Serverless in AI Insights

Oct. 2022

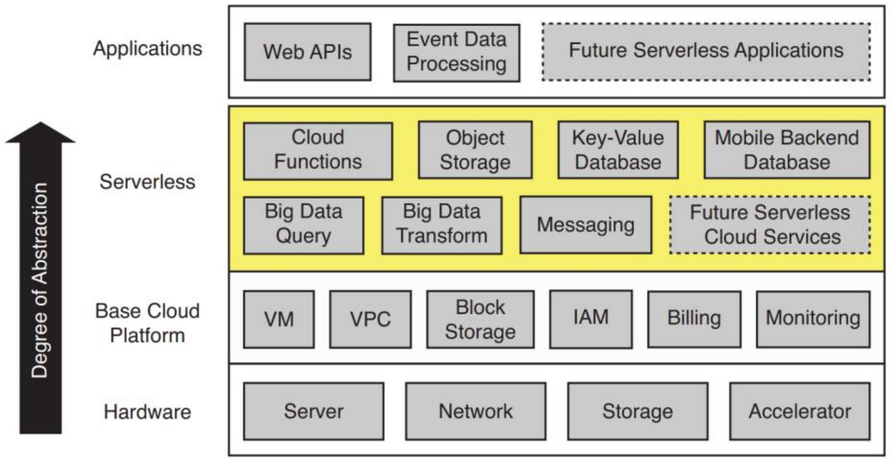
Outline

- Definition: serverless, serverless service, FaaS
- Serverless ML architecture
- Google Colab Notebook and AWS SageMaker severless
- AWS serverless service

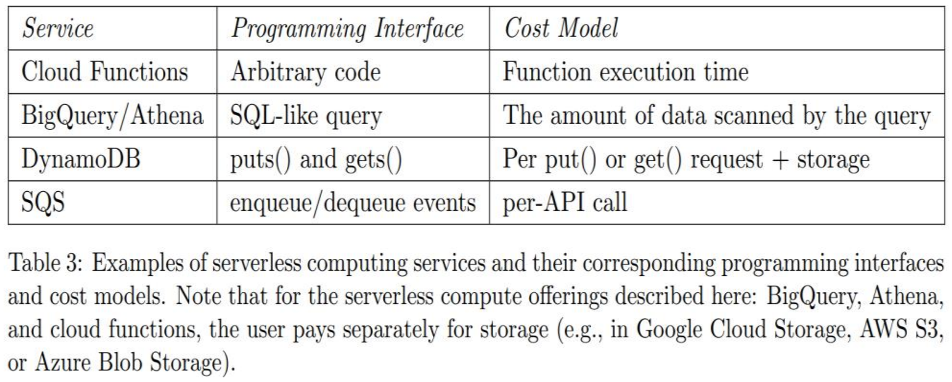
Definition

- Serverless: no server management, further simplified cloud computing
- Serverless service: components in the serverless computing paradigm
- FaaS (function-as-a-service): no services management, building block for serverless applications

Berkeley View on Serverless Computing



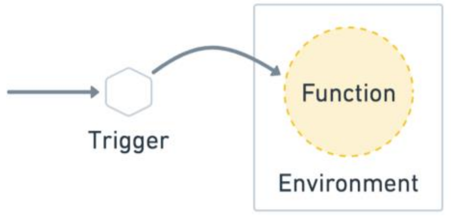
Serverless Service

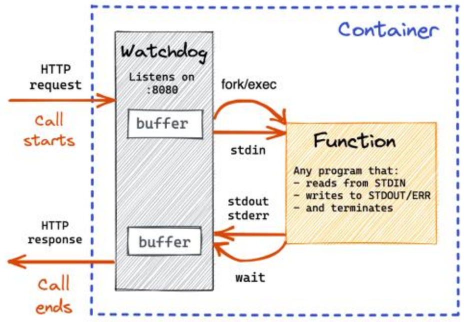


Service	*Programming Interface*	*Cost Model*
Cloud Functions	Arbitrary code	Function execution time
BigQuery/Athena	SQL-like query	The amount of data scanned by the query
DynamoDB	puts() and gets()	Per put() or get() request + storage
SQS	enqueue/dequeue events	per-API call

Table 3: Examples of serverless computing services and their corresponding programming interfaces and cost models. Note that for the serverless compute offerings described here: BigQuery, Athena, and cloud functions, the user pays separately for storage (e.g., in Google Cloud Storage, AWS S3, or Azure Blob Storage).

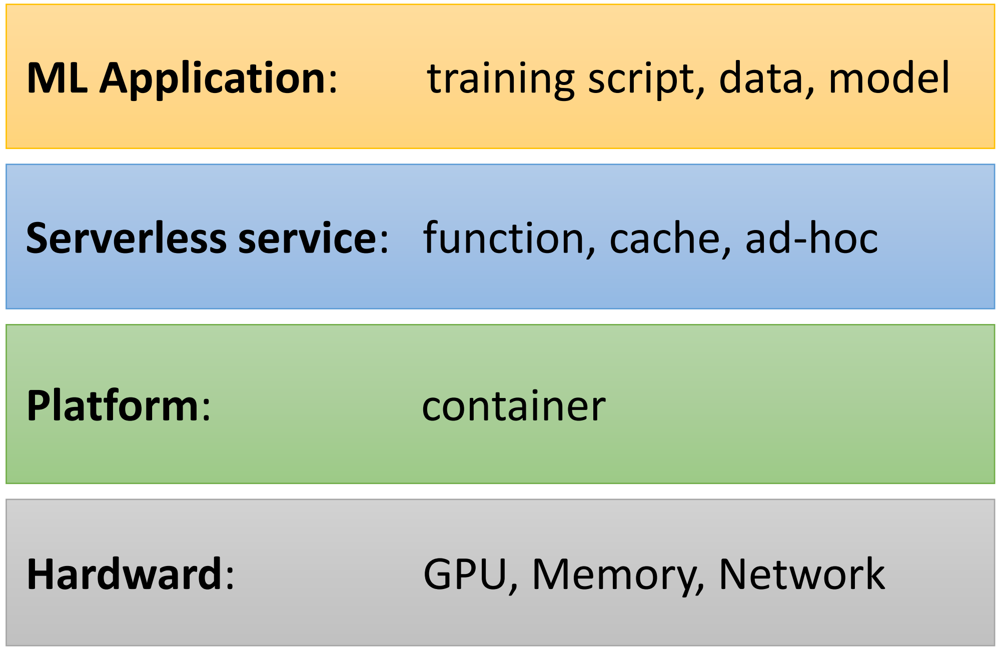
FaaS



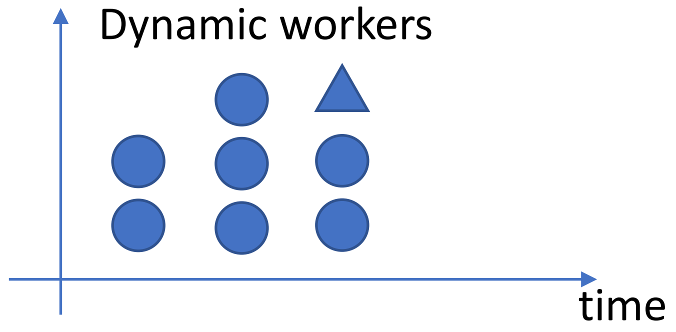


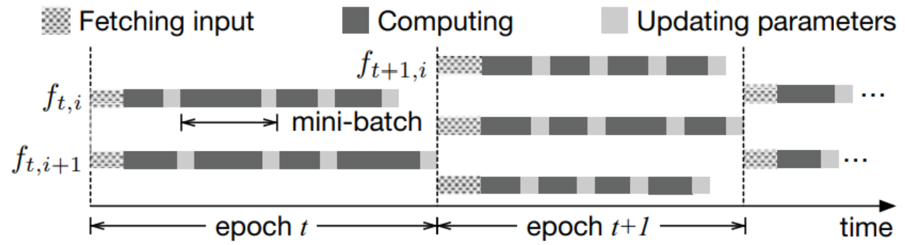
Serverless ML Architecture

- Training
  - Multi-dimensional split on training tasks (e.g, epoch, data, preprocessing)
  - Dynamically allocate workers(services/resources) and coordinate states between workers
  - High performance function and cache service, and ad-hoc data processing service
- Serving
  - Smart warm-up and batch, meet SLO with minimal cost


ML Application: training script, data, model
Serverless service: function, cache, ad-hoc
Platform: container
Hardward: GPU, Memory, Network

→ Elastic and Intelligent

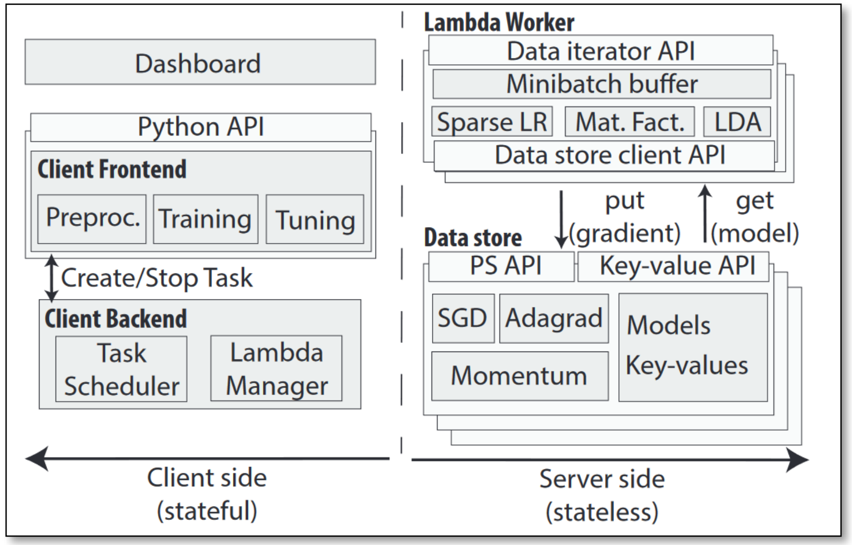




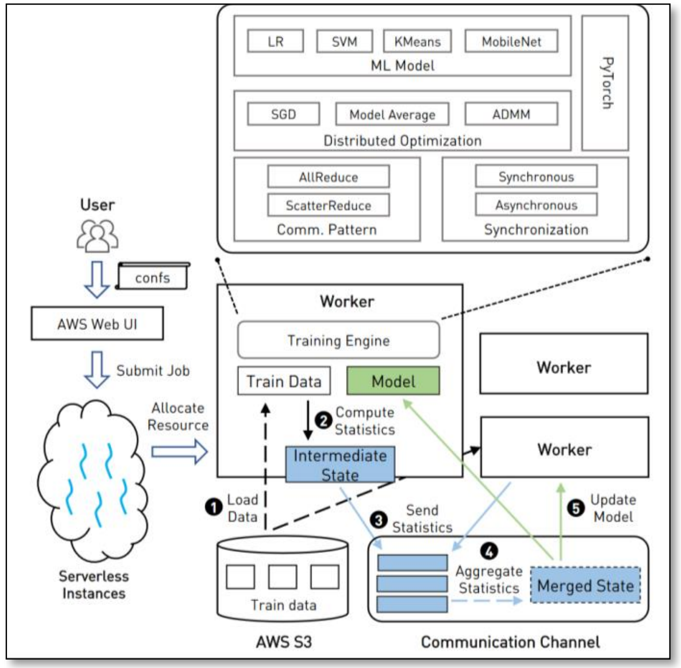
Reference: Serverless ML Framework

- CIRRUS, 2019: https://andrewmzhang.com/assets/pdf/p13-Carreira.pdf
- LambdaML, 2021, https://arxiv.org/abs/2105.07806
- SMLT, 2022, https://arxiv.org/abs/2205.01853

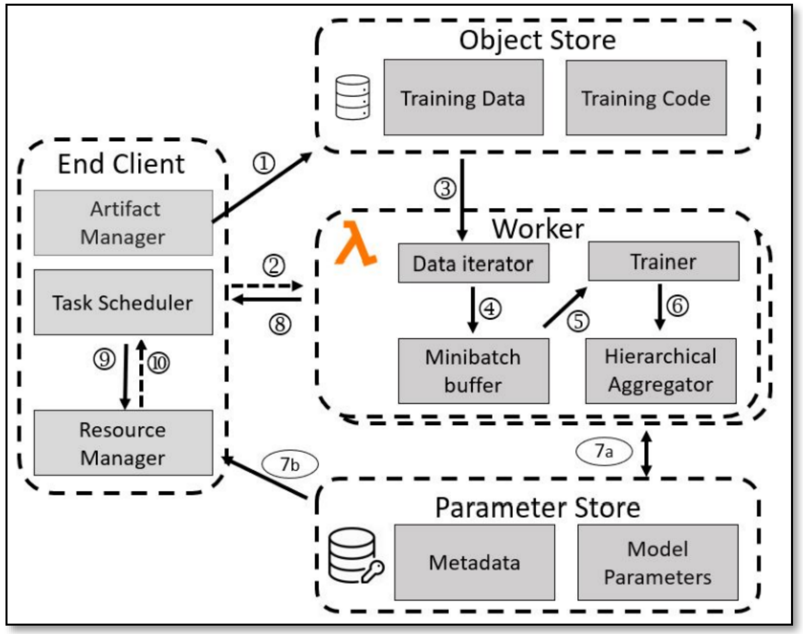
CIRRUS



LambdaML



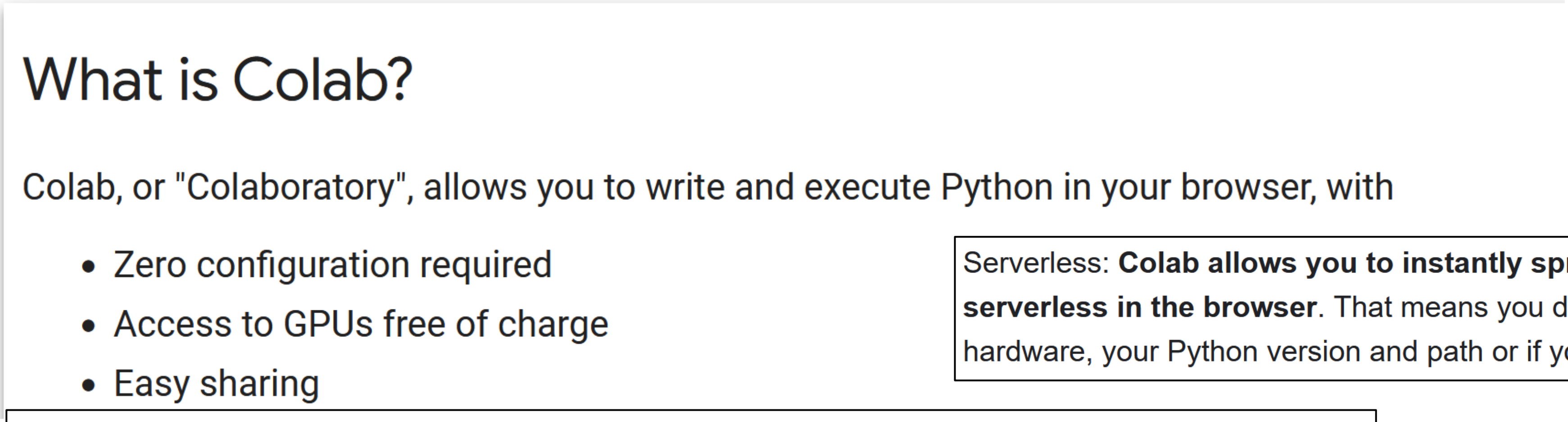
SMLT



Design Notes

- Fast storage service is critical to exchange gradients and parameters
- Data parallelism is the first step, add advanced services to parse and coordinate model for model parallelism
- Leverage vGPU features, dynamically assign GPU memory in a worker as training goes
- Dynamically schedule the number of workers without training interruption
- Leverage open source fass project (e.g., fission), build ML oriented container (gpu, rdma), avoid general FaaS limits.

Serverless notebook training, Google Colab

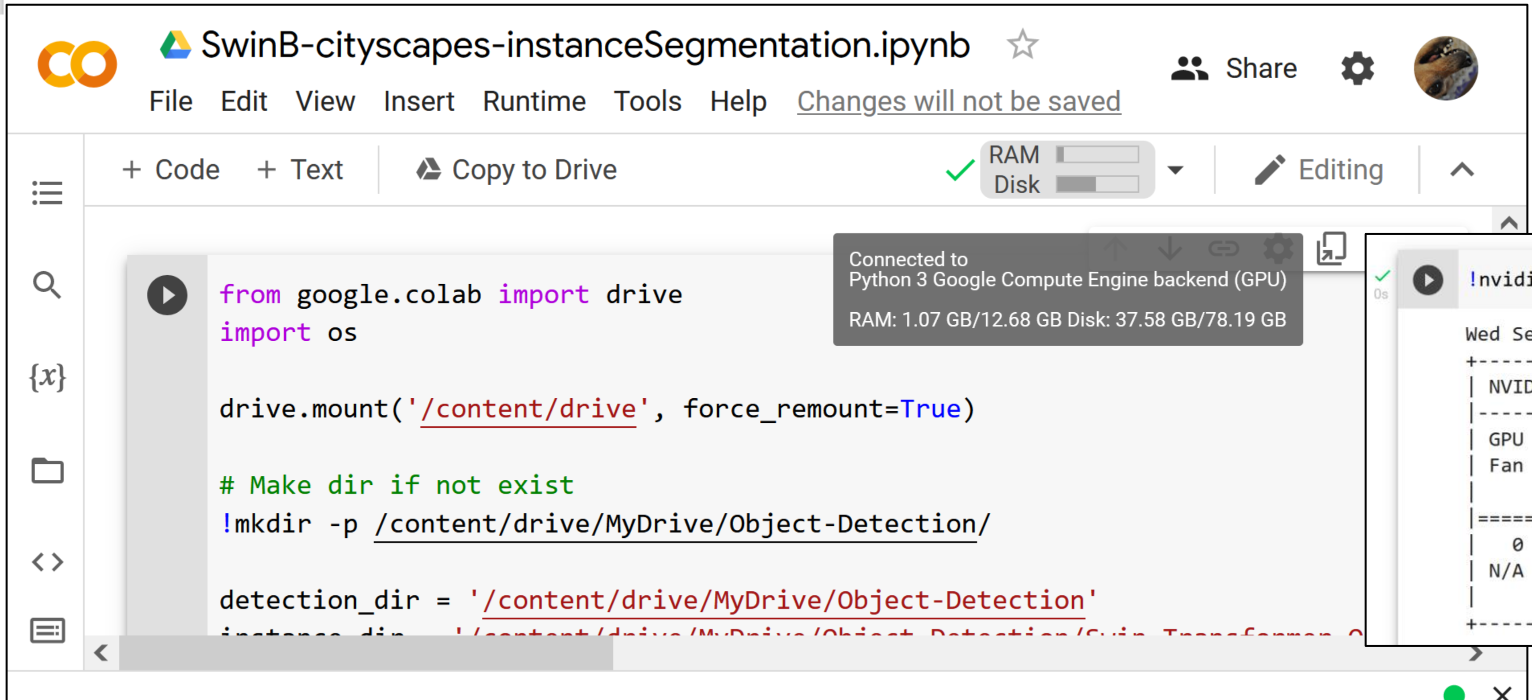


What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

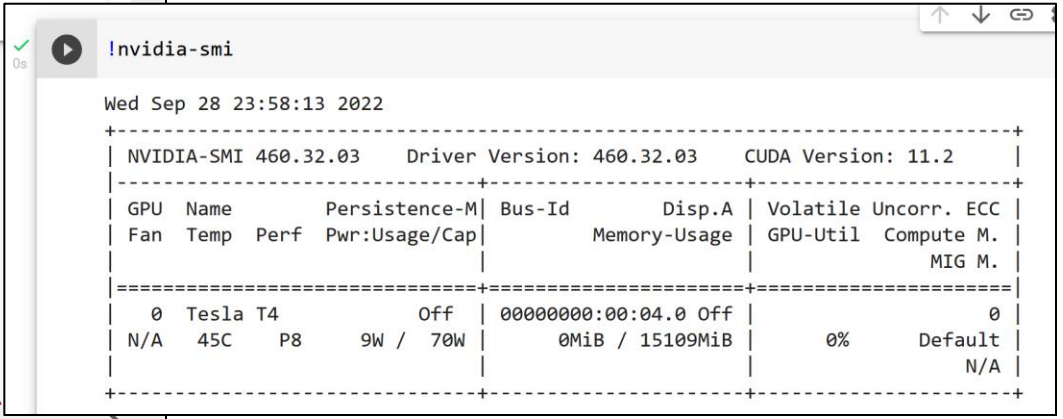
- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Serverless: **Colab allows you to instantly spring up a Jupyter Notebook completely serverless in the browser**. That means you don't have to worry about provisioning hardware, your Python version and path or if you're on Windows, MacOS or even a phone!


SwinB-cityscapes-instanceSegmentation.ipynb
Share
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
RAM Disk
Editing

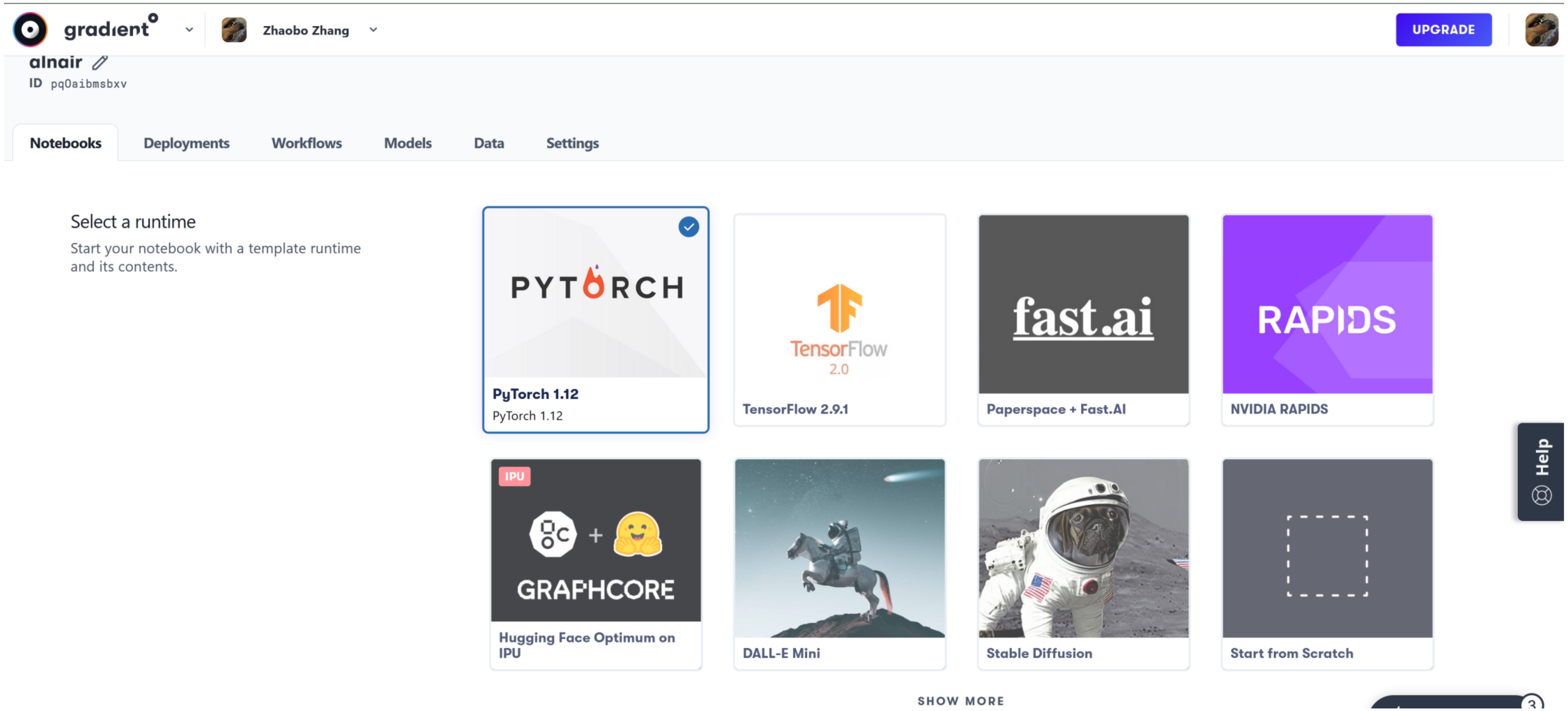
Connected to
Python 3 Google Compute Engine backend (GPU)
RAM: 1.07 GB/12.68 GB Disk: 37.58 GB/78.19 GB

```
from google.colab import drive
import os

drive.mount('/content/drive', force_remount=True)

# Make dir if not exist
!mkdir -p /content/drive/MyDrive/Object-Detection/

detection_dir = '/content/drive/MyDrive/Object-Detection'
instance_dir = '/content/drive/MyDrive/Object-Detection/Swin-Transformer-O
```



```
!nvidia-smi

Wed Sep 28 23:58:13 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   45C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
```

Serverless notebook training, gradient notebook



gradient Zhaobo Zhang UPGRADE

alnair

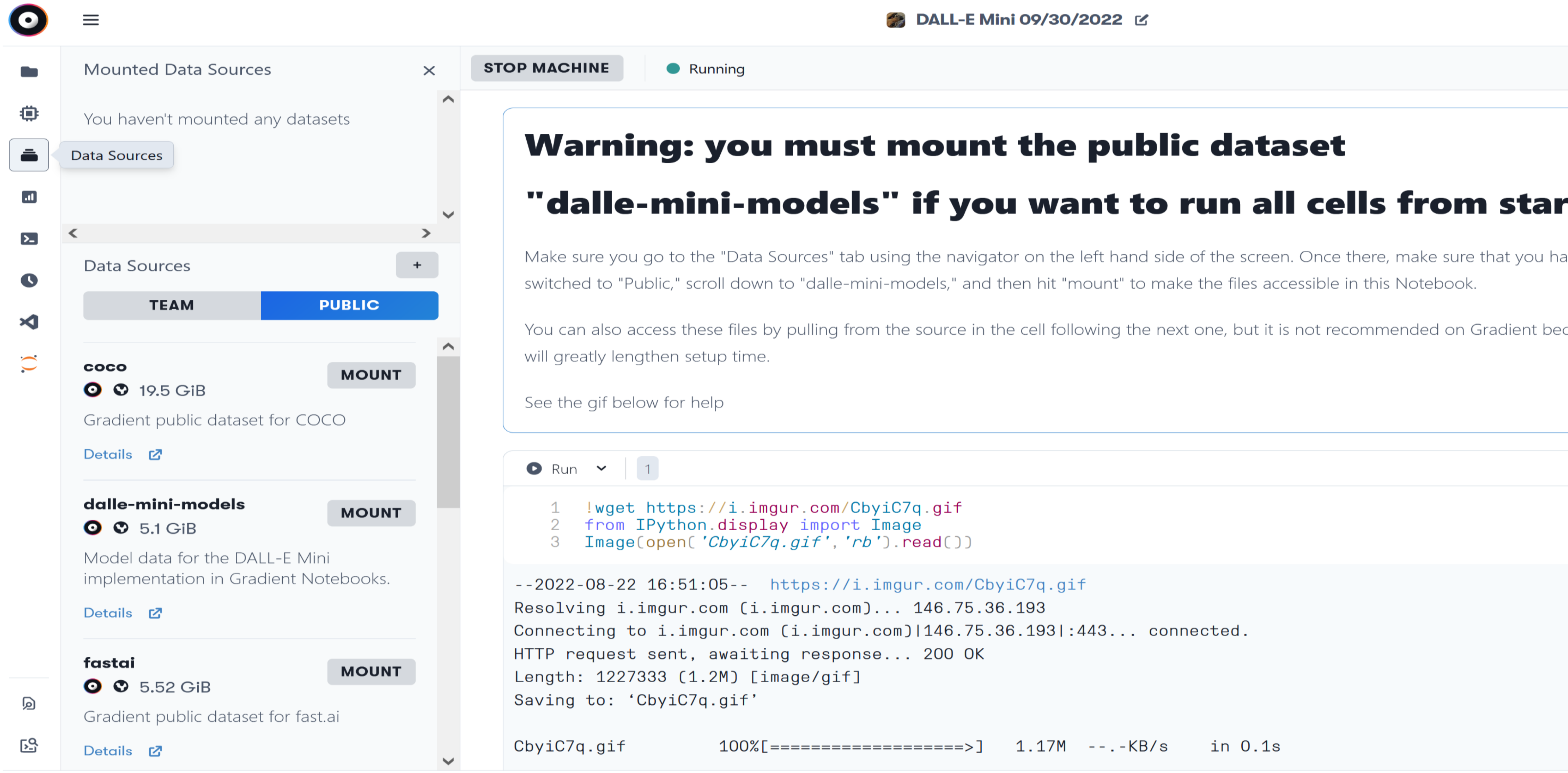
ID pq0aibmsbxv

Notebooks Deployments Workflows Models Data Settings

Select a runtime

Start your notebook with a template runtime and its contents.

Disaggregated data, Resource Utils, Workflow



DALL-E Mini 09/30/2022

Mounted Data Sources

You haven't mounted any datasets

Data Sources

STOP MACHINE   Running

Data Sources

TEAM   PUBLIC

coco   MOUNT
19.5 GiB
Gradient public dataset for COCO
Details

dalle-mini-models   MOUNT
5.1 GiB
Model data for the DALL-E Mini implementation in Gradient Notebooks.
Details

fastai   MOUNT
5.52 GiB
Gradient public dataset for fast.ai
Details

Warning: you must mount the public dataset "dalle-mini-models" if you want to run all cells from star

Make sure you go to the "Data Sources" tab using the navigator on the left hand side of the screen. Once there, make sure that you ha switched to "Public," scroll down to "dalle-mini-models," and then hit "mount" to make the files accessible in this Notebook.

You can also access these files by pulling from the source in the cell following the next one, but it is not recommended on Gradient bec will greatly lengthen setup time.

See the gif below for help

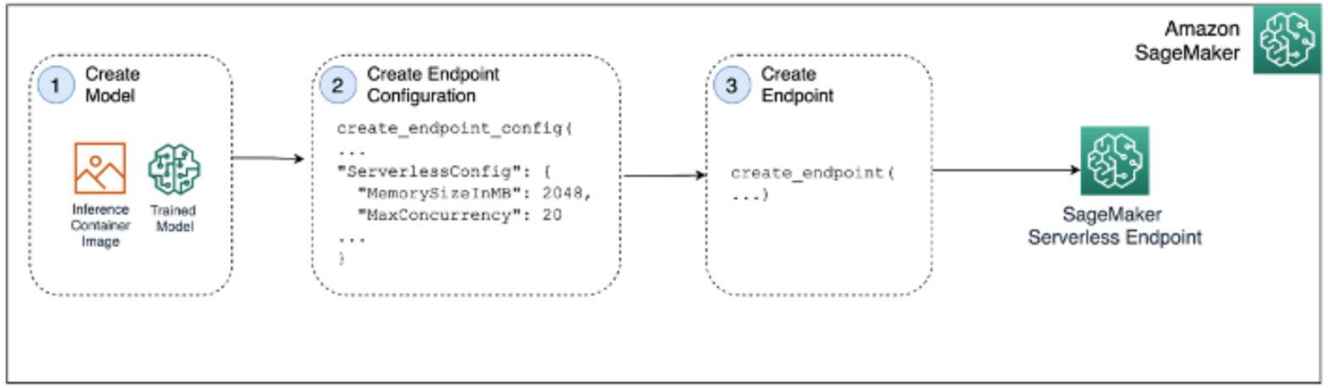
Run   1

```
!wget https://i.imgur.com/CbyiC7q.gif
from IPython.display import Image
Image(open('CbyiC7q.gif','rb').read())
```

```
--2022-08-22 16:51:05--  https://i.imgur.com/CbyiC7q.gif
Resolving i.imgur.com (i.imgur.com)... 146.75.36.193
Connecting to i.imgur.com (i.imgur.com)|146.75.36.193|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1227333 (1.2M) [image/gif]
Saving to: 'CbyiC7q.gif'

CbyiC7q.gif         100%[===================>]   1.17M  --.-KB/s    in 0.1s
```

AWS SageMaker serverless Inference (FaaS)

- SageMaker serverless inference


Amazon SageMaker
1 Create Model
Inference Container Image
Trained Model
2 Create Endpoint Configuration
3 Create Endpoint
SageMaker Serverless Endpoint

```
create_endpoint_config(
...
"ServerlessConfig": {
  "MemorySizeInMB": 2048,
  "MaxConcurrency": 20
...
}
```

```
create_endpoint(
...)
```

- 4 modes of ML model deployment
  - <u>Serverless inference </u>(sparse request, intermittent, infrequent)
  - <u>Real-time inference </u>(constant request, mili-sec response time)
  - Asynchronous
  - Batch transform
- Use case
  - Fraud detection (serverless inference) → Business grow, consistent traffic (real-time inference)

Serverless still require config

- memory selection
- max concurrency

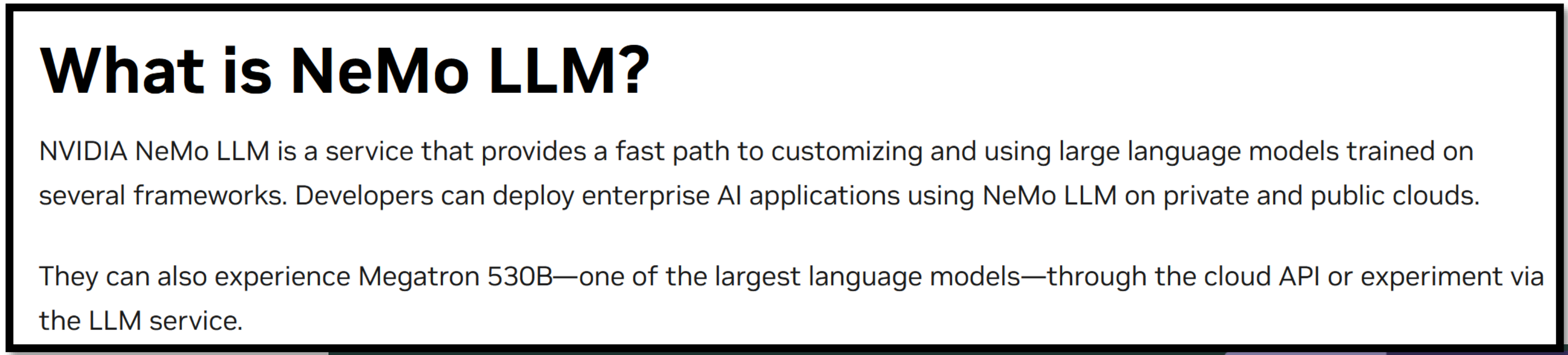
Real-time requires config

- compute/GPU selection
- Autoscaling policy

https://aws.amazon.com/blogs/machine-learning/deploying-ml-models-using-sagemaker-serverless-inference-preview/

One step further, ML as a service (serverless inference)

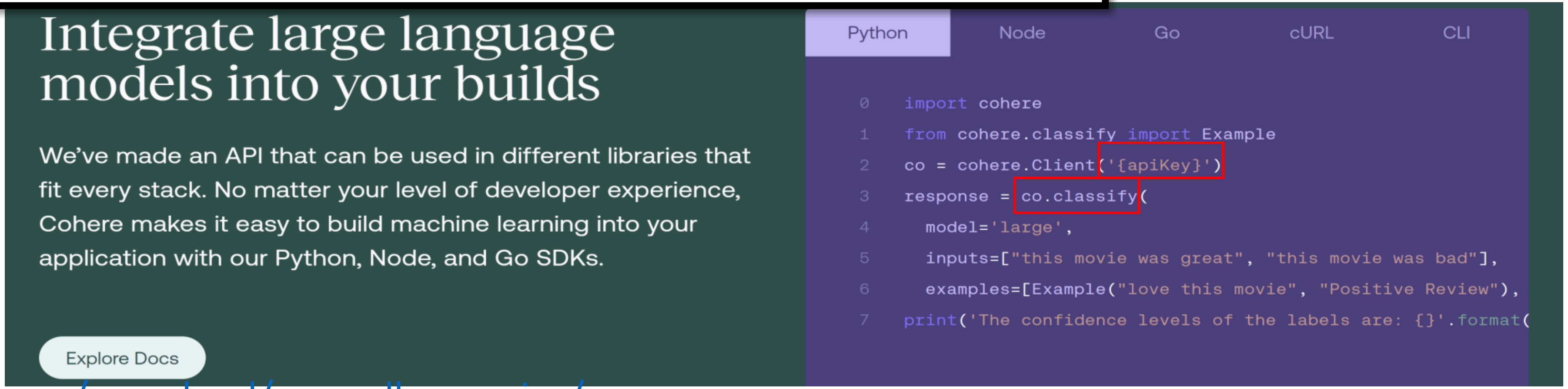
- Large language model, e.g., Nvidia NeMo API, Cohere.ai



What is NeMo LLM?

NVIDIA NeMo LLM is a service that provides a fast path to customizing and using large language models trained on several frameworks. Developers can deploy enterprise AI applications using NeMo LLM on private and public clouds.

They can also experience Megatron 530B—one of the largest language models—through the cloud API or experiment via the LLM service.



Integrate large language models into your builds

We've made an API that can be used in different libraries that fit every stack. No matter your level of developer experience, Cohere makes it easy to build machine learning into your application with our Python, Node, and Go SDKs.

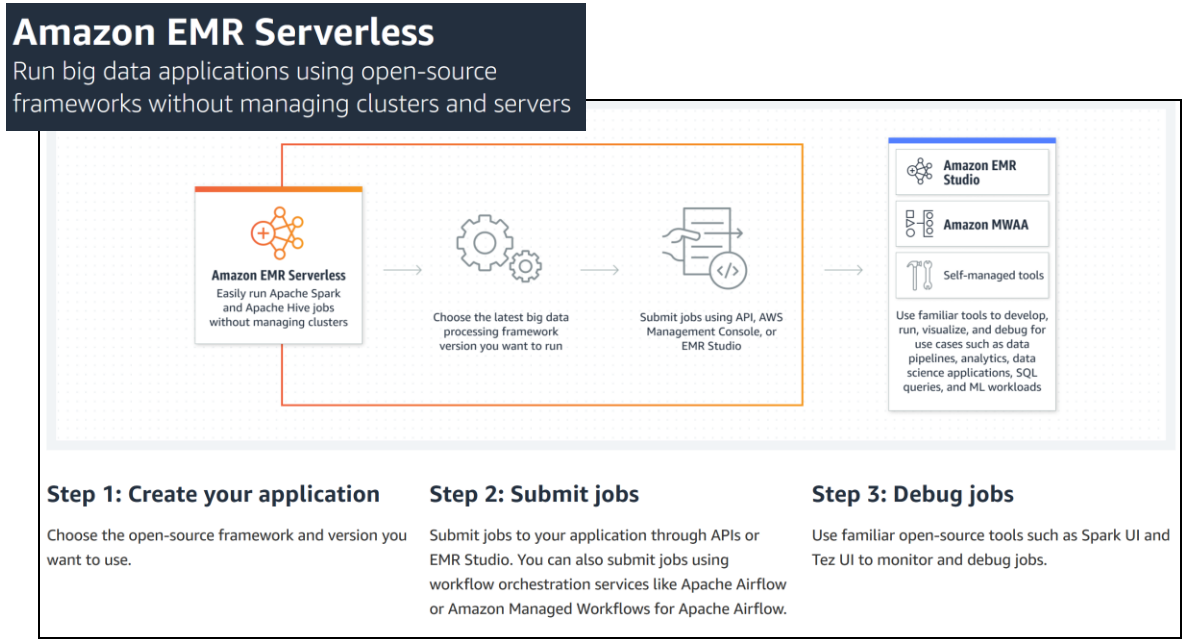
Explore Docs

Python Node Go cURL CLI

```
import cohere
from cohere.classify import Example
co = cohere.Client('{apiKey}')
response = co.classify(
  model='large',
  inputs=["this movie was great", "this movie was bad"],
  examples=[Example("love this movie", "Positive Review"),
print('The confidence levels of the labels are: {}'.format(
```

Reference:
https://www.nvidia.com/en-us/gpu-cloud/nemo-llm-service/
https://cohere.ai/
https://sambanova.ai/products/dataflow-as-a-service/

AWS serverless service (applications)



Amazon EMR Serverless

Run big data applications using open-source frameworks without managing clusters and servers

Amazon EMR Serverless
Easily run Apache Spark and Apache Hive jobs without managing clusters
Choose the latest big data processing framework version you want to run
Submit jobs using API, AWS Management Console, or EMR Studio
Amazon EMR Studio
Amazon MWAA
Self-managed tools
Use familiar tools to develop, run, visualize, and debug for use cases such as data pipelines, analytics, data science applications, SQL queries, and ML workloads

Step 1: Create your application

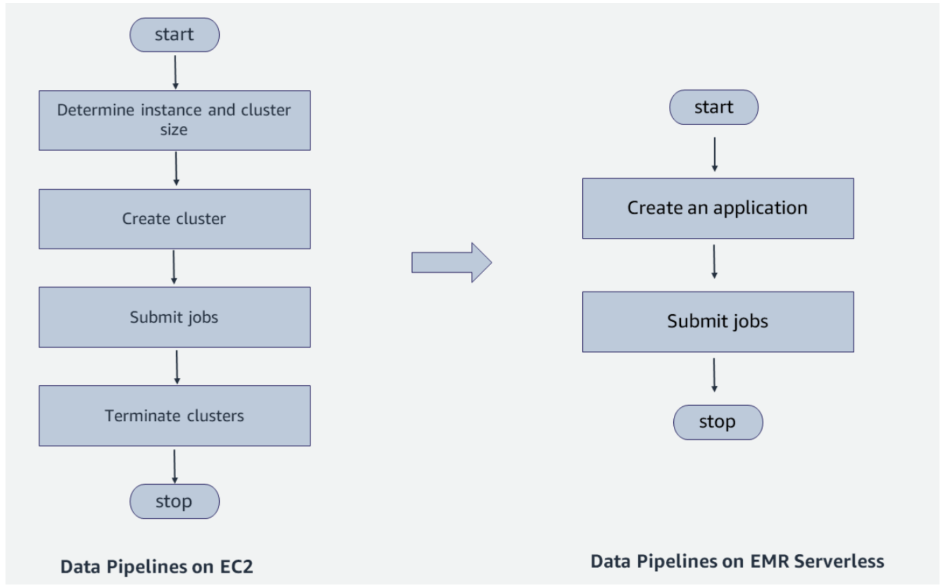
Choose the open-source framework and version you want to use.

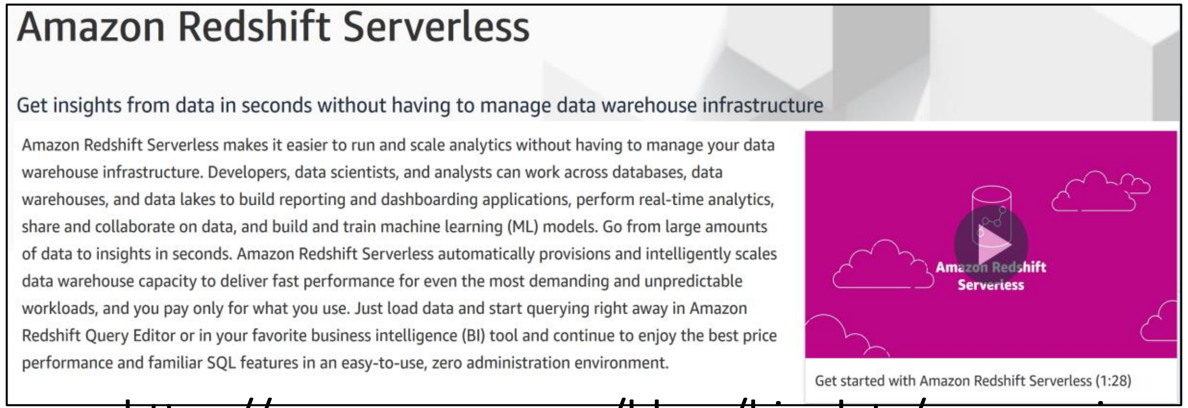
Step 2: Submit jobs

Submit jobs to your application through APIs or EMR Studio. You can also submit jobs using workflow orchestration services like Apache Airflow or Amazon Managed Workflows for Apache Airflow.

Step 3: Debug jobs

Use familiar open-source tools such as Spark UI and Tez UI to monitor and debug jobs.





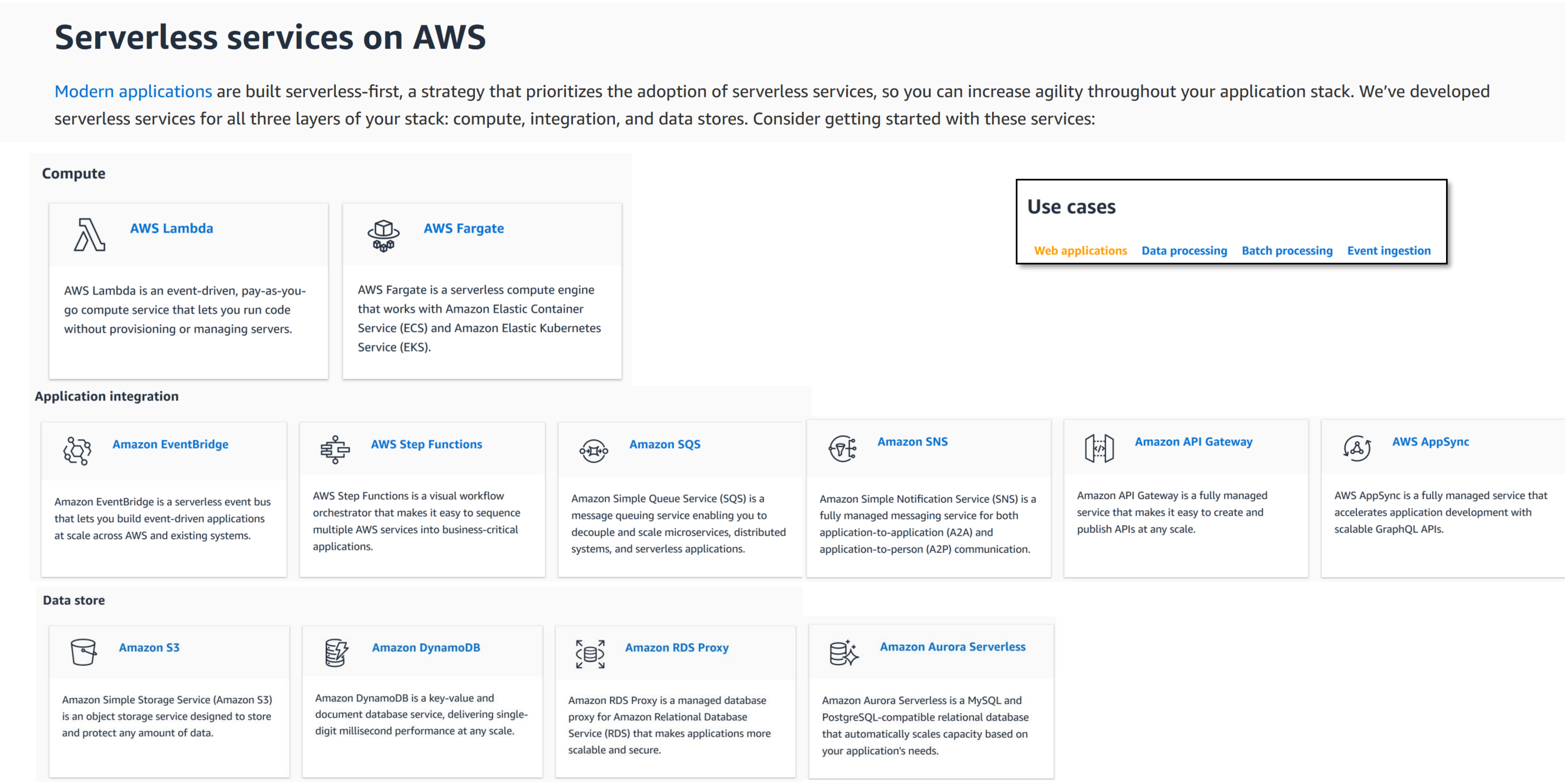
Amazon Redshift Serverless

Get insights from data in seconds without having to manage data warehouse infrastructure

Amazon Redshift Serverless makes it easier to run and scale analytics without having to manage your data warehouse infrastructure. Developers, data scientists, and analysts can work across databases, data warehouses, and data lakes to build reporting and dashboarding applications, perform real-time analytics, share and collaborate on data, and build and train machine learning (ML) models. Go from large amounts of data to insights in seconds. Amazon Redshift Serverless automatically provisions and intelligently scales data warehouse capacity to deliver fast performance for even the most demanding and unpredictable workloads, and you pay only for what you use. Just load data and start querying right away in Amazon Redshift Query Editor or in your favorite business intelligence (BI) tool and continue to enjoy the best price performance and familiar SQL features in an easy-to-use, zero administration environment.

Get started with Amazon Redshift Serverless (1:28)

Reference: https://aws.amazon.com/blogs/big-data/announcing-amazon-emr-serverless-preview-run-big-data-applications-without-managing-servers/



Serverless services on AWS

Modern applications are built serverless-first, a strategy that prioritizes the adoption of serverless services, so you can increase agility throughout your application stack. We've developed serverless services for all three layers of your stack: compute, integration, and data stores. Consider getting started with these services:

Use cases

Web applications Data processing Batch processing Event ingestion

Compute

AWS Lambda

AWS Lambda is an event-driven, pay-as-you-go compute service that lets you run code without provisioning or managing servers.

AWS Fargate

AWS Fargate is a serverless compute engine that works with Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS).

Application integration

Amazon EventBridge

Amazon EventBridge is a serverless event bus that lets you build event-driven applications at scale across AWS and existing systems.

AWS Step Functions

AWS Step Functions is a visual workflow orchestrator that makes it easy to sequence multiple AWS services into business-critical applications.

Amazon SQS

Amazon Simple Queue Service (SQS) is a message queuing service enabling you to decouple and scale microservices, distributed systems, and serverless applications.

Amazon SNS

Amazon Simple Notification Service (SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.

Amazon API Gateway

Amazon API Gateway is a fully managed service that makes it easy to create and publish APIs at any scale.

AWS AppSync

AWS AppSync is a fully managed service that accelerates application development with scalable GraphQL APIs.

Data store

Amazon S3

Amazon Simple Storage Service (Amazon S3) is an object storage service designed to store and protect any amount of data.

Amazon DynamoDB

Amazon DynamoDB is a key-value and document database service, delivering single-digit millisecond performance at any scale.

Amazon RDS Proxy

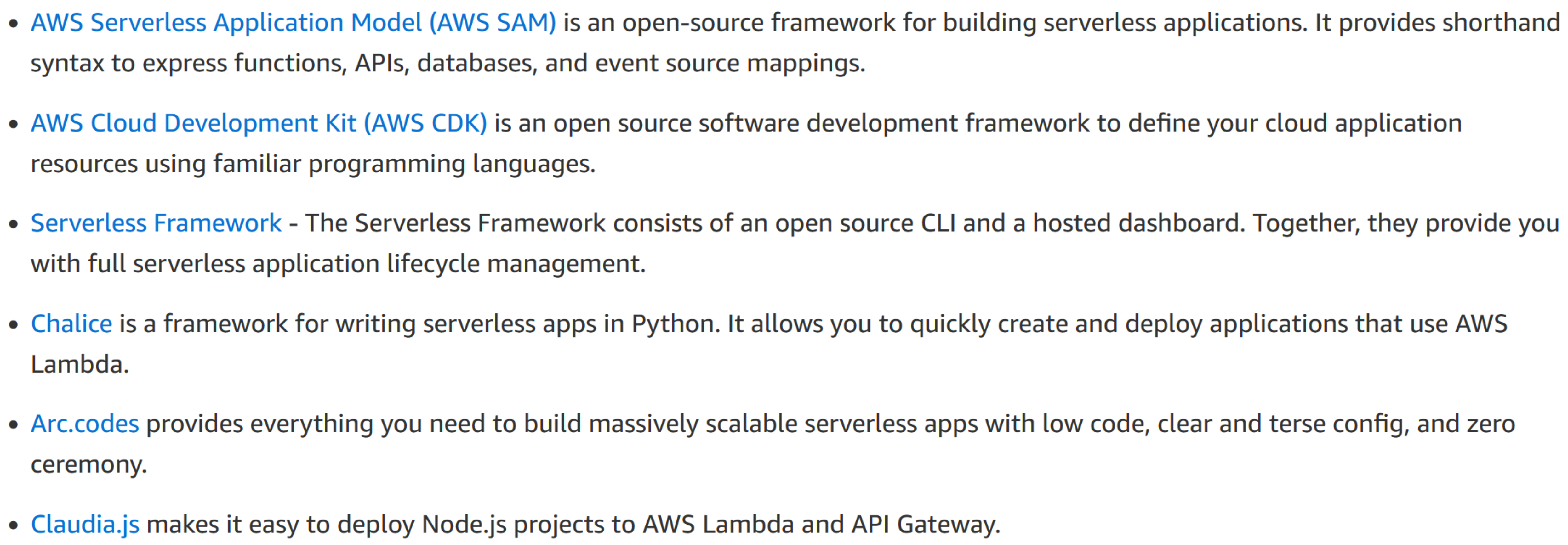
Amazon RDS Proxy is a managed database proxy for Amazon Relational Database Service (RDS) that makes applications more scalable and secure.

Amazon Aurora Serverless

Amazon Aurora Serverless is a MySQL and PostgreSQL-compatible relational database that automatically scales capacity based on your application's needs.

Reference: https://aws.amazon.com/serverless/

AWS serverless framework (developer tools)



- AWS Serverless Application Model (AWS SAM) is an open-source framework for building serverless applications. It provides shorthand syntax to express functions, APIs, databases, and event source mappings.
- AWS Cloud Development Kit (AWS CDK) is an open source software development framework to define your cloud application resources using familiar programming languages.
- Serverless Framework - The Serverless Framework consists of an open source CLI and a hosted dashboard. Together, they provide you with full serverless application lifecycle management.
- Chalice is a framework for writing serverless apps in Python. It allows you to quickly create and deploy applications that use AWS Lambda.
- Arc.codes provides everything you need to build massively scalable serverless apps with low code, clear and terse config, and zero ceremony.
- Claudia.js makes it easy to deploy Node.js projects to AWS Lambda and API Gateway.

Reference: https://aws.amazon.com/serverless/getting-started/

Recent serverless paper highlights (SOCC’21)

1. Llama: A Heterogeneous & Serverless Framework for Auto-Tuning Video Analytics Pipelines (Standford)
2. Faa$T: A Transparent Auto-Scaling Cache for Serverless Applications (Microsoft)
3. Atoll: A Scalable Low-Latency Serverless Platform (UW and Google)
4. Kraken : Adaptive Container Provisioning for Deploying Dynamic DAG applications in Serverless platforms (PSU)
5. Speedo: Fast dispatch and orchestration of serverless workflows (IIT)

Feedback

1. Concrete feature, well-defined Interface, direct reuse
2. Serverless essential service (data-intensive application)
3. Fine-grained resource scheduling (Google reference, plaque)
4. User view (security, cost saving, bill model)
5. Platform view (serverless vs PaaS, resource utilization, pro/cons, profits)