

Fondamenti MLOps:

parte 2

Programma di oggi

Faremo il deploy del modello su cui abbiamo lavorato la scorsa settimana. Un po' meno slides e un po' più esercizi sul campo, che saranno:

- Impostare una funzione AWS **Lambda** che effettui delle batch prediction ogni qualvolta vengano uploadati in un **bucket**
- Online serving di un modello tramite **EC2**
- (Se rimane tempo) Deploy di un modello su **SageMaker** tramite **mlflow**

Concetti dalla scorsa settimana

Efficienza computazionale per la messa in produzione dei modelli: passaggio da librerie per l'allenamento a librerie per l'inferenza (**ONNX**)

Esistenza di librerie end-to-end disegnate per aiutare data scientists meno esperti di IT ad affrontare il tema della messa in produzione (**mlflow**)

Batch vs **online** serving, esercizi di oggi affronteranno entrambe le modalità.

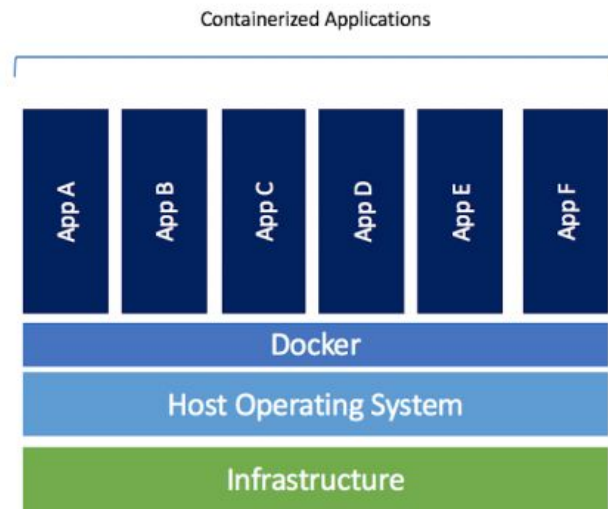
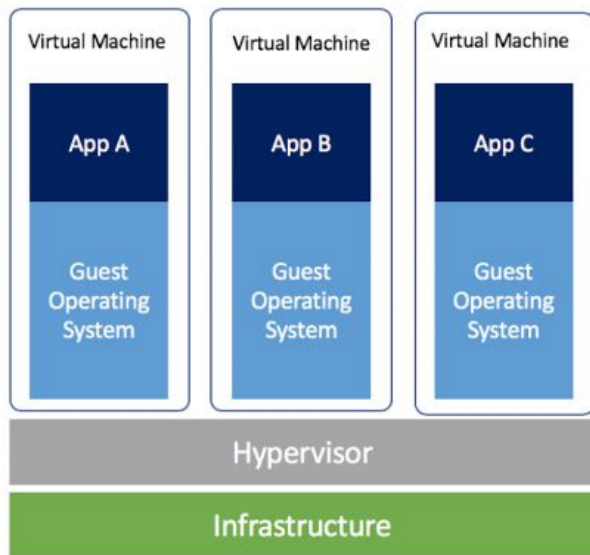
Modelli in produzione

La scorsa settimana abbiamo visto come impacchettare modelli per la messa in produzione.

Passo successivo: creare un ambiente **isolato** in cui operi il modello.

Come? **Macchine virtuali** o **containers**

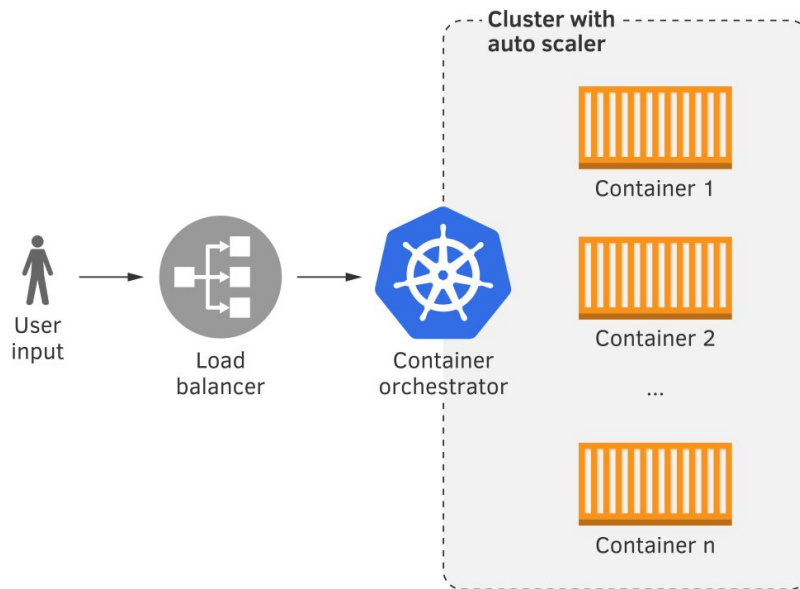
Containers vs Machine virtuali



Deployment modelli con containers

Online serving dei modelli può richiedere tra le altre cose:

- Autoscaling risorse computazionali
- Sistema orchestrazione container
- Load balancer
- Rest API



‘Machine Learning Engineering’, A. Burkov

Da on-prem a soluzione su cloud

Varie opzioni per deployment



- Sui propri server



- Su macchine virtuali all'interno di infrastruttura cloud



- Implementazione di alcuni servizi (ad esempio per il load balancing, auto-scaling, gestione API) usando soluzioni del fornitore cloud



- Servizi completamente gestiti da fornitore cloud



Da cloud computing a serverless computing

Con serverless computing si intende quando il cloud provider si prende cura della gestione delle risorse hardware che devono effettuare un determinato task.

Soluzioni serverless possono essere offerte per particolari tipi di task semplificati con l'idea di offrire un modo di ridurre costi di gestione.

Risorse computazionali create e distrutte all'esecuzione di ogni task.

MLOps su AWS

Esercizi di oggi → Casi d'uso di messa in produzione di modelli ML su AWS

Gratteremo solo la superficie dell'universo AWS, idea e' di mostrare alcuni principi di funzionamento tramite **esempi pratici**.

Non toccheremo argomenti importanti come l'uso di databases (verranno affrontati in sessioni successive).



Concetti base

Prima cosa: come interagire con l'infrastruttura AWS?

Opzioni:

1. **AWS Console**
2. **Command Line Interface**
3. **Software Development Kit per diversi linguaggi**

Tool terzi (non trattati oggi)

Infrastructure as code



AWS CLI

Command Line Interface → Permette di comunicare con i vari servizi disponibili su AWS.

Esercizio: Setup AWS CLI

```
git clone https://github.com/Clearbox-AI/Corso_MLOps.git
```

Step by step su:

```
https://github.com/Clearbox-AI/Corso_MLOps/blob/main/Esercizi_sessione2.md
```

Boto3

Software Development Kit per usare AWS via Python.

Installabile tramite pip. Offre 2 tipe di interfacce, una di basso e una di alto livello.

IAM, Policies and roles

Identity Access Management

Identity and Access Management (IAM) ✕

Pannello di controllo

▼ Gestione degli accessi

Gruppi di utenti

Utenti

Ruoli

Policy

Provider di identità

Impostazioni account

▼ Report di accesso

Analizzatore di accessi

Regole di archivio

Analizzatori

Impostazioni

Report sulle credenziali


Attività dell'organizzazione

Policy di controllo dei servizi (SCP)

Riepilogo Modifica

Nome del gruppo di utenti
corso_mlops

Data di creazione
April 23, 2021, 17:57 (UTC+02:00)

ARN
 iam:aws:iam:088093517335:group/corso_mlops

Utenti

Autorizzazioni


Access advisor

Utenti in questo gruppo (10) [Informazioni](#)

Un utente IAM è un'entità che crei in AWS per rappresentare la persona o l'applicazione che la utilizza per interagire con AWS.

↺ Rimuovi utenti Aggiungi utenti

< 1 > 🔍

<input type="checkbox"/>	Nome utente 	Gruppi	Ultima attività	Data di creazione
<input type="checkbox"/>	user8	1	Nessuno	15 ore fa
<input type="checkbox"/>	user3	1	Nessuno	15 ore fa
<input type="checkbox"/>	user9	1	Nessuno	15 ore fa
<input type="checkbox"/>	user6	1	Nessuno	15 ore fa
<input type="checkbox"/>	user5	1	Nessuno	15 ore fa
<input type="checkbox"/>	user7	1	Nessuno	15 ore fa
<input type="checkbox"/>	user1	1	15 ore fa	15 ore fa

IAM, Policies and roles

Identity and Access Management (IAM) e' il servizio responsabile del controllo degli accessi alle varie risorse su AWS.

All'interno di IAM e' possibile definire **Policies** ovvero autorizzazioni associate a diversi **Ruoli**.

Es: Questa macchina virtuale e' autorizzata ad accedere solamente a questo bucket.

Esempio policy

Definibili tramite file JSON.

Autorizza l'accesso a leggere da 2 bucket e scrivere in un terzo →

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "logs:PutLogEvents",  
8         "logs:CreateLogGroup",  
9         "logs:CreateLogStream"  
10      ],  
11      "Resource": "arn:aws:logs:*:*:*"  
12    },  
13    {  
14      "Effect": "Allow",  
15      "Action": [  
16        "s3:GetObject"  
17      ],  
18      "Resource": "arn:aws:s3:::batch-input-user1/*"  
19    },  
20    {  
21      "Effect": "Allow",  
22      "Action": [  
23        "s3:GetObject"  
24      ],  
25      "Resource": "arn:aws:s3:::model-user1/*"  
26    },  
27    {  
28      "Effect": "Allow",  
29      "Action": [  
30        "s3:PutObject"  
31      ],  
32      "Resource": "arn:aws:s3:::batch-output-user1/*"  
33    }  
34  ]  
35 }
```

Simple Storage Service (S3)

(serverless storage)



In ordine cronologico, il primo servizio offerto da AWS.

Archiviazione dati ad ‘**oggetti**’ → No file system

Oggetti definiti da un **bucket** di appartenenza e da una **key** identificativa. No struttura a cartelle.

Ogni oggetto può essere **versionato** e descritto da metadati.

Altre opzioni storage

S3 Glacier: sistema di archiviazione backup a lungo termine, economico ma ad alta latenza.

Elastic File Storage (EFS): sistema di archiviazione basato su filesystem condivisibile tra vari servizi e regioni. Più caro di S3.

Elastic Block Storage (EBS): disco associato esclusivamente a una singola macchina virtuale, prestazioni elevate.

Esercizio S3 (Console & Boto3)

Step by step su:

https://github.com/Clearbox-AI/Corso_MLOps/blob/main/Esercizi_sessione2.md



AWS Lambda

(serverless computing)

Function as a Service (Faas): possibilità' di implementare task descrivibili come funzioni stateless.

Risorse create e distrutte da AWS ad ogni esecuzione, gestione completamente effettuata da AWS da qui la definizione serverless.

Risorse pagate in funzione della memoria usata per fare girare la funzione e del numero di millisecondi richiesto a terminare un task.

Notevoli vantaggi in termini di costi per particolari tipi di applicazione.

AWS Lambda

```
26
27 def lambda_handler(event, context):
28     #print("Received event: " + json.dumps(event, indent=2))
29
30     # Get the object from the event and show its content type
31     bucket = event['Records'][0]['s3']['bucket']['name']
32     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
33     print(bucket, key)
34     try:
35         response = s3.get_object(Bucket=bucket, Key=key)
36         x = pd.read_csv(response['Body'])
37
38         s3_resource.Object('onnxmodel', 'model.onnx').download_file('/tmp/model.onnx')
39         sess = rt.InferenceSession('/tmp/model.onnx')
40
41         onnx_outputs = sess.run(None, process_inputs(x))
42         pd.DataFrame(onnx_outputs[1]).to_csv('/tmp/predictions.csv')
43         s3_resource.Object(bucket_name='batchpredictions', key='predictions.csv').upload_file('/tmp/predictions.csv')
44
45     return response['ContentType']
46
```

AWS Lambda: layers

Ciascun task Lambda gira all'interno di un ambiente Linux predefinito (basato su **Amazon Linux 2 Image**)

Librerie esterne possono essere impacchettate nel momento dell'upload di una funzione.

Librerie usate spesso possono essere 'salvate' come livelli (layers) utilizzabili da funzioni diverse. (*esempio nel file layer_file.zip scaricato prima*)

AWS Lambda: limiti d'uso

Limiti di memoria recentemente
augmentati a 10GB.

vCPU scalano
proporzionalmente alla
memoria usata (da 1 a 6).

Da qualche mese viene fornita
la possibilità di definire la
funzione tramite un'immagine
container.

Resource	Quota
Function memory allocation	128 MB to 10,240 MB, in 1-MB increments.
Function timeout	900 seconds (15 minutes)
Function environment variables	4 KB
Function resource-based policy	20 KB
Function layers	five layers
Function burst concurrency	500 - 3000 (varies per Region)
Invocation payload (request and response)	6 MB (synchronous)
	256 KB (asynchronous)
Deployment package (.zip file archive) size	50 MB (zipped, for direct upload)
	250 MB (unzipped, including layers)
	3 MB (console editor), 512 KB maximum for an individual file
Container image code package size	10 GB
Test events (console editor)	10
/tmp directory storage	512 MB

https://docs.amazonaws.cn/en_us/lambda/

AWS Lambda (partenza a freddo)



<https://mikhail.io/serverless/coldstarts/aws/>

Esercizio Lambda

Creazione di un servizio di 'batch predict' per il modello allenato la scorsa settimana



Esercizio Lambda

Steps:

- Creazione 3 buckets (per input, output, modello)
- Creazione policy per garantire accesso i bucket creati durante eseguimento funzione
- Creazione funzione Lambda da template
- Upload layer contenente env per l'inferenza
- Modifica funzione
- Test

Esercizio Lambda

Step by step su:

https://github.com/Clearbox-AI/Corso_MLOps/blob/main/Esercizi_sessione2.md





Elastic Compute Cloud

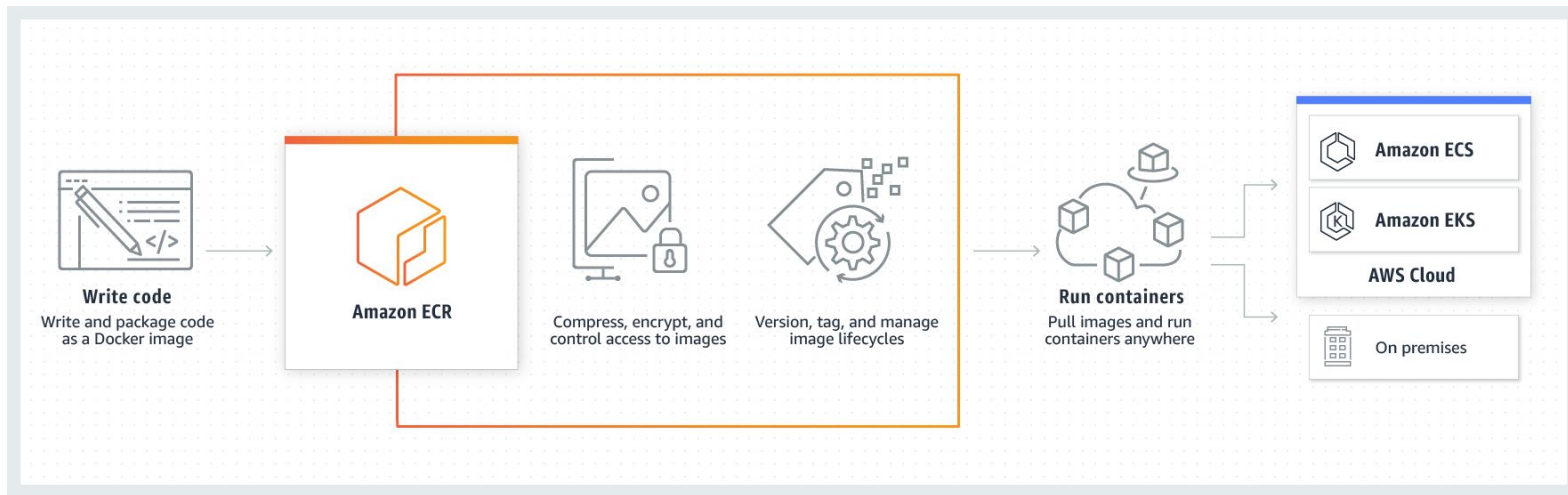
(macchine virtuali)

Secondo servizio offerto in ordine temporale (dopo S3) per il noleggio di macchine virtuali.

Grande varietà di istanze (ottimizzate per CPU, memoria, GPU, etc) e di immagini preimpostate.

Spot instances possono essere usate per ridurre i costi.

Elastic Container Registry



Registro privato dove salvare immagini container da mettere in produzione su vari servizi AWS o localmente (simile a Docker Hub)

API Gateway



Servizio per la creazione di API con cui interfacciare servizi AWS con richieste esterne.

Esercizio: online serving con EC2+Docker

Online serving del modello allenato la scorsa settimana usando un container fatto girare in un'istanza EC2.

Step by step su:

https://github.com/Clearbox-AI/Corso_MLOps/blob/main/Esercizi_sessione2.md

Elastic Container Service (ECS)

Esempio affrontato: 1 container su 1 istanza EC2. Cosa fare quando il carico aumenta e di conseguenza anche il numero container e istanze? **Servizio orchestrazione.**

ECS e' il servizio di orchestrazione nativo AWS, alternativa a Kubernetes o Docker Swarm. Gestisce gruppi di istanze EC2 preinstallate con Docker.

Integrabile con altri servizi AWS come l'**Elastic Load Balancer**.

SageMaker

Servizio completamente dedicato al machine learning.

- Preparazione e processamento dati
- Creazione di feature stores
- Allenamento modelli su Jupyter notebook hostati o con funzioni autopilot
- Creazione endpoints per il serving
- Servizi di monitoraggio e ri-etichettatura



Amazon SageMaker

SageMaker

Modelli possono essere allenati e messi in produzione in pochi click su macchine virtuali dedicate.

Contro: Livello di astrazione introdotta limita il range operativo dell'utente. In più a parità di macchine virtuali utilizzate per allenare e mettere in produzione i modelli il costo è 30%-40% più alto.



Amazon SageMaker

Esempio SageMaker

Deploy di un modello su SageMaker tramite mlflow.

Step by step su:

https://github.com/Clearbox-AI/Corso_MLOps/blob/main/Esercizi_sessione2.md

Alternative a AWS

PRODUCT	aws	Microsoft Azure	Google Cloud Platform
Virtual Servers	Instances	VMs	VM Instances
Platform-as-a-Service	Elastic Beanstalk	Cloud Services	App Engine
Serverless Computing	Lambda	Azure Functions	Cloud Functions
Docker Management	ECS	Container Service	Container Engine
Kubernetes Management	EKS	Kubernetes Service	Kubernetes Engine
Object Storage	S3	Block Blob	Cloud Storage
Archive Storage	Glacier	Archive Storage	Coldline
File Storage	EFS	Azure Files	ZFS / Avere
Global Content Delivery	CloudFront	Delivery Network	Cloud CDN
Managed Data Warehouse	Redshift	SQL Warehouse	Big Query

<https://www.cloudhealthtech.com/blog/aws-vs-azure-vs-google>



Thanks for Reading

Feel free to contact us:



www.clearbox.ai



shalini@clearbox.ai
giovannetti@clearbox.ai



[@ClearboxAI](https://twitter.com/ClearboxAI)

Alternative AWS

Alternativa piu' competitiva dal punto del costo per il noleggio di macchine virtuali e' Digital Ocean.

Vincolo: noleggio mensile

