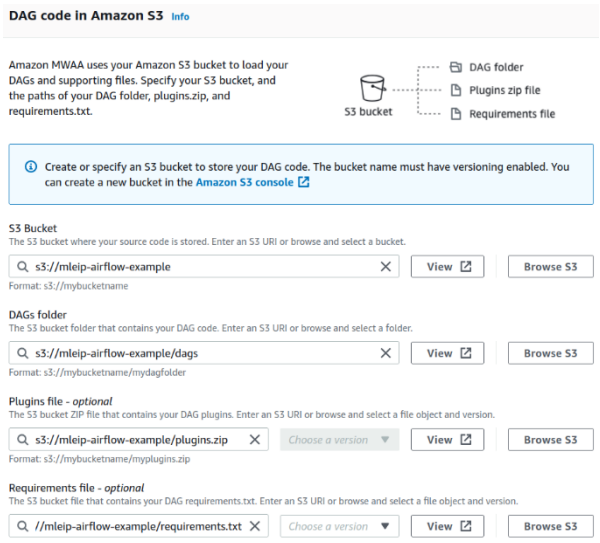
Load balancer mleip-app-lb was successfully created.

Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks.

Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within mleip-app-lb
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. AWS Global Accelerator console


Application Integration
Amazon Managed Workflows for Apache Airflow (MWAA)
Run Apache Airflow without provisioning or managing servers
Create an Airflow environment
Launch a complete, auto-scaling Airflow environment in minutes.
Create environment
How Amazon MWAA works
Create an environment
An environment contains your Airflow cluster, including your scheduler, workers, and web server.
Upload your DAGs to Amazon S3
Package and upload your DAG (Directed Acyclic Graph) code to Amazon S3. Amazon MWAA loads the code into Airflow.
Run your DAGs in Airflow
Run your DAGs from the Airflow UI or CLI. Monitor your environment with Amazon CloudWatch.
Environment details Info
Name
mleip-airflow-dev-env
Use only letters, numbers, dashes, or underscores. Max 80 characters.
Airflow version
2.0.2 (Latest)
Buckets (6) Info
Copy ARN
Empty
Delete
Create bucket
Buckets are containers for data stored in S3. Learn more
Find buckets by name
1
Name
AWS Region
Access
Creation date
mleip-airflow-example
EU (Ireland) eu-west-1
Bucket and objects not public
May 18, 2021, 19:26:16 (UTC+01:00)



Quick create stack

Template

Template URL

https://mwaa-downloads.s3-us-west-2.amazonaws.com/mwaa-vpc-cfn-template.yaml

Stack description

This template deploys a VPC, with a pair of public and private subnets spread across two Availability Zones. It deploys an internet gateway, with a default route on the public subnets. It deploys a pair of NAT gateways (one in each AZ), and default routes for them in the private subnets.

Stack name

Stack name

MWAA-VPC

Web server access

- (•) Private network (Recommended)
  Additional setup required. Your Airflow UI can only be accessed by secure login behind your VPC. Choose this option if your Airflow UI is only accessed within a corporate network. IAM must be used to handle user authentication.
- ( ) Public network (No additional setup)
  Your Airflow UI can be accessed by secure login over the Internet. Choose this option if your Airflow UI is accessed outside of a corporate network. IAM must be used to handle user authentication.

ⓘ For private network access, the Airflow web server is reached via a VPC endpoint inside your VPC. Connecting to the endpoint requires additional setup. Learn more about VPC endpoints

Security group(s)
A VPC security group is required to allow traffic between your environment and your web server.

- [x] Create new security group
  Allow MWAA to create a VPC security group with inbound and outbound rules based on your selection for web server access.

Existing security group(s)
You can choose 1 or more existing security groups to configure the inbound and outbound rules for your environment.

Choose security group ▼ ↻

Max 5 security groups

	DAG capacity*	Scheduler CPU	Worker CPU	Web server CPU
(•) mw1.small	Up to 50	1 vCPU	1 vCPU	0.5 vCPU
( ) mw1.medium	Up to 250	2 vCPU	2 vCPU	1 vCPU
( ) mw1.large	Up to 1000	4 vCPU	4 vCPU	2 vCPU

*under typical usage

Maximum worker count
The maximum number of workers your environment is permitted to scale up to.

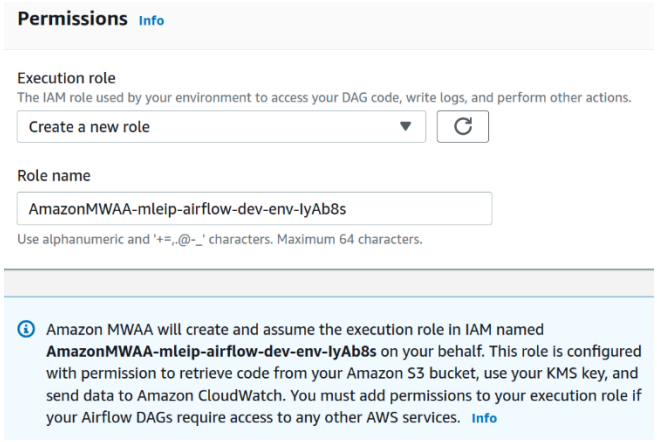
10

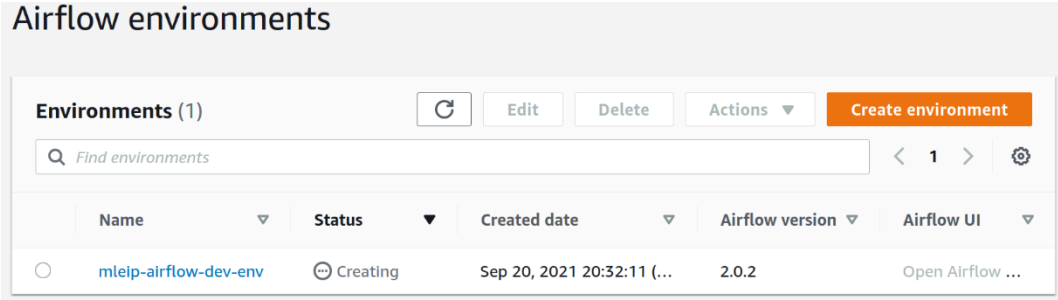
Must be between 1 and 25

Minimum worker count
The minimum number of workers always present in your environment.

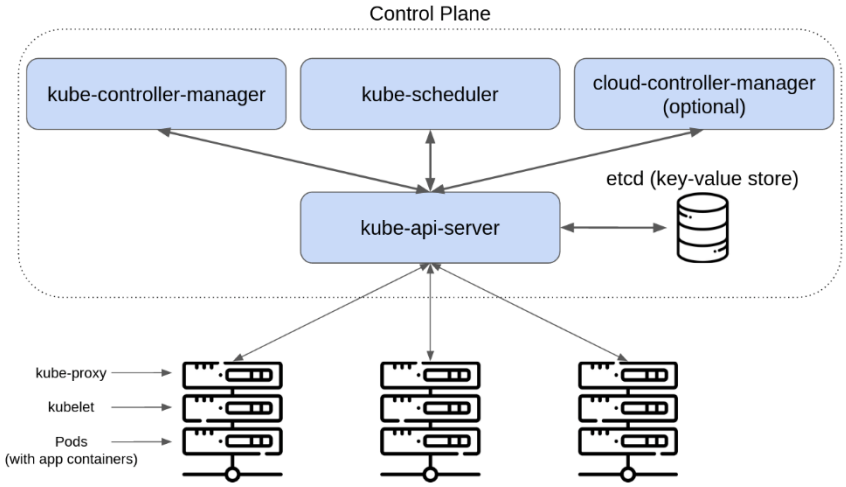
1

Must be less than or equal to maximum workers. Minimum 1 worker


Permissions Info
Execution role
The IAM role used by your environment to access your DAG code, write logs, and perform other actions.
Create a new role
Role name
AmazonMWAA-mleip-airflow-dev-env-IyAb8s
Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.
Amazon MWAA will create and assume the execution role in IAM named AmazonMWAA-mleip-airflow-dev-env-IyAb8s on your behalf. This role is configured with permission to retrieve code from your Amazon S3 bucket, use your KMS key, and send data to Amazon CloudWatch. You must add permissions to your execution role if your Airflow DAGs require access to any other AWS services. Info


Airflow environments
Environments (1)
Edit
Delete
Actions
Create environment
Find environments
1
Name
Status
Created date
Airflow version
Airflow UI
mleip-airflow-dev-env
Creating
Sep 20, 2021 20:32:11 (...
2.0.2
Open Airflow ...

Chapter 6: Scaling Up




```
+---+-------------+-------+---------+-------+-------+-------+--------+---+-----+--------+--------+-----+--------+--------+---+
|age|          job|marital|education|default|balance|housing|    loan| contact|day|month|duration|campaign|pdays|previous|poutcome|  y|
+---+-------------+-------+---------+-------+-------+-------+--------+--------+---+-----+--------+--------+-----+--------+--------+---+
| 30|   unemployed|married|  primary|     no|   1787|     no|      no|cellular| 19|  oct|      79|       1|   -1|       0| unknown| no|
| 33|     services|married|secondary|     no|   4789|    yes|     yes|cellular| 11|  may|     220|       1|  339|       4| failure| no|
| 35|   management| single| tertiary|     no|   1350|    yes|      no|cellular| 16|  apr|     185|       1|  330|       1| failure| no|
| 30|   management|married| tertiary|     no|   1476|    yes|     yes| unknown|  3|  jun|     199|       4|   -1|       0| unknown| no|
| 59|  blue-collar|married|secondary|     no|      0|    yes|      no| unknown|  5|  may|     226|       1|   -1|       0| unknown| no|
| 35|   management| single| tertiary|     no|    747|     no|      no|cellular| 23|  feb|     141|       2|  176|       3| failure| no|
| 36|self-employed|married| tertiary|     no|    307|    yes|      no|cellular| 14|  may|     341|       1|  330|       2|   other| no|
| 39|   technician|married|secondary|     no|    147|    yes|      no|cellular|  6|  may|     151|       2|   -1|       0| unknown| no|
| 41| entrepreneur|married| tertiary|     no|    221|    yes|      no| unknown| 14|  may|      57|       2|   -1|       0| unknown| no|
| 43|     services|married|  primary|     no|    -88|    yes|     yes|cellular| 17|  apr|     313|       1|  147|       2| failure| no|
+---+-------------+-------+---------+-------+-------+-------+--------+--------+---+-----+--------+--------+-----+--------+--------+---+
```

```
|-- age: integer (nullable = true)
|-- job: string (nullable = true)
|-- marital: string (nullable = true)
|-- education: string (nullable = true)
|-- default: string (nullable = true)
|-- balance: integer (nullable = true)
|-- housing: string (nullable = true)
|-- loan: string (nullable = true)
|-- contact: string (nullable = true)
|-- day: integer (nullable = true)
|-- month: string (nullable = true)
|-- duration: integer (nullable = true)
|-- campaign: integer (nullable = true)
|-- pdays: integer (nullable = true)
|-- previous: integer (nullable = true)
|-- poutcome: string (nullable = true)
|-- y: string (nullable = true)
```

pdays	previous	poutcome	y	month_as_int
-1	0	unknown	no	10
339	4	failure	no	5
330	1	failure	no	4
-1	0	unknown	no	6
-1	0	unknown	no	5
176	3	failure	no	2
330	2	other	no	5
-1	0	unknown	no	5
-1	0	unknown	no	5
147	2	failure	no	4
-1	0	unknown	no	5
-1	0	unknown	no	4
-1	0	unknown	no	8
-1	0	unknown	yes	4
241	1	failure	no	1
-1	0	unknown	no	8
-1	0	unknown	no	8
152	2	failure	no	4
-1	0	unknown	no	5
152	1	other	no	7

_1	_2	_3	_4	_5	_6	_7	_8	_9	_10	_11	_12	_13	class
14.23	1.71	2.43	15.6	127.0	2.8	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
13.2	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.4	1050.0	0
13.16	2.36	2.67	18.6	101.0	2.8	3.24	0.3	2.81	5.68	1.03	3.17	1185.0	0
14.37	1.95	2.5	16.8	113.0	3.85	3.49	0.24	2.18	7.8	0.86	3.45	1480.0	0
13.24	2.59	2.87	21.0	118.0	2.8	2.69	0.39	1.82	4.32	1.04	2.93	735.0	2

General Configuration

Cluster name My cluster

Logging

S3 folder s3://aws-logs-508972911348-eu-west-1/elasticmapr

Launch mode Cluster Step execution

Software configuration

Release emr-5.33.0

Applications

- Core Hadoop: Hadoop 2.10.1, Hive 2.3.7, Hue 4.9.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- HBase: HBase 1.4.13, Hadoop 2.10.1, Hive 2.3.7, Hue 4.9.0, Phoenix 4.14.3, and ZooKeeper 3.4.14
- Presto: Presto 0.245.1 with Hadoop 2.10.1 HDFS and Hive 2.3.7 Metastore
- Spark: Spark 2.4.7 on Hadoop 2.10.1 YARN and Zeppelin 0.9.0

Use AWS Glue Data Catalog for table metadata

Hardware configuration

Instance type m1.medium

Number of instances 2 (1 master and 1 core nodes)

Cluster scaling scale cluster nodes based on workload

Security and access

EC2 key pair No key pairs found Learn how to create an EC2 key pair.

Permissions Default Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role EMR_DefaultRole Use EMR_DefaultRole_V2

EC2 instance profile EMR_EC2_DefaultRole

Clone | Terminate | AWS CLI export

Cluster: My cluster Starting

Summary | Application user interfaces | Monitoring | Hardware | Configurations | Events | Steps | Bootstrap actions

Summary

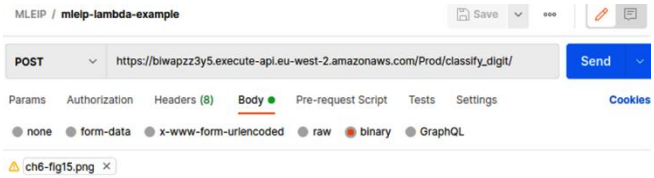
ID: j-17UN3RZ4S7K7T
Creation date: 2021-09-22 19:44 (UTC+1)
Elapsed time: 1 minute
After last step completes: Cluster waits
Termination protection: Off Change
Tags: -- View All / Edit
Master public DNS:
ec2-34-243-3-14.eu-west-1.compute.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

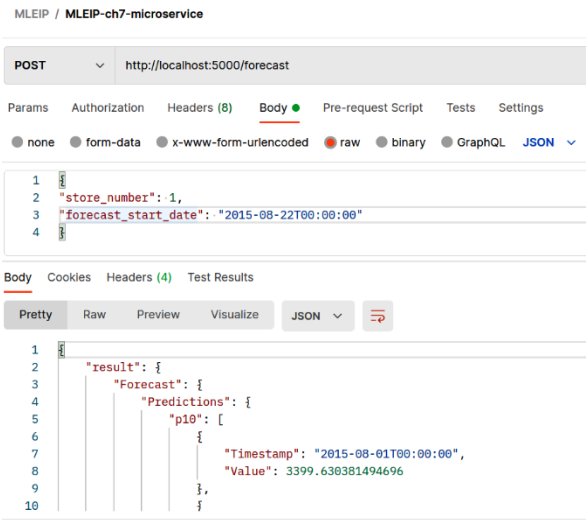
Release label: emr-5.33.0
Hadoop distribution: Amazon
Applications: Spark 2.4.7, Zeppelin 0.9.0
Log URI: s3://aws-logs-508972911348-eu-west-1/elasticmapreduce/
EMRFS consistent view: Disabled

```
Which template source would you like to use?What package type would you like to use?
        1 - AWS Quick Start Templates               1 - Zip (artifact is a zip uploaded to S3)
        2 - Custom Template Location                2 - Image (artifact is an image uploaded to an ECR image repository)
Choice:                                     Package type:

                Which base image would you like to use?
                        1 - amazon/nodejs14.x-base
                        2 - amazon/nodejs12.x-base
                        3 - amazon/nodejs10.x-base
                        4 - amazon/python3.8-base
                        5 - amazon/python3.7-base
                        6 - amazon/python3.6-base
                        7 - amazon/python2.7-base
                        8 - amazon/ruby2.7-base
                        9 - amazon/ruby2.5-base
                        10 - amazon/go1.x-base
                        11 - amazon/java11-base
                        12 - amazon/java8.al2-base
                        13 - amazon/java8-base
                        14 - amazon/dotnet5.0-base
                        15 - amazon/dotnetcore3.1-base
                        16 - amazon/dotnetcore2.1-base
                Base image: 4                           Project name [sam-app]: mleip-lambda-example
                             AWS quick start application templates:
                                    1 - Hello World Lambda Image Example
                                    2 - PyTorch Machine Learning Inference API
                                    3 - Scikit-learn Machine Learning Inference API
                                    4 - Tensorflow Machine Learning Inference API
                                    5 - XGBoost Machine Learning Inference API
                             Template selection: 3
                  -----------------------
                  Generating application:
                  -----------------------
                  Name: mleip-lambda-example
                  Base Image: amazon/python3.8-base
                  Dependency Manager: pip
                  Output Directory: .

                  Next steps can be found in the README file at ./mleip-lambda-example/README.md
```

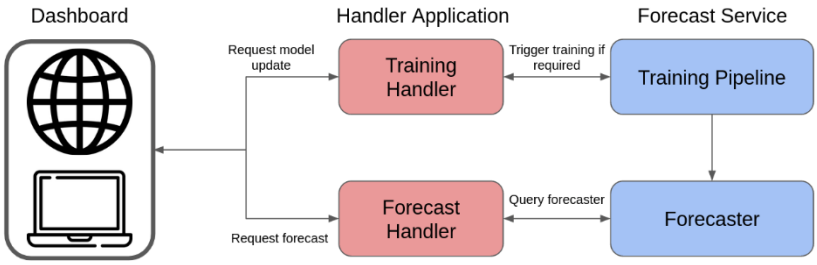
```
CloudFormation outputs from deployed stack
---------------------------------------------------------------------------------------------------------------------
Outputs
---------------------------------------------------------------------------------------------------------------------
Key                 InferenceApi
Description         API Gateway endpoint URL for Prod stage for Inference function
Value               https://biwapzz3y5.execute-api.eu-west-2.amazonaws.com/Prod/classify_digit/

Key                 InferenceFunctionIamRole
Description         Implicit IAM Role created for Inference function
Value               arn:aws:lambda:eu-west-2:508972911348:function:sam-app-InferenceFunction-WKDKOefcLIWn

Key                 InferenceFunction
Description         Inference Lambda Function ARN
Value               arn:aws:lambda:eu-west-2:508972911348:function:sam-app-InferenceFunction-WKDKOefcLIWn
---------------------------------------------------------------------------------------------------------------------

Successfully created/updated stack - sam-app in eu-west-2
```



Chapter 7: Building an Example ML Microservice

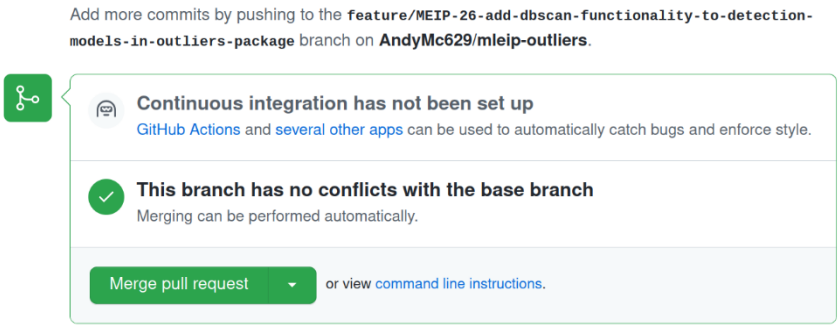


User story	Details	Technical requirements
1	As a local logistics planner, I want to log in to a dashboard in the morning and be able to see forecasts of item demand at store level for the next few days.	• Target variable = item demand • Forecast horizon = 1–7 days • Interface with dashboard required
2	As a local logistics planner, I want to be able to request an update of my forecast if I see it is out of date. I want the new forecast to be retrieved in a reasonable time.	• Lightweight retraining • Model per store?
3	As a local logistics planner, I want to be able to filter for forecasts based on specific stores.	• Model per store

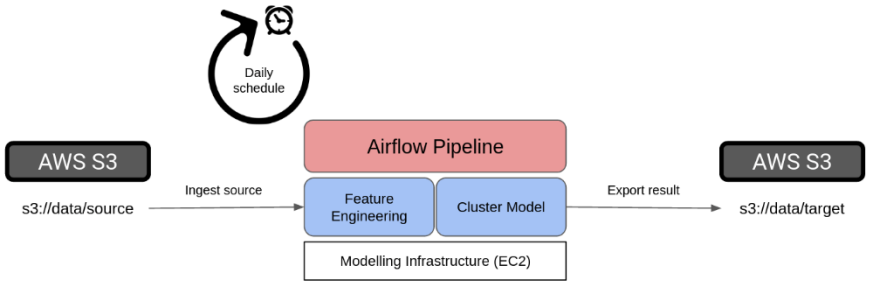


Tool/ framework	Pros	Cons
Sklearn	• Already understood by almost all data scientists • Very easy-to-use syntax • Lots of great community support • Good feature engineering and pipelining support	• No native time series modeling capabilities • Will require some more feature engineering to apply to models to time series data • More work and understanding required by engineer/scientist
Prophet	• Purely focused on forecasting • Has inbuilt hyperparameter optimization capabilities • Provides a lot of easy-to-use functionality out of the box • Often gives accurate results on a wide variety of problems • Provides confidence intervals out of the box	• Not as commonly used as sklearn (but still relatively popular) • Underlying methods are quite sophisticated – may lead to *black box* usage • Not inherently scalable
Spark MLlib	• Natively scalable to large volumes • Good feature engineering and pipelining support	• No native time series modeling capabilities • Algorithm options are relatively limited

Chapter 8: Building an Extract Transform Machine Learning Use Case



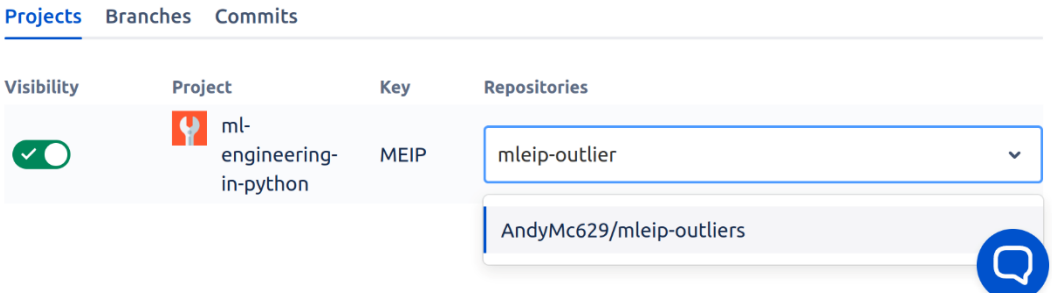
User Story	Details	Technical Requirements
1	As an operations analyst, I want to be given clear labels of rides that have anomalously long ride times or distances.	• Algorithm type = anomaly detection/ clustering/outlier detection • Features = ride time and distance
2	As an internal application developer, I want to have a clear access point for data with anomalous labels. This data should be stored in the cloud.	• System output destination = S3 on AWS
3	As an operations analyst, I would like to see labels for the previous day's rides every morning.	• Batch frequency = daily

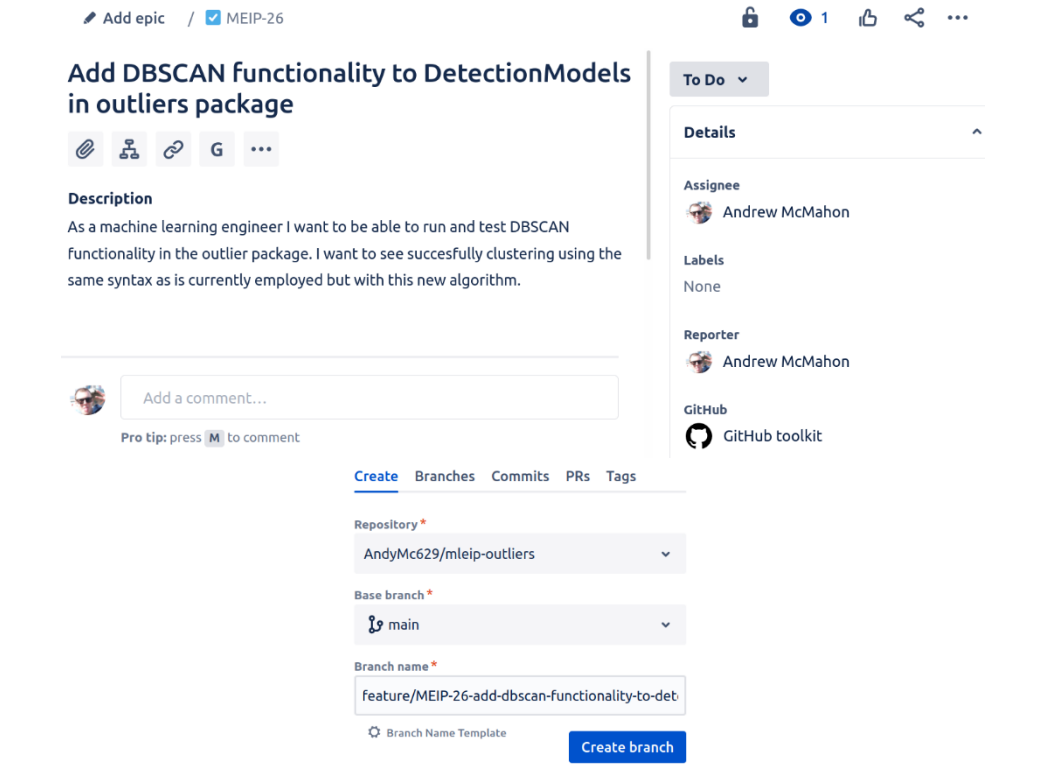


Solution Aspect	Potential Tools	Pros	Cons
Interfaces	AWS **Command-Line Interface** (**CLI**) and `boto3`	• Simple to use • Connects to a wide variety of other AWS tools and services	• Not cloud-agnostic • Not applicable outside of AWS (on-premises systems, for example)

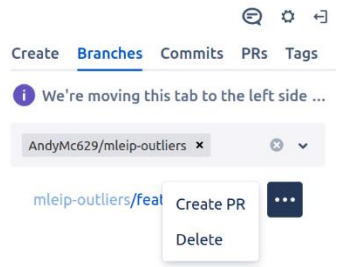
Potential Tools	Pros	Cons
Spark MLlib	Can scale to very large datasets	• Requires cluster management • Overkill for smaller datasets/ processing requirements • Limited algorithm set
Spark with pandas **User-Defined Function** (**UDF**)	• Can scale to very large datasets • Can use any Python-based algorithm	Might not make sense for some problems where parallelization is not easily applicable
Scikit-learn	• Well known by many data scientists • Can run on many different types of infrastructure	Not very scalable

Potential Tools	Pros	Cons
Apache Airflow	• Good scheduling management • Ability to build relatively complex pipelines • Good documentation • Cloud-hosted services available, such as AWS **Managed Workflows for Apache Airflow** (**MWAA**)	• Learning curve for usage • Takes time to test pipelines and scheduling • Cloud services (MWAA) can be expensive




Add epic / MEIP-26
1
Add DBSCAN functionality to DetectionModels in outliers package
Description
As a machine learning engineer I want to be able to run and test DBSCAN functionality in the outlier package. I want to see succesfully clustering using the same syntax as is currently employed but with this new algorithm.
Add a comment...
Pro tip: press M to comment
To Do
Details
Assignee
Andrew McMahon
Labels
None
Reporter
Andrew McMahon
GitHub
GitHub toolkit
Create
Branches
Commits
PRs
Tags
Repository *
AndyMc629/mleip-outliers
Base branch *
main
Branch name *
feature/MEIP-26-add-dbscan-functionality-to-det
Branch Name Template
Create branch


Overview
Yours
Active
Stale
All branches
Default branch
main
Updated last month by AndyMc629
Default
Your branches
feature/MEIP-26-add-dbscan-functionality-to-detection-models-in-outliers-package
Updated last month by AndyMc629
#1
Open
Active branches
feature/MEIP-26-add-dbscan-functionality-to-detection-models-in-outliers-package
Updated last month by AndyMc629
#1
Open



Feature/meip 26 add dbscan functionality to detection models in outliers package #1

Open

AndyMc629 wants to merge 3 commits into `main` from `feature/MEIP-26-add-dbscan-functionality-to-detection-mod`

Conversation 0 · Commits 3 · Checks 0 · Files changed 5

AndyMc629 commented on 12 Aug

Main changes:

- Logging
- Exception handling
- Inclusion of DBSCAN functionality
- Taxi data simulation