

Corso di Sistemi Distribuiti e Cloud Computing

Corso di Laurea Magistrale in Ingegneria Informatica A.A. 2021/22
DIMES - Università degli Studi della Calabria



AMAZON WEB SERVICES (AWS)

Amazon Web Services

- È uno dei principali fornitori di servizi cloud del mondo.
- Fornisce più di 200 diversi servizi Cloud.
- Opera in 25 regioni del mondo, tra cui anche Milano (dal 2020).
- Ogni regione è costituita da 2+ zone di disponibilità.



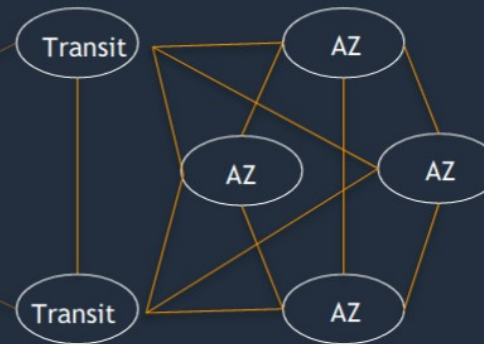
Regioni attive



Regioni annunciate

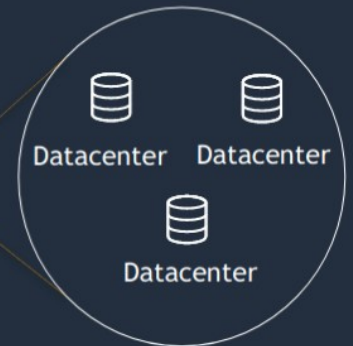
Spagna, Jakarta, e Giappone

AWS Region



Una **regione** è una posizione fisica nel mondo in cui abbiamo più **zone di disponibilità**.

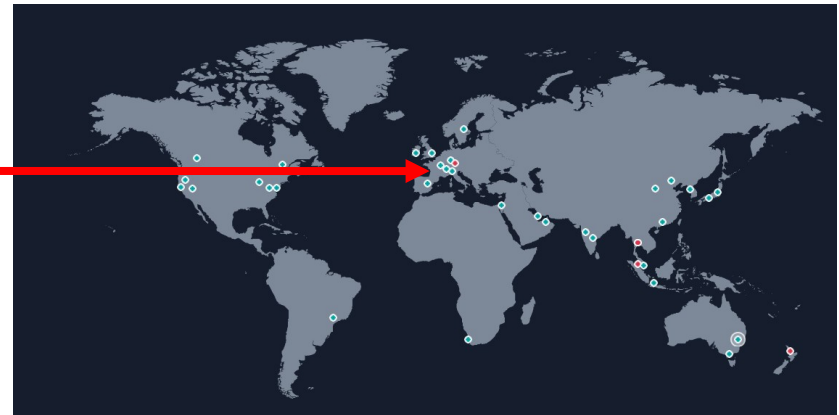
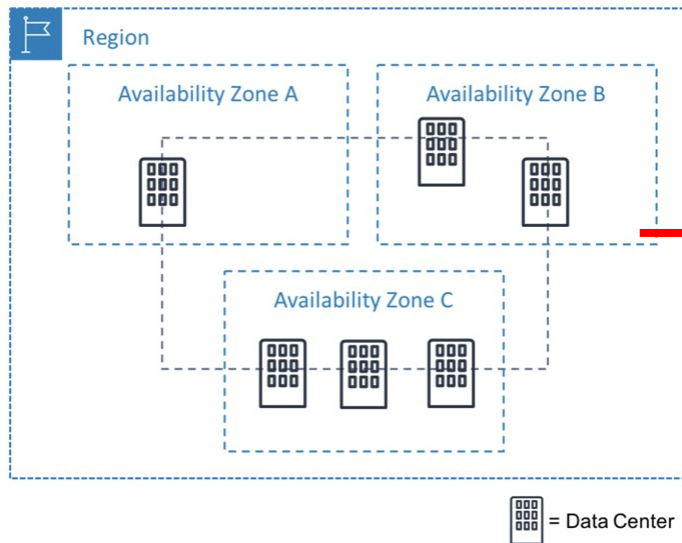
AWS Availability Zone (AZ)



Le **zone di disponibilità** sono costituite da uno o più data center, ciascuno con alimentazione, rete e connettività ridondanti, alloggiati in strutture separate.

Amazon Web Services

- Ogni **AWS Region** consiste di 2+ **Availability Zone (AZ)**, fisicamente separate ed isolate, che consistono in datacenter fisici con alimentazione e sistemi di connettività ridondanti ed autonomi.
- Le AZ servono a garantire high availability, fault tolerance, e scalabilità.



AWS - Livello gratuito

Esplora più di 100 prodotti e comincia a utilizzare AWS con il piano gratuito. Sono disponibili tre diversi tipi di piani a seconda del prodotto utilizzato. Clicca sull'icona sottostante per visualizzare le offerte.



Prove gratuite

Le offerte relative alle prove gratuite a breve termine iniziano dalla data di attivazione di un particolare servizio



Gratis per 12 mesi

Utilizza queste offerte per 12 mesi a partire dalla data di iscrizione ad AWS



Gratis senza limiti di tempo

I piani gratuiti sono sempre validi e sono disponibili per tutti i clienti AWS

CALCOLO	ARCHIVIAZIONE	DATABASE
Piano gratuito <small>GRATIS PER 12 MESI</small>	Piano gratuito <small>GRATIS PER 12 MESI</small>	Piano gratuito <small>GRATIS PER 12 MESI</small>
Amazon EC2	Amazon S3	Amazon RDS
750 ore	5 GB	750 ore
<small>al mese</small>	<small>di storage standard</small>	<small>al mese di istanze database db.t2.micro (solo motori di database applicabili)</small>
<small>Capacità di elaborazione ridimensionabile nel cloud.</small>	<small>Infrastruttura di storage di oggetti sicura, durevole e scalabile.</small>	<small>Servizio di database relazionale gestito per MySQL, PostgreSQL, MariaDB, Oracle BYOL o SQL Server.</small>
<small>750 ore al mese di istanze t2.micro o t3.micro</small>	<small>5 GB di storage standard</small>	
▼	▼	▼



<https://aws.amazon.com/it/free/free-tier/>

AWS - Services

- **Calcolo:** EC2, Lambda, Elastic Container Registry, Lightsail, Load Balancing
- **Storage:** S3, RDS, DynamoDB, Redshift
- **Machine Learning:** Rekognition, Lex, Comprehend
- **Networking e CDN:** CloudFront, Route 53
- **Sicurezza:** Cognito



AWS - Come si usa?

- I servizi di AWS possono essere configurati mediante:
 - Web Console
 - AWS Command Line Interface
 - SDK per i diversi linguaggi (Java, PHP, Python, Ruby, Javascript, C++,...)
- L'utente deve essere registrato ad AWS e selezionare una AWS Region sulla quale vuole operare.
- I servizi attivati in una regione, infatti, non possono essere gestiti da un'altra regione AWS.

Strumenti per creare in AWS

Strumenti per sviluppare e gestire le applicazioni su AWS

Cerca per linguaggio di programmazione

Sviluppa applicazioni facilmente su AWS nel linguaggio di programmazione che preferisci

[C++](#) [Vai](#) [Java](#) [JavaScript](#) [.NET](#) [Node.js](#) [PHP](#) [Python](#) [Ruby](#)



<https://aws.amazon.com/it/tools/>

Amazon EC2

- **Amazon Elastic Compute Cloud (Amazon EC2)** offre capacità di calcolo sicura e scalabile per gestire diversi carichi di lavoro
- Con il piano gratuito di AWS si possono usare 750 ore al mese per 12 mesi (solo per istante aderenti al piano)
- Esistono diverse tipologie di istanze di **macchine virtuali** EC2:
 - **Istanze on demand:** consentono di pagare per la capacità di elaborazione consumata all'ora o al secondo, a seconda delle istanze eseguite. Non occorrono impegni a lungo termine né pagamenti anticipati. La capacità di elaborazione può essere aumentata o diminuita, pagando solo per istanze effettivamente impiegate, in base alla tariffa oraria specificata.
 - **Istanze spot:** permettono di sfruttare le capacità EC2 inutilizzate all'interno del cloud AWS, ottenute con un processo simile ad un'asta online. Le istanze Spot sono disponibili con uno sconto fino al 90% rispetto ai prezzi delle istanze on demand.
 - **Istanze host dedicati:** è un server fisico EC2 dedicato al tuo utilizzo; offrono la possibilità di usare licenze software esistenti legate ai server, ad esempio licenze per Windows Server, SQL Server e SUSE Linux Enterprise Server.



Amazon EC2 - On-Demand vs Spot

	Spot Instances	On-Demand Instances
Ora di avvio	Può essere avviata immediatamente solo se è attiva la richiesta dell'istanza spot e se la capacità è disponibile.	Può essere avviata immediatamente solo se si effettua una richiesta di avvio manuale e la capacità è disponibile.
Capacità disponibile	Se la capacità non è disponibile, la richiesta dell'istanza spot continuerà a effettuare automaticamente la richiesta di avvio fino a quando la capacità non diventa disponibile.	Se la capacità non è disponibile quando si effettua una richiesta di avvio, si ottiene un errore di capacità insufficiente (ICE).
Tariffa oraria	Il prezzo orario per istanze spot varia in base alla fornitura a lungo termine e alla domanda.	Il prezzo orario per le Istanze on demand è statico.
Raccomandazione di ribilanciamento	Il segnale che Amazon EC2 emette per un'istanza spot in esecuzione quando presenta un rischio elevato di interruzione.	L'utente determina quando un'Istanza on demand viene interrotta (arrestata, ibernata o terminata).
Interruzione istanza	Un'istanza spot supportata da Amazon EBS può essere arrestata e avviata. Amazon EC2, inoltre, può interrompere una singola istanza spot se la capacità non è più disponibile.	L'utente determina quando un'Istanza on demand viene interrotta (arrestata, ibernata o terminata).



Amazon EC2 - Istanze

- E' possibile avviare istanze con diverse quantità di risorse:
 - CPU + RAM
 - Storage
 - Tipo di processori (ARM, x86, GPU)
 - Sistema operativo (Linux, Windows)
 - I costi variano in funzione del tipo di istanza e della quantità di risorse (si parte da poco più di 0.01\$/ora per le istanze on demand T3 micro.nano)

Nome dell'istanza ▲	Tariffa oraria on demand ▼	vCPU ▼	Memoria ▼	Storage ▼	Prestazioni di rete ▼
a1.medium	0,0255 USD	1	2 GiB	Solo EBS	Fino a 10 Gigabit
a1.large	0,051 USD	2	4 GiB	Solo EBS	Fino a 10 Gigabit
a1.xlarge	0,102 USD	4	8 GiB	Solo EBS	Fino a 10 Gigabit
a1.2xlarge	0,204 USD	8	16 GiB	Solo EBS	Fino a 10 Gigabit
a1.4xlarge	0,408 USD	16	32 GiB	Solo EBS	Fino a 10 Gigabit
a1.metal	0,408 USD	16	32 GiB	Solo EBS	Fino a 10 Gigabit
t4g.nano	0,0042 USD	2	0,5 GiB	Solo EBS	Fino a 5 Gigabit
t4g.micro	0,0084 USD	2	1 GiB	Solo EBS	Fino a 5 Gigabit
t4g.small	0,0168 USD	2	2 GiB	Solo EBS	Fino a 5 Gigabit
t4g.medium	0,0336 USD	2	4 GiB	Solo EBS	Fino a 5 Gigabit



Amazon EC2 - Creare un'istanza

- Accedere alla console di AWS → Servizio EC2 → Avvia un'istanza
- Sarà necessario selezionare:
 - il sistema operativo dell'immagine: Amazon Linux (AMI), Ubuntu, Windows, Red Hat, Suse
 - L'architettura: 64bit (x86) o 64 bit (ARM)
 - La size dell'istanza: esempio, t2.micro con 1vCPU 1GiB RAM
- Per accedere alle istanze è necessario indicare una coppia di chiavi pubblica/private per un accesso sicuro via SSH
 - Se non si dispone di una coppia di chiavi, il wizard consente di creare una.
 - La coppia di chiavi indicate in fase di creazione dell'istanza vengono automaticamente installate nella VM, in modo da consentire l'accesso all'istanza.
- E' necessario anche configurare un **Security Group** che consenta il traffico SSH (porta 22 di default) e quello HTTP/S (porta 80 e 443) in ingresso.

```
# Comando per connettersi all'istanza via SSH  
ssh -i <file.pem> ec2-user@<Public IP/Public DNS>
```



Amazon EC2 - Creare un'istanza

New EC2 Experience
Tell us what you think

Pannello di controllo EC2

Visualizzazione di EC2

Global

Eventi

Tag

Limiti

▼ Istanze

Istanze **New**

Tipi di istanza

Modelli di avvio

Richieste Spot

Savings Plans

Istanze riservate **New**

Host dedicati

Istanze pianificate

Prenotazioni di capacità

▼ Immagini

AMI **New**

Catalogo AMI

EC2 > Istanze > i-0c2ff91448ef630d3

Riepilogo dell'istanza per i-0c2ff91448ef630d3 (Istanza di prova) **Informazioni**

Aggiornato less than a minute fa

ID istanza

i-0c2ff91448ef630d3 (Istanza di prova)

Indirizzo IPv6

-

Tipo di nome host

Nome IP: ip-172-31-29-9.eu-west-1.compute.internal

Tipo di istanza

t2.micro

ID VPC

vpc-b916ccc0

Indirizzo IPv4 pubblico

54.194.78.228 | [indirizzo aperto](#)

Stato dell'istanza

In attesa

Nome DNS IP privato (solo IPv4)

ip-172-31-29-9.eu-west-1.compute.internal

Indirizzi IP elastici

-

Risultato di AWS Compute Optimizer

Consenso esplicito ad AWS Compute Optimizer per le raccomandazioni. | [Ulteriori informazioni](#)

Indirizzi IPv4 privati

172.31.29.9

DNS IPv4 pubblico

ec2-54-194-78-228.eu-west-1.compute.amazonaws.com | [indirizzo aperto](#)

Risponde al nome DNS della risorsa privata

IPv4 (A)

Indirizzo IP assegnato automaticamente

54.194.78.228 [IP pubblico]

Connettiti all'istanza **Informazioni**

Connettiti all'istanza i-0c2ff91448ef630d3 (Istanza di prova) utilizzando una qualsiasi di queste opzioni

EC2 Instance Connect

Gestore di sessioni

Client SSH

Console seriale EC2

ID Istanza

i-0c2ff91448ef630d3 (Istanza di prova)

Indirizzo IP pubblico

54.194.78.228

Nome utente

ubuntu

Nota: nella maggior parte dei casi, il nome utente ipotizzato è corretto. Tuttavia, leggi le istruzioni di utilizzo dell'AMI per verificare che il relativo proprietario non abbia modificato il nome utente predefinito.

Annulla

Connetti

Group name

Monitoraggio in corso

Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1022-aws x86_64)

* Documentation: <https://help.ubuntu.com>

* Management: <https://landscape.canonical.com>

* Support: <https://ubuntu.com/advantage>

System information as of Sat May 7 07:50:50 UTC 2022

System load: 0.23 Processes: 105

Usage of /: 18.6% of 7.69GB Users logged in: 0

Memory usage: 21% IPv4 address for eth0: 172.31.29.9

Swap usage: 0%

1 update can be applied immediately.

To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.

To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;

the exact distribution terms for each program are described in the

individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by

applicable law.

To run a command as administrator (user "root"), use "sudo <command>".

See "man sudo_root" for details.

ubuntu@ip-172-31-29-9:~\$

11

Loris Belcastro – Sistemi Distribuiti

Amazon EC2 - Creare un'istanza

- Un'istanza può essere associata ad uno o più gruppi di sicurezza, che ne definiscono le regole del firewall per consentire/bloccare il traffico in ingresso ed in uscita.
- Di default le istanze hanno dei security group molto restrittivi.

Regole in entrata

Regole in uscita

Tag

Regole in entrata (7)



Gestisci tag

Modifica le regole in entrata

Q Filtra regole del gruppo di sicurezza

< 1 > ⚙

<input type="checkbox"/>	Name ▾	ID della regola del ... ▾	Versione IP ▾	Tipo ▾	Protocollo ▾	Intervallo porte ▾	Origine
<input type="checkbox"/>	–	sgr-09631b4f31ccb8909	IPv6	HTTP	TCP	80	::/0
<input type="checkbox"/>	–	sgr-075d2051ed1088...	IPv4	Regola TCP personaliz...	TCP	33221	0.0.0.0/0
<input type="checkbox"/>	–	sgr-032176739a8ce60...	IPv6	HTTPS	TCP	443	::/0
<input type="checkbox"/>	–	sgr-09b7ccd01249d30f6	IPv6	Regola TCP personaliz...	TCP	33221	::/0
<input type="checkbox"/>	–	sgr-0040bd74a13bdaa...	IPv4	Regola TCP personaliz...	TCP	5601	0.0.0.0/0
<input type="checkbox"/>	–	sgr-0b98936b454d79...	IPv4	HTTPS	TCP	443	0.0.0.0/0
<input type="checkbox"/>	–	sgr-094df88c8956e837d	IPv4	HTTP	TCP	80	0.0.0.0/0



Amazon EC2 - Elastic IP

- Un'istanza viene avviata con un IP privato ed un IP pubblico temporaneo.
- Quando l'istanza viene riavviata, l'IP pubblico potrebbe cambiare
- Per rendere l'assegnazione dell'IP pubblico permanente, è necessario utilizzare il servizio Amazon Elastic IP
- Ogni istanza può avere assegnati più indirizzi IP (pubblici e/o privati).

EC2 > Indirizzi IP elastici > Assegna indirizzo IP elastico

Assegna indirizzo IP elastico [Informazioni](#)

Impostazioni indirizzo IP elastico [Informazioni](#)

Pool di indirizzi IPv4 pubblico

- ☒ Pool di indirizzi IPv4 di Amazon
- ☐ Indirizzo IPv4 pubblico da importare nel tuo account AWS (l'opzione è stata disabilitata perché non è stato trovato nessun pool) [Ulteriori informazioni](#)
- ☐ Pool di indirizzi IPv4 di proprietà del cliente (opzione disabilitata perché nessun pool di proprietà del cliente è stato trovato) [Ulteriori informazioni](#)

Indirizzi IP statici globali

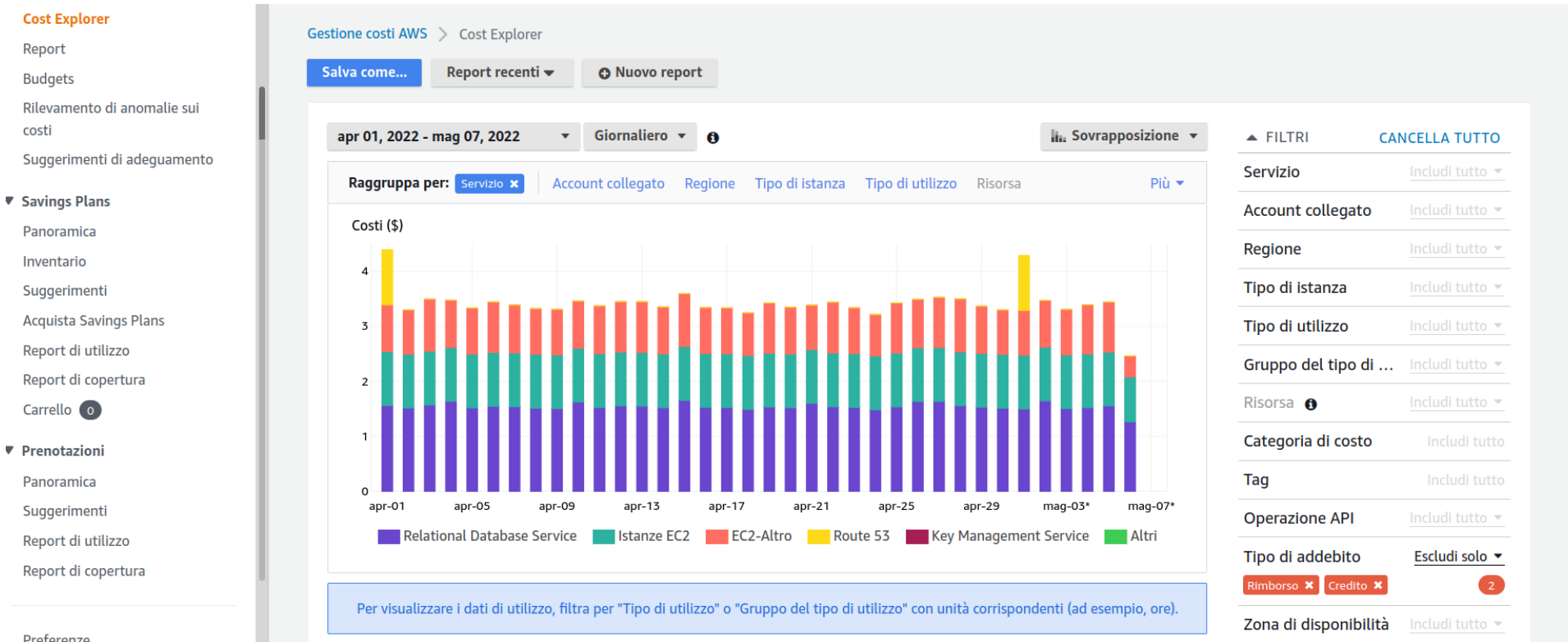
AWS Global Accelerator è in grado di fornire indirizzi IP statici globali annunciati in tutto il mondo utilizzando anycast dalle edge location di AWS. Questo può contribuire a migliorare la disponibilità e la latenza per il traffico degli utenti utilizzando la rete globale di Amazon. [Ulteriori informazioni](#)

[Crea acceleratore](#)



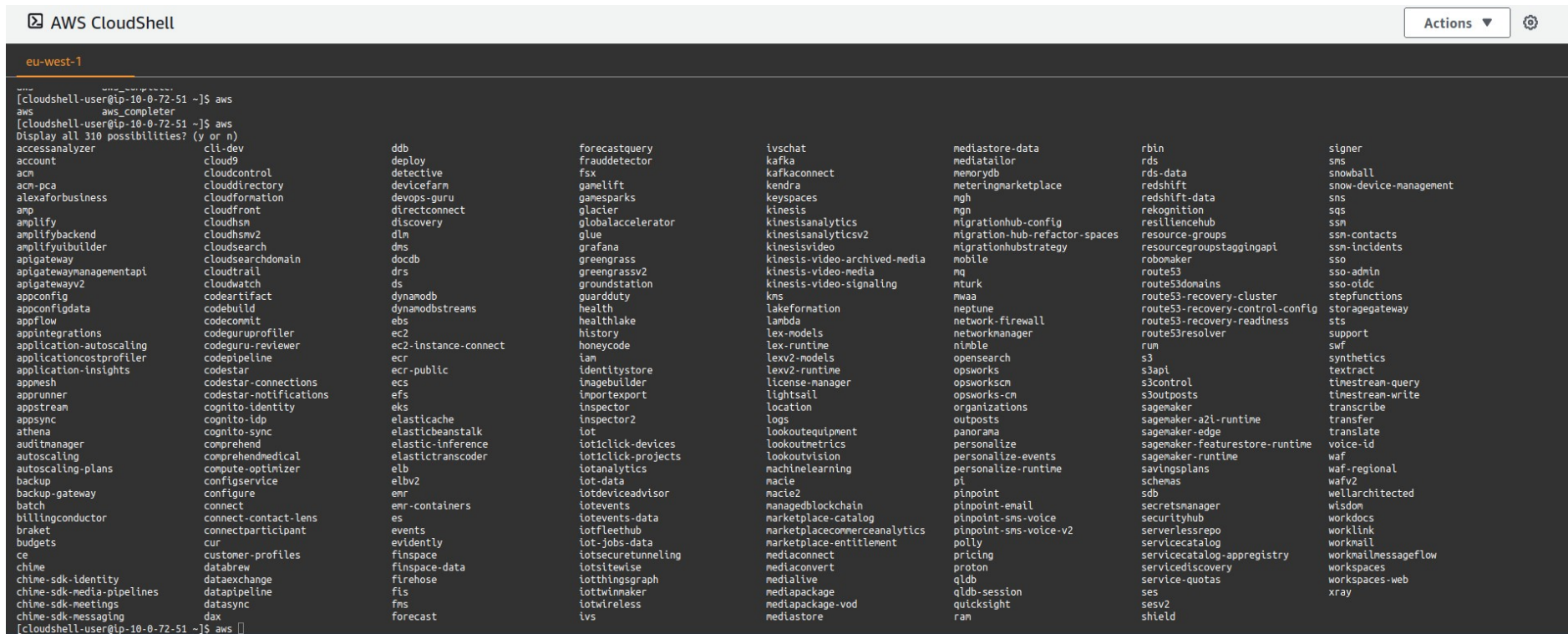
AWS Cost Management

- Quando si usa il Cloud è importante monitorare sempre i consumi!
- Si possono anche impostare dei budget e ricevere degli avvisi via email quando si superano le soglie di consumo stabilite.



AWS CloudShell

- AWS CloudShell è una shell basata su browser per la gestione da CLI delle risorse AWS.
- CloudShell è preautenticato con le credenziali dell'utente collegato alla console AWS.



The screenshot shows the AWS CloudShell interface. At the top, there's a header with the AWS CloudShell logo and an 'Actions' dropdown menu. Below the header, the terminal window displays the following content:

```
eu-west-1

[cloudshell-user@ip-10-0-72-51 ~]$ aws
aws
aws_completer
[cloudshell-user@ip-10-0-72-51 ~]$ aws
Display all 310 possibilities? (y or n)
accessanalyzer      cli-dev             ddb                 forecastquery       ivschat             mediastore-data     rbin                signer
account             cloud9              deploy             frauddetector       kafka               mediatailor         rds                 sms
acn                 cloudcontrol        detective           fsx                 kafkaconnect        memorydb            rds-data            snowball
acn-pca             clouddirectory       devicefarm          gameslift           kendra              meteringmarketplace redshift            snow-device-management
alexaforbusiness    cloudformation       devops-guru         glacier             keyspace            mgn                 rekognition         sns
amp                 cloudfront           directconnect       globalaccelerator   kinesis             migrationhub-config resiliencehub        sqs
amplify             cloudhsm             discovery           glue                kinesisanalytics   migrationhub-refactor-spaces ssm                 sns
amplifyuibuilder    cloudhsmv2           dns                 grafana             kinesisanalyticsv2 migrationhubstrategy ssm-incidents      sso
apigateway           cloudsearch          docdb              greengrass          kinesis-video-archived-media mobile               stepfunctions
apigatewaymanagementapi cloudtrail            drs                 greengrassv2        kinesis-video-media mturk                sso-admin
apigatewayv2         cloudwatch           ds                  groundstation       kinesis-video-signaling mq                    sso-oidc
appconfig            codeartifact         dynamodb           guardduty            kms                  neptune              stepfunctions
appconfigdata        codebuild            dynamodbstreams   health              lakeformation       network-firewall    storagegateway
appflow              codecommit           ebs                 healthlake          lambda              networkmanager      sts
appintegrations      codeguruprofiler     ec2                 history             lex-models           nimble               support
application-autoscaling codeguru-reviewer    ec2-instance-connect honeycode            lex-runtime           nimble               swf
applicationcostprofiler codepipeline          ecr                 iam                  lexx2-models         opensearch           synthetics
application-insights  codestar             ecr-public          identitystore        license-manager       opsworks             textract
appmesh              codestar-connections ecs                   imagebuilder         importexport          opsworks-cm          s3control            timestream-query
apprunner            cognito-identity     eks                  inspector            inspector2            organizations        s3outposts           timestream-write
appstream            cognito-idp          elasticache          iot                  iotclick-devices     outlookvision        sagemaker             transcribe
appsync              cognito-sync          elastic-inference    iotclick-projects   iotanalytics          machinelearning      sagemaker-a2i-runtime transfer
athena                comprehend            elastictranscoder    iot-data             iotdeviceadvisor     macie                 sagemaker-edge        translate
auditmanager          comprehendmedical    elb                  iotjobsdata          iotsecuretunneling   macle2                sagemaker-featurestore-runtime voice-id
autoscaling           compute-optimizer    elbv2                iotthingsgraph       iottwinmaker          managedblockchain     sagemaker-runtime     waf
autoscaling-plans     config               enr                  iotwireless          iotwireless           marketplace-catalog  savingsplans          waf-regional
backup                connect               enr-containers       iotwireless          iotwireless           marketplace-entitlement pi                     wafv2
backup-gateway        connect-contact-lens es                     iotwireless          iotwireless           marketplace-entitlement pricing                wellarchitected
batch                 connectparticipant   events              iotwireless          iotwireless           mediaconnect          qldb                  workdocs
billingconductor      cur                  evidently            iotwireless          iotwireless           mediaconvert          qldb-session          worklink
braket                customer-profiles    fincspace            iotwireless          iotwireless           medialive             quicksight            workmail
budgets               dataexchange         firehose             iotwireless          iotwireless           mediapackage          ram                    workmailmessageflow
ce                    datapipeline         fms                  iotwireless          iotwireless           mediapackage-vod     rbin                  workspaces
chime                  datasync             forecast              iotwireless          iotwireless           mediastore            rds                    workspaces-web
chime-sdk-identity    dax                  forecast              iotwireless          iotwireless           mediastore            rds-data              xray
chime-sdk-media-pipelines datasync             forecast              iotwireless          iotwireless           mediastore            rds-data              xray
chime-sdk-meetings    dax                  forecast              iotwireless          iotwireless           mediastore            rds-data              xray
chime-sdk-messaging   dax                  forecast              iotwireless          iotwireless           mediastore            rds-data              xray
```



AWS CloudShell - Esempi

Create a new security group in our VPC:

```
$ aws ec2 create-security-group --group-name my-sg \
--description "My security group" --vpc-id vpc-12345
```

See the properties of any security group:

```
$ aws ec2 describe-security-groups --group-ids <groupId>
```

set inbound traffic rules:

```
$ aws ec2 authorize-security-group-ingress --group-id <ID> \
--protocol tcp --port 22 --cidr 0.0.0.0/0
```

```
$ aws ec2 authorize-security-group-ingress --group-id <ID> \
--protocol tcp --port 80 --cidr 0.0.0.0/0
```

```
$ aws ec2 describe-security-groups --group-ids <ID DEL GRUPPO>
```



AWS CloudShell - Esempi

Run an EC2 instance

```
$ aws ec2 run-instances --image-id <ID AMI> --count 1 \  
--instance-type t2.nano \  
--key-name <MyKeyPair> --security-group-ids <sgId> \  
--subnet-id <subnetId> --associate-public-ip-address
```

Associate the instance with a tag:

```
$ aws ec2 create-tags --resources <instID> \  
--tags Key=Name,Value=SDCC
```

Get information about active instances using different filters:

```
$ aws ec2 describe-instances \  
--filters "Name=tag:Name,Values=MyInstance"  
  
$ aws ec2 describe-instances \  
--filters "Name=instance-type,Values=t2.nano"
```

Terminate the instance:

```
$ aws ec2 terminate-instances --instance-ids <ID>
```



Object Storage

- **Object Storage:** architettura di archiviazione che gestisce i dati come oggetti.
 - I dati sono memorizzati all'interno di un **bucket** definito nel proprio account utente e sono caratterizzati da un **URI** univoco.
 - Viene usato spesso per backup di grandi volumi di dati, video sorveglianza, audio.
- **PRO**
 - Garantisce ridondanza dei dati e può memorizzare fino a Petabyte di dati.
 - E' consigliata per memorizzare file statici (video o immagini di macchine virtuali).
 - Ad ogni oggetto è possibile associare una serie di metadati e dei permessi di lettura/scrittura.
- **CONTRO:**
 - Impossibile bloccare i file: tutti gli utenti con accesso al cloud, alla rete o al dispositivo hardware possono accedere agli oggetti archiviati.
 - Prestazioni più lente rispetto ad altri tipi di archiviazione.
 - Impossibile modificare una singola porzione di un file: una volta creato un oggetto, non è possibile modificare l'oggetto; puoi solo ricreare un nuovo oggetto.



Amazon S3

- Amazon Simple Storage Service (Amazon S3) è un servizio di archiviazione di **oggetti** che offre scalabilità, disponibilità dei dati, sicurezza e prestazioni elevate.
- Può essere usato per archiviare qualsiasi quantità di dati binari.
- Consente di organizzare i dati e configurare controlli di accesso.



Amazon S3 Glacier

- Amazon S3 Glacier è un servizio di archiviazione a costi estremamente contenuti, inferiori a quelli di S3, che consente la memorizzazione sicura e durevole per backup e archiviazione dei dati a lungo termine, ai quali si accede **raramente**.
- Durabilità degli oggetti su più zone di disponibilità pari al 99,999999999%
- Resilienza dei dati in caso di distruzione di una intera zona di disponibilità
- Progettato per una disponibilità di dati del 99,9% in un anno
- Nella classe Glacier Deep Archive ha costi estremamente bassi e garantisce la salvaguardia digitale di dati a cui si potrebbe accedere 1-2 volte all'anno.
 - Prevede tempi di recupero dei dati elevati (fino a 12 ore). Tutti i dati sono replicati e archiviati in almeno tre zone di disponibilità geograficamente distribuite



Amazon S3 - CLI

```
# s3 make bucket (create bucket)
```

```
aws s3 mb s3://mybucket --region us-west-2
```

```
# s3 remove bucket
```

```
aws s3 rb s3://mybucket
```

```
aws s3 rb s3://mybucket --force
```

```
# s3 list commands
```

```
aws s3 ls
```

```
aws s3 ls s3://mybucket
```

```
aws s3 ls s3://mybucket --recursive
```

```
aws s3 ls s3://mybucket --recursive --human-readable --summarize
```

```
# s3 copy command example
```

```
$ aws s3 cp myimage.jpg s3://bucket
```

```
# s3 move command example
```

```
$ aws s3 mv myimage.jpg s3://bucket
```

```
# s3 copy command example
```

```
$ aws s3 rm s3://mybucket/myimage.jpg
```



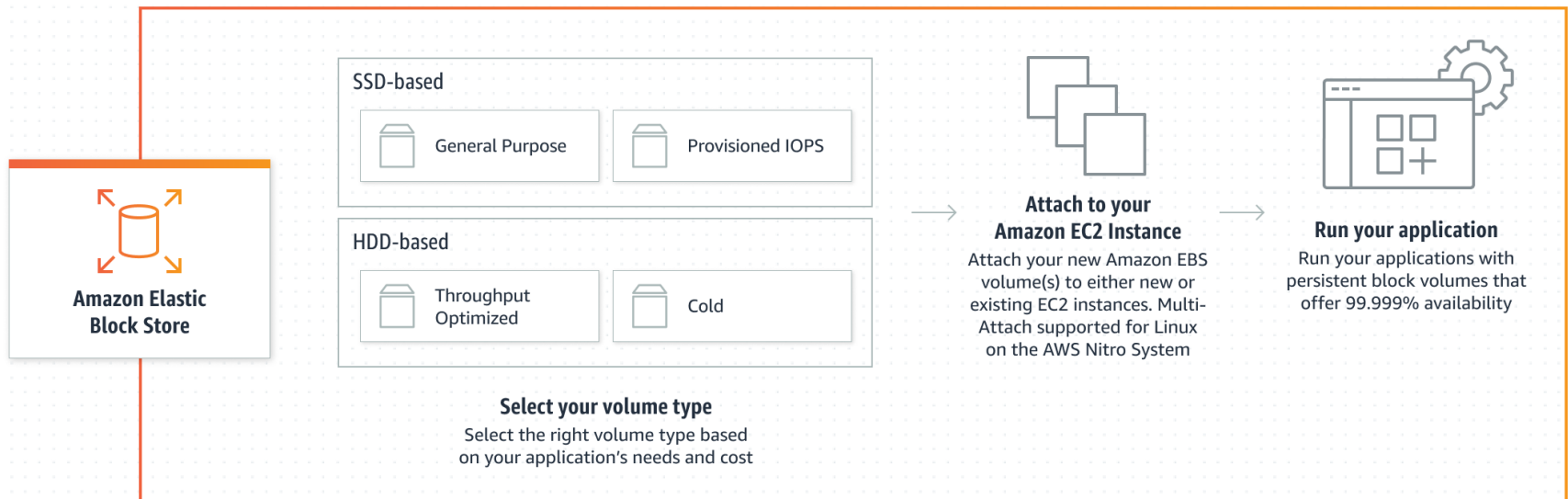
Block Storage

- **Block Storage:** architettura di archiviazione in cui i dati vengono suddivisi in blocchi di dimensione fissa e archiviati separatamente con identificatori univoci.
 - I blocchi possono essere archiviati in diversi ambienti, come un blocco in Windows e il resto in Linux.
 - Quando un utente recupera un blocco, il sistema di archiviazione riassembla i blocchi in un'unica unità.
 - L'archiviazione a blocchi è l'archiviazione predefinita sia per l'unità disco rigido che per i dati aggiornati di frequente.
- **PRO**
 - Grande velocità in fase di lettura e scrittura quando tutti i blocchi sono memorizzati localmente o letti/chiusi contemporaneamente.
 - Alta affidabilità perché i blocchi sono entità self-contained, che possono essere ridondate.
 - Ogni blocco può essere modificato, senza necessità di ricrearlo per intero.
- **CONTRO:**
 - Non consente l'uso di metadati.
 - Limitate capacità di ricerca in presenza di grandi volumi di blocchi.
 - Costi molto alti.



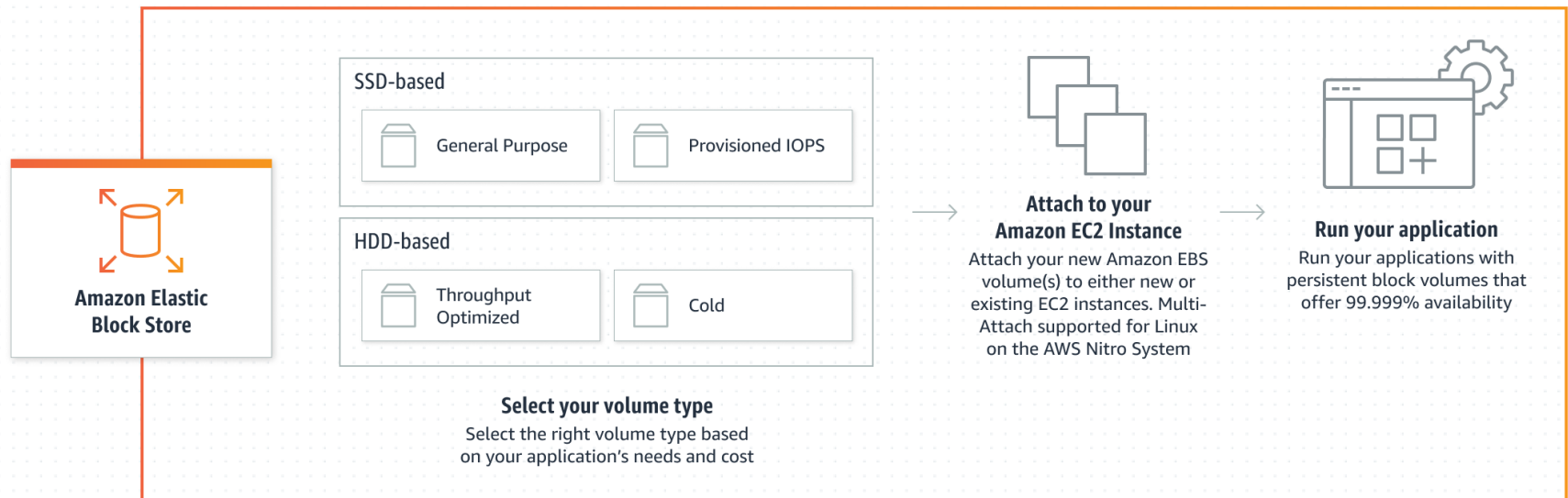
Amazon EBS

- **Amazon Elastic Block Store** (Amazon EBS) è un servizio di archiviazione a **blocchi** scalabile, ad alte prestazioni e facile da utilizzare, progettato per essere usato con Amazon Elastic Compute Cloud (Amazon EC2).
- Può essere usato per archiviare qualsiasi quantità di dati binari.
- I volumi EBS possono essere connessi facilmente ad istanze EC2.
- Può essere configurato per migliorare le prestazioni (IOPS e/o MB/s) e ottimizzare i costi.



Amazon EBS

- **Amazon Elastic Block Store** (Amazon EBS) è un servizio di archiviazione a **blocchi** scalabile, ad alte prestazioni e facile da utilizzare, progettato per essere usato con Amazon Elastic Compute Cloud (Amazon EC2).
- Può essere usato per archiviare qualsiasi quantità di dati binari.
- I volumi EBS possono essere connessi facilmente ad istanze EC2.
- Può essere configurato per migliorare le prestazioni (IOPS e/o MB/s) e ottimizzare i costi.



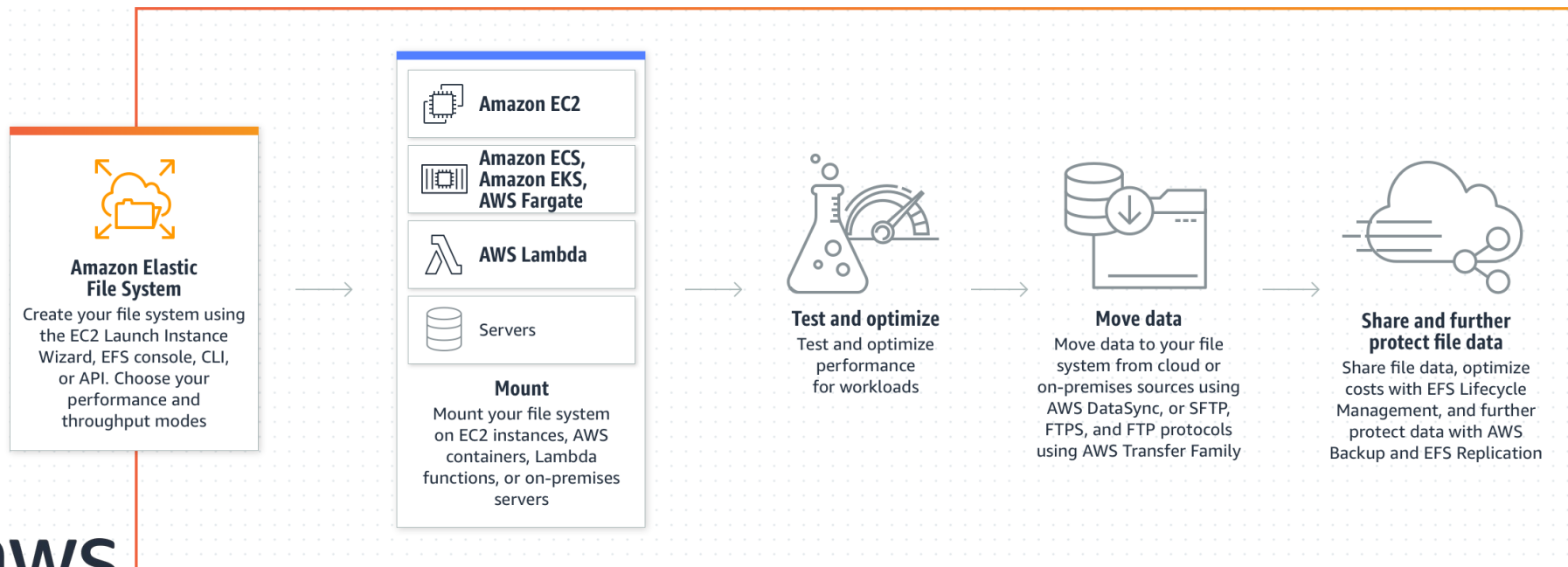
File Storage

- **File Storage:** architettura di archiviazione in cui i dati salvati all'interno di file con una particolare estensione (es. .jpg, .doc, .txt).
 - L'archiviazione dei file utilizza una struttura gerarchica in cui i file sono organizzati dall'utente in cartelle e sottocartelle, il che semplifica la ricerca e la gestione dei file.
 - Per accedere a un file, l'utente seleziona o immette il percorso del file, che include le sottodirectory e il nome del file.
 - La maggior parte degli utenti gestisce l'archiviazione dei file tramite un semplice file system.
- **PRO**
 - Facile accesso su piccola scala: con pochi file, gli utenti possono facilmente individuare e fare clic sul file desiderato.
 - Facile da usare per la maggior parte degli utenti.
 - Possibilità di organizzare i contenuti in cartelle.
 - Possibilità di impostare diritti di accesso/condivisione file/blocco file a livello utente o di proteggere i file con password.
- **CONTRO:**
 - Difficile da gestire in presenza di grandi quantità di file e/o di sottocartelle annidate.
 - Limitate capacità di gestire grandi moli di dati non strutturati (es. social media, IoT data).
 - Costi molto alti.



Amazon EFS

- Amazon Elastic File System (Amazon EFS) è un file system semplice, serverless, impostabile in maniera permanente ed elastico (si espande e si riduce automaticamente quando aggiungi e rimuovi file senza bisogno di gestione o provisioning).
- Può essere usato per condividere di dati tra più istanze EC2 o server.
- E' solitamente più costoso e più lento delle altre soluzioni di storage.



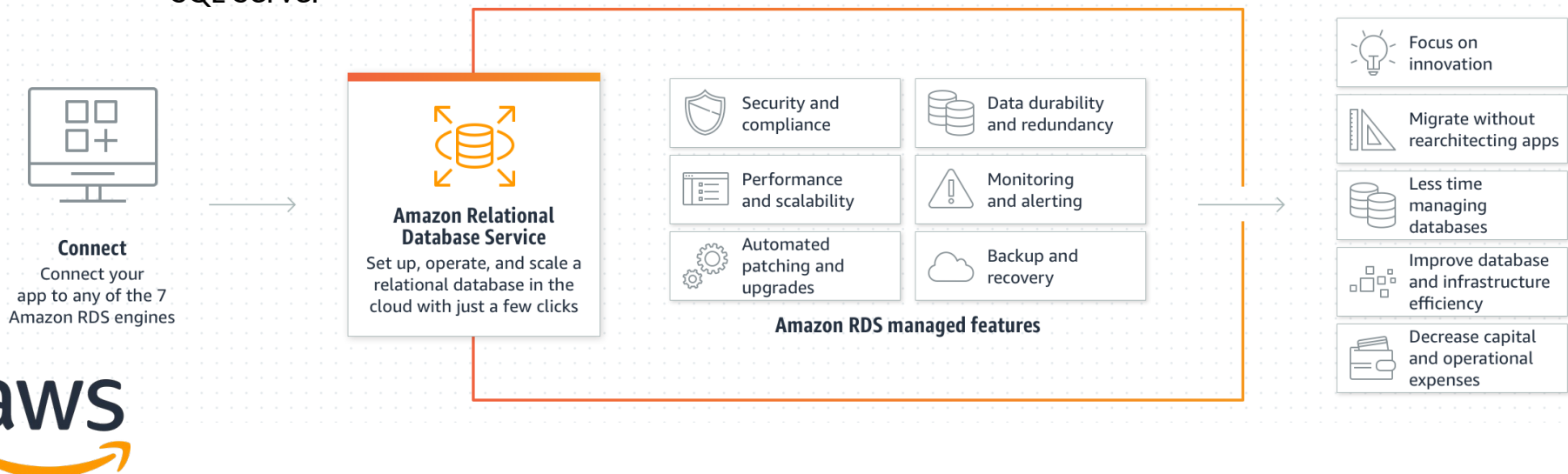
Amazon S3 vs EBS vs EFS

	EBS	EFS	S3
STORAGE SIZE	Max -16 TB	Unlimited	Unlimited
STANDARD PRICE (GB-MONTH)	0.10 USD	0.30 USD	0.023 USD
MAX FILE SIZE	Unlimited	47.9 TiB Per File	5TB Per File
ACCESSIBILITY	Single EC2	Multiple EC2s	Over Internet
DATA STORAGE	Data stays in the same Availability zone	Data stays in the region	Data stays in the region
AVAILABILITY	Cannot withstand AZ failure	Can survive one AZ failure.	Can withstand two concurrent AZ failures



Amazon RDS

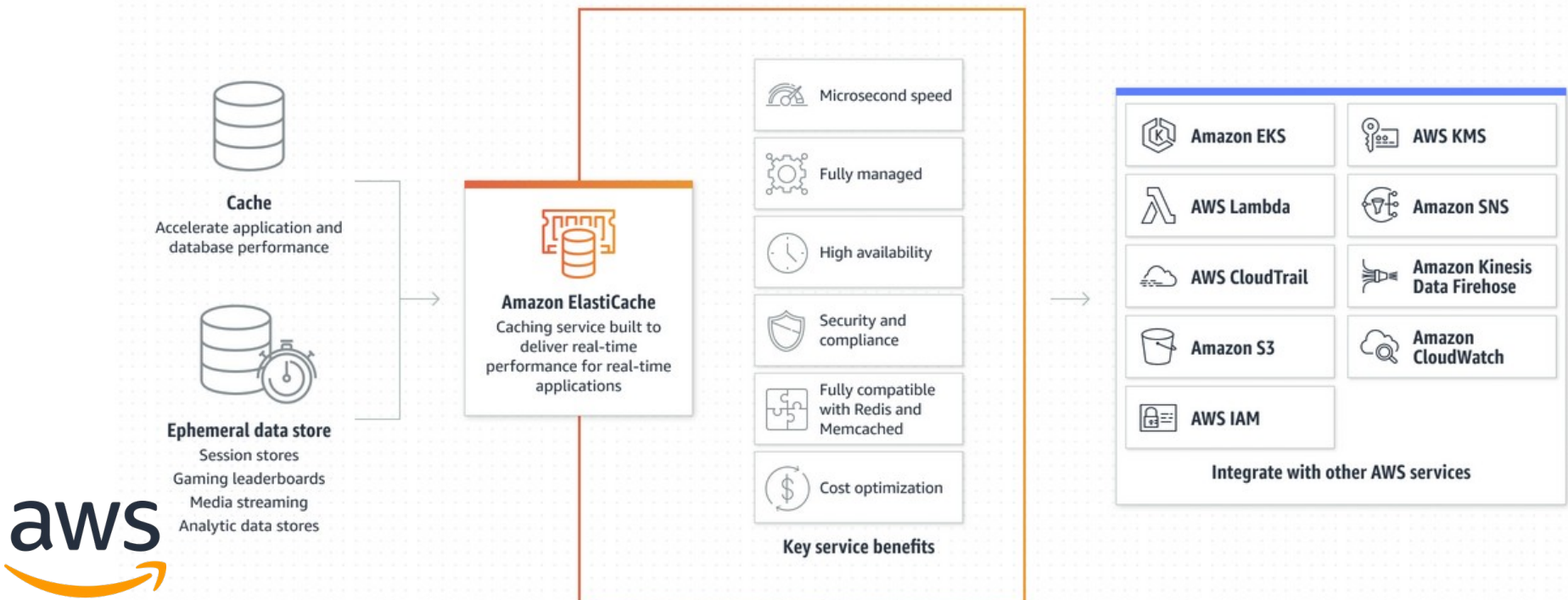
- Amazon Relational Database Service (RDS) è una raccolta di servizi gestiti che rende semplice impostare, operare e scalare i database nel cloud.
- Consente di scegliere tra sette motori popolari:
 - Amazon Aurora con compatibilità con MySQL
 - Amazon Aurora con compatibilità con PostgreSQL
 - MySQL
 - MariaDB
 - PostgreSQL
 - Oracle
 - SQL Server



Amazon ElastiCache

Amazon ElastiCache fornisce un servizio di cache in-memory per accelerare applicazioni web, con possibilità di scalare fino a centinaia di milioni di operazioni/secondo.

- Supporta due open-source cache in memoria motori:
 - **Memcached**: ampiamente adottato memorizzazione nella cache degli oggetti di memoria sistema
 - **Redis**: storage open-source di tipo chiave-valore in-memory che supporta strutture dati come set e elenchi ordinati;
- ElastiCache supporta master/replica slave e Multi-AZ



Amazon CloudFront

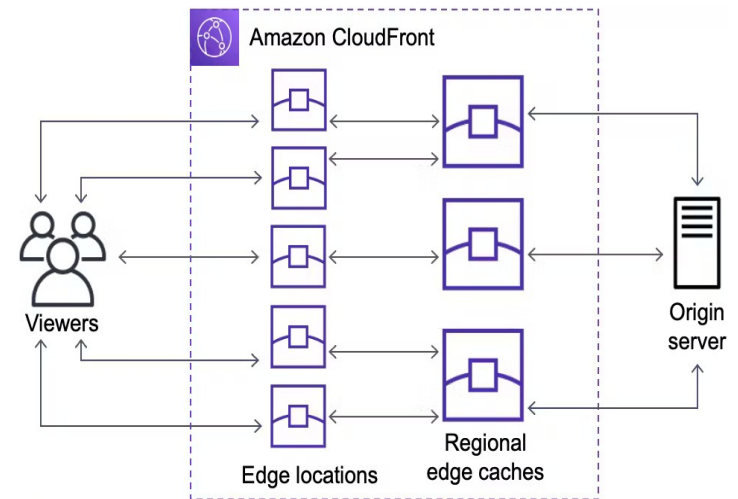
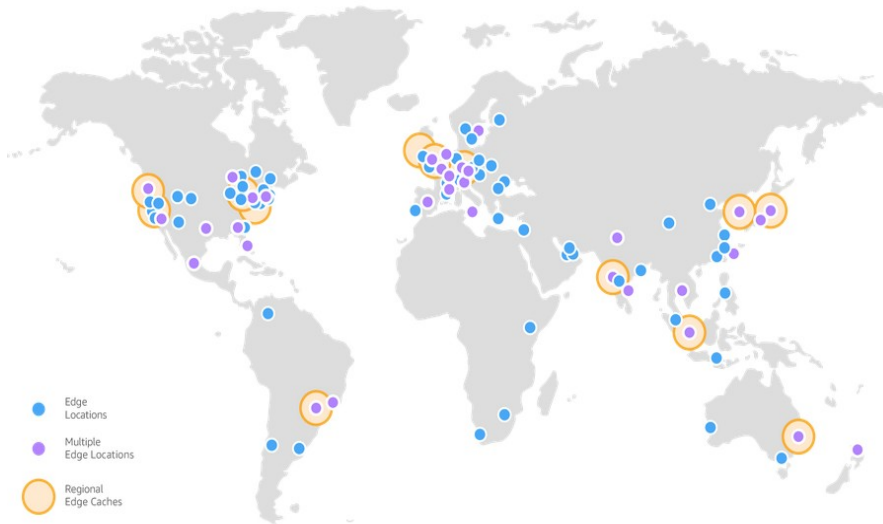
Amazon CloudFront è un servizio di **Content Delivery Network (CDN)** che si basa su un'infrastruttura di storage distribuita gestita da Amazon.

- Fornisce contenuti statici, dinamici e in streaming utilizzando una rete globale di server di replica edge.
- Le richieste vengono instradate alla replica edge più vicina
- **Oggetti organizzati in distribuzioni**
 - Una distribuzione identifica il server di origine (bucket S3, istanza Amazon EC2, un file o un server esterno)
 - Una distribuzione ha un nome di dominio CloudFront.net univoco (ad es. mydist.cloudfront.net)



Amazon CloudFront: Global Edge Network

- **Amazon CloudFront** viene utilizzato per distribuire in modo sicuro contenuti statici (ad es., file, video) con bassa latenza e velocità di trasferimento elevate.
- CloudFront si avvale di una rete globale che comprende oltre 600 punti di presenza (**Point of Presence** o **Edge Location**) e 13 **Cache Edge** regionali in oltre 100 città in 50 paesi.



Amazon CloudFront: Global Edge Network

- **Lambda@Edge** in Amazon CloudFront consente di eseguire il codice più vicino agli utenti, migliorando le prestazioni delle applicazioni e riducendo la latenza.
- Elimina la necessità di gestire l'infrastruttura a livello globale (è un servizio serverless). Richiede il pagamento solo per il tempo di elaborazione utilizzato ed elimina i costi di amministrazione del server per le applicazioni web distribuite a livello globale.



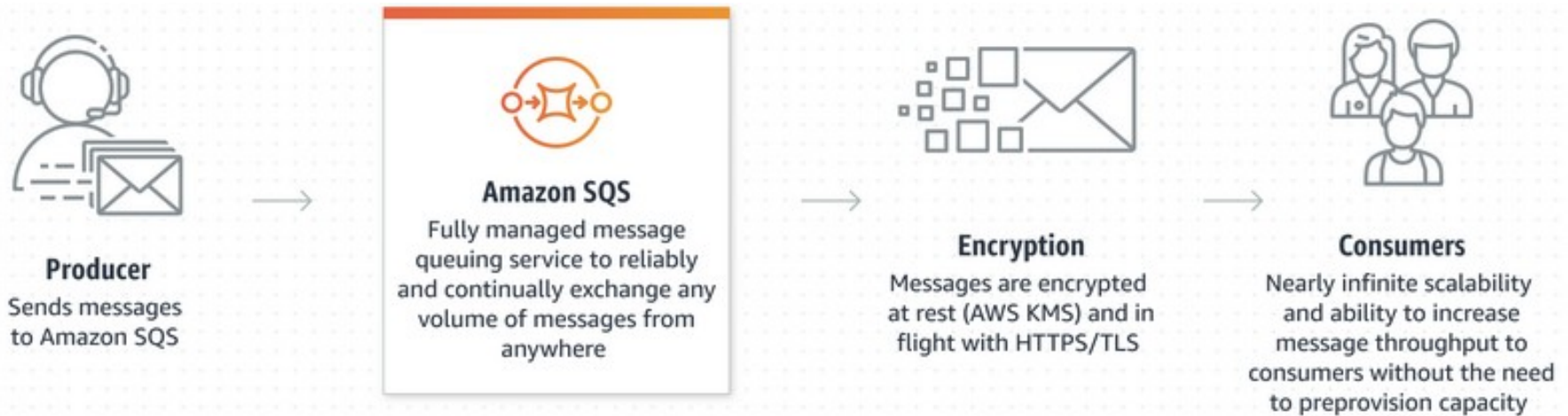
Amazon CloudFront



Simple Queue Service (SQS)

Amazon CloudFront è un servizio di accodamento messaggi che abilita comunicazione asincrona basata su messaggi tra componenti distribuiti di un'applicazione

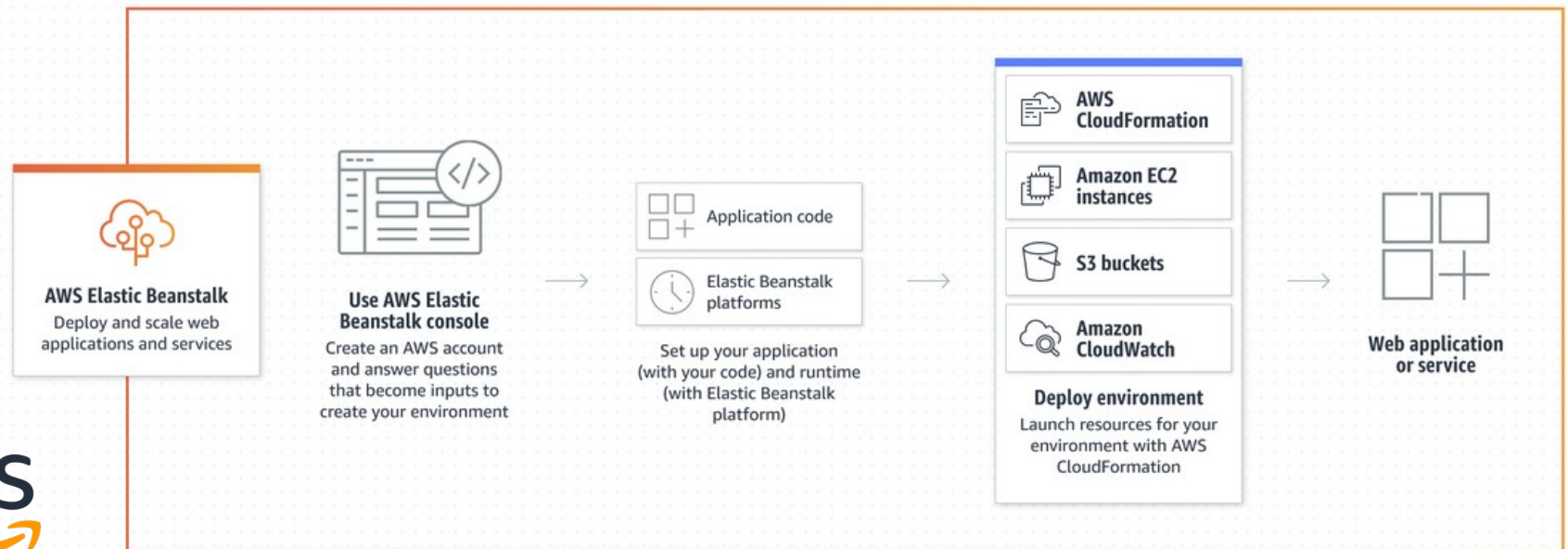
- Quando un messaggio viene ricevuto, questo viene "bloccato" finché non è in lavorazione.
- Se l'elaborazione del messaggio fallisce, il blocco scadrà e il file messaggio sarà nuovamente disponibile
- Può essere usato per disaccoppiare il frontend dal backend. Ad esempio in un'applicazione web, i clienti sottomettono un task e ricevono immediatamente una risposta di avvenuta sottomissione, ma il task viene elaborato in background non appena un consumer sarà disponibile.



AWS Elastic Beanstalk

Consente di distribuire e gestire le applicazioni web combinando i servizi AWS (EC2, S3, ELB, Auto Scaling, ElastiCache, SQS) ma senza la necessità di gestire i relativi servizi singolarmente.

- Supporta più ambienti applicativi: Docker, Python, PHP, Node.js, .NET, Ruby e Java
- Consente di creare e gestire in modo semplice e rapido le applicazioni tramite la console
- Quando lo sviluppatore crea l'applicazione ambiente, Elastic Beanstalk le risorse necessarie per eseguirlo
- Consente allo sviluppatore di personalizzare e configurare l'ambiente (es. JVM, variabili ambiente)
- Monitoraggio dei costi e delle risorse tramite AWS CloudWatch
- Prezzi: pur essendo una sorta di PaaS, nessun costo aggiuntivo viene pagato per Elastic Beanstalk, l'utente paga solo per le risorse AWS sottostanti che l'applicazione consuma



Amazon AWS Lambda

- AWS Lambda è un servizio di calcolo basato su eventi **serverless** che ti permette di eseguire codici per qualsiasi tipo di applicazione o servizio back-end senza effettuare il provisioning o gestire server.
- E' possibile usare lambda da oltre 200 servizi AWS ed applicazioni di software come servizio (SaaS), pagando solo per il servizio utilizzato.
- Il costo dipende dalla quantità e dalla durata della memoria allocata per la funzione, nonché dal numero di richieste (0,20\$ per milione di richieste).
- AWS Lambda può essere collegato allo storage di AWS (S3 o EFS) per processare un bulk file caricati (es. video o immagini), applicando su di essi una particolare funzione.



Amazon AWS Lambda - Esempio (step 1)

- Costruiamo una semplice Lambda function per il calcolo della radice quadrata di un numero passato in input.
- 1) Andare dalla Console sul servizio AWS Lambda → Crea funzione → Crea da zero
 - 2) Configurare il nome della funzione e scegliere un runtime

The screenshot shows the 'Create Function' wizard in the AWS Lambda console. At the top, there are three tabs: 'Crea da zero' (selected), 'Usa un piano', and 'Immagine del container'. Below the tabs, the 'Informazioni di base' section is visible. It contains three fields: 'Nome funzione' with the value 'CalcoloSQRT', 'Runtime' with the value 'Node.js 14.x', and 'Architettura' with the value 'x86_64'. The 'Autorizzazioni' section is partially visible at the bottom.

Crea da zero ☒ Inizia con un semplice esempio di Hello World.

Usa un piano ☐ Crea un'applicazione Lambda dal codice di esempio e dai set di impostazioni di configurazione per i casi di utilizzo comuni.

Immagine del container ☐ Seleziona un'immagine del container per la funzione.

Informazioni di base

Nome funzione
Inserisci un nome che descriva lo scopo della funzione.

Utilizza solo lettere, numeri, trattini o caratteri di sottolineatura senza spazi.

Runtime [Info](#)
Scegli il linguaggio da utilizzare per scrivere la funzione. L'editor di codice della console supporta solo Node.js, Python e Ruby.


Architettura [Info](#)
Scegli l'architettura del set di istruzioni desiderata per il codice della funzione.
☒ x86_64
☐ arm64

Autorizzazioni [Info](#)
Come impostazione predefinita, Lambda creerà un ruolo di esecuzione con le autorizzazioni a caricare i log in Amazon CloudWatch Logs. Potrai personalizzare questo ruolo predefinito ulteriormente durante la configurazione.



Amazon AWS Lambda - Esempio (step 2)

- E' possibile configurare un trigger per azionare la funzione e una destinazione verso la quale indirizzare la risposta (es. per salvare il risultato o azionare lo step successivo di un workflow).



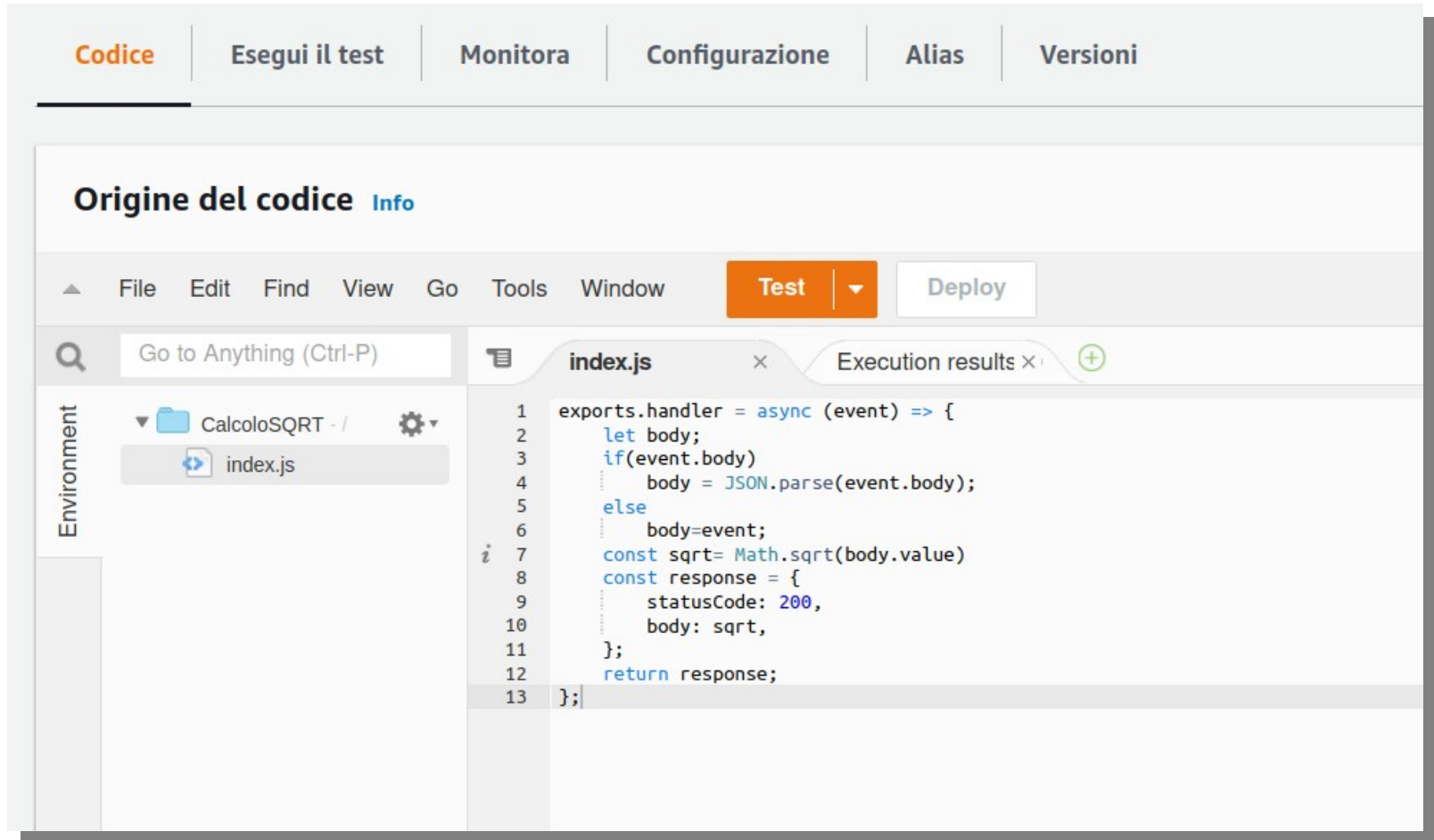
The screenshot displays the AWS Lambda console interface for a function named "CalcoloSQRT". At the top, there are three buttons: "Limitatore", "Copia ARN", and "Operazioni". Below the function name, there is a section titled "Panoramica della funzione" with an "Info" link. The main area shows the function icon, the name "CalcoloSQRT", and a "Layers" section with "(0)" layers. There are two buttons: "+ Aggiungi trigger" on the left and "+ Aggiungi destinazione" on the right. On the right side, there is a details panel with the following information:

- Descrizione: -
- Ultima modifica: 2 minuti fa
- ARN della funzione: `arn:aws:lambda:us-east-1:068305421964:function:CalcoloSQRT`
- URL della funzione: [Info](#)



Amazon AWS Lambda - Esempio (step 3)

- Nella sezione "Codice" è possibile inserire il codice della funzione, ma può essere anche passato un archivio (jar/zip) o caricato da S3. Una volta creata/modificata la funzione, fare clic su Deploy.



Amazon AWS Lambda - Esempio (step 4)

- Clic su "Test" per configurare un evento per verificare il corretto funzionamento della funzione. Nel body dell'evento inseriamo il valore di cui calcolare la radice quadrata.

Configura evento di test

Un evento di test è un oggetto JSON che simula la struttura delle richieste emesse dai servizi AWS per invocare una funzione Lambda. Utilizzalo per visualizzare il risultato dell'invocazione della funzione.

Per invocare la funzione senza salvare un evento, modificare l'evento, quindi scegli Test. Lambda utilizza l'evento modificato per invocare la funzione, ma non sovrascrive l'evento originale finché non scegli Salva modifiche.

Operazione evento di test

☐ Crea nuovo evento

☒ Modifica evento salvato

Nome evento

TestSqrt

↺

↻

Elimina

JSON dell'evento

Formato JSON

```
1 {  
2   "value": 4  
3 }
```



Amazon AWS Lambda - Esempio (step 5)

- Fare nuovamente clic su Test per inviare l'evento di test alla funzione.
- Dalla risposta si può estrarre il risultato della funzione dal body, ma anche verificare i tempi di completamento e la memoria usata.

The screenshot displays the AWS Lambda console interface. At the top, there are tabs for 'Codice', 'Esegui il test', 'Monitora', 'Configurazione', 'Alias', and 'Versioni'. The 'Esegui il test' tab is active. Below the tabs, there's a section titled 'Origine del codice' with an 'Info' link and a 'Carica da' button. The main area shows the 'Test' results for the function 'CalcoloSqrt'. The 'Test Event Name' is 'TestSqrt'. The 'Response' is a JSON object:

```
{  "statusCode": 200,  "body": "2"}
```

. The 'Function Logs' section shows the execution details:

```
START RequestId: 2f9ffaac-601d-4405-9ef9-fc674d13039b Version: $LATEST
END RequestId: 2f9ffaac-601d-4405-9ef9-fc674d13039b
REPORT RequestId: 2f9ffaac-601d-4405-9ef9-fc674d13039b  Duration: 77.85 ms  Billed Duration: 78 ms  Memory Size: 128 MB Max Memory Used: 56 MB
```


. The 'Request ID' is '2f9ffaac-601d-4405-9ef9-fc674d13039b'. The status is 'Succeeded', max memory used is '56 MB', and time is '77.85 ms'.



Amazon AWS Lambda - Esempio (step 6)


- Dalla sezione Configurazione è possibile abilitare un URL per la funzione, così da renderla accessibile come un'API REST.


URL della funzione [Info](#)
Utilizza gli URL delle funzioni per assegnare endpoint HTTP (S) alla funzione Lambda.


Tipo di autenticazione
Scegli il tipo di autorizzazione per l'URL della funzione. [Ulteriori informazioni](#) 

☐ **AWS_IAM**
Solo gli utenti e i ruoli IAM autenticati possono effettuare richieste all'URL della funzione.

☒ **NONE**
Lambda non eseguirà l'autenticazione IAM sulle richieste all'URL della funzione. L'endpoint dell'URL sarà pubblico a meno che non si implementi la propria logica di autorizzazione nella funzione.

 Esiste una policy che concede l'accesso pubblico all'URL della funzione. Se scegli il tipo di autorizzazione NONE, chiunque con l'URL può accedere alla funzione.

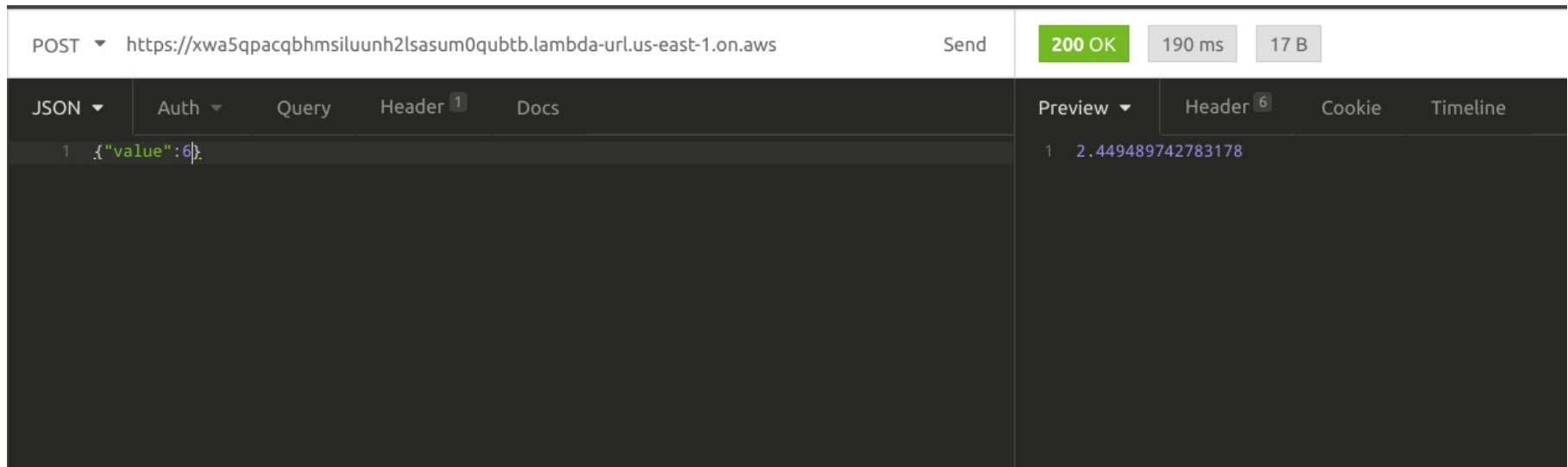
☐ **Configure Cross-origin Resource Sharing (CORS)**
Utilizza CORS per consentire l'accesso all'URL della funzione da qualsiasi dominio. Puoi utilizzare CORS anche per controllare l'accesso per intestazioni e metodi HTTP specifici nelle richieste all'URL della funzione. [Ulteriori informazioni](#) 



[Annulla](#) [Salva](#)

Amazon AWS Lambda - Esempio (step 6)

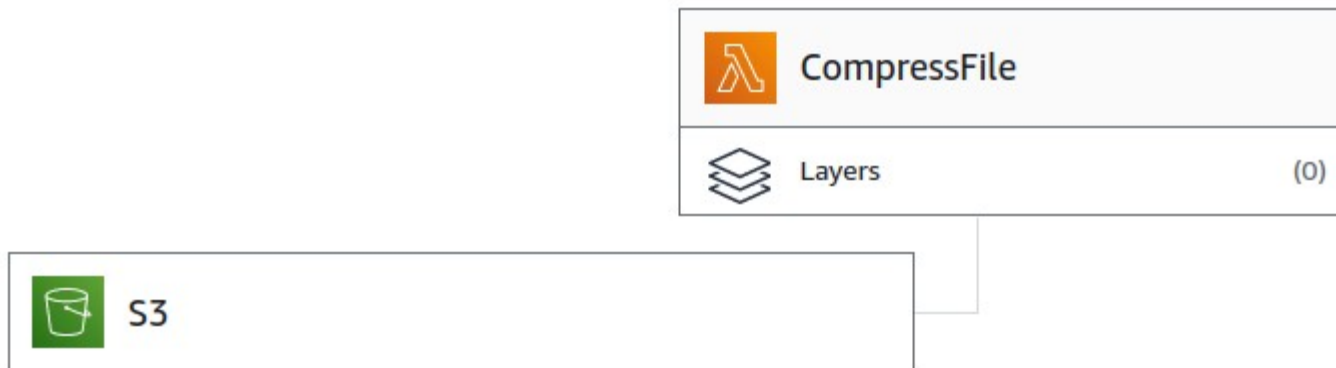
- L'API appena creata può essere invocata mediante una POST HTTP.
- In questo esempio, l'API è pubblica (non necessita di autenticazione), quindi possiamo invocarla liberamente senza credenziali di accesso alla funzione.



Amazon AWS Lambda - Esempio 2

- Vogliamo realizzare una funzione Lambda function per la compressione di file.
- Ogni volta che viene caricato un file su un bucket S3, questo file viene compresso e copiato su un'altra cartella di S3.
- Per realizzare l'applicazione useremo Python e relative dipendenze.
- Per interagire con i servizi AWS da Python, useremo **Boto3** (AWS SDK for Python)

<https://boto3.amazonaws.com>



Amazon AWS Lambda - Esempio 2 (Step 1)

- Iniziamo a creare una funzione Lambda, selezionando Python 3.11 come Runtime.

Informazioni di base

Nome funzione

Inserisci un nome che descriva lo scopo della funzione.

CompressS3

Utilizza solo lettere, numeri, trattini o caratteri di sottolineatura senza spazi.

Runtime [Info](#)

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.11

Architettura [Info](#)

Scegli l'architettura del set di istruzioni desiderata per il codice della funzione.

☒ x86_64

☐ arm64




Amazon AWS Lambda - Esempio 2 (Step 2)


- Nella sezione Autorizzazioni, dobbiamo creare un apposito ruolo per consentire alla funzione Lambda di accedere in sola lettura ad S3.
- Fatto questo possiamo salvare la nostra funzione Lambda.

▼ Modifica del ruolo di esecuzione predefinito

Ruolo di esecuzione

Scegli un ruolo che definisce le autorizzazioni della funzione. Per creare un ruolo personalizzato, vai alla [console IAM](#) .

- ☐ Crea un nuovo ruolo con le autorizzazioni Lambda di base
- ☐ Utilizza un ruolo esistente
- ☒ Crea un nuovo ruolo dalla policy AWS templates

 La creazione del ruolo potrebbe richiedere alcuni minuti. Non eliminare il ruolo né modificare le policy di attendibilità o autorizzazione in questo ruolo.

Nome ruolo


Immetti un nome per il nuovo ruolo.

CompressS3File

Utilizza solo lettere, numeri, trattini o caratteri di sottolineatura senza spazi.

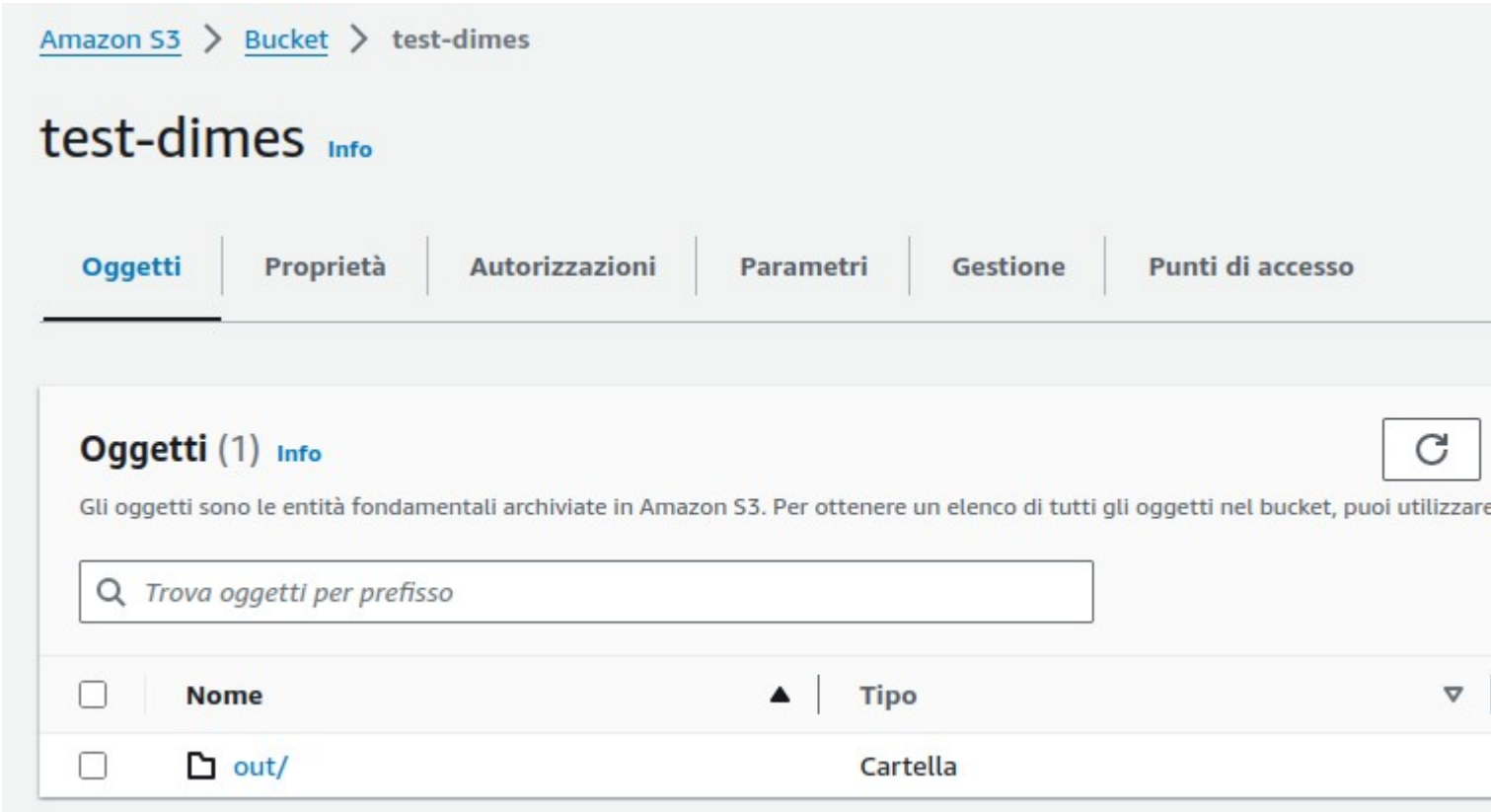
Modelli di policy - *facoltativo* [Info](#)

Seleziona uno o più modelli di policy.

Autorizzazioni di sola lettura su oggetti di Amazon S3 
S3

Amazon AWS Lambda - Esempio 2 (Step 3)

- Ora creiamo un apposito bucket S3 per il caricamento dei file da comprimere. Dentro il bucket creiamo, per comodità una cartella *out*, che useremo per i memorizzare i file compressi.
- **Attenzione:** S3 bucket e funzione Lambda devono stare sulla stessa AZ, altrimenti si avranno dei problemi di visibilità.



The screenshot shows the Amazon S3 console interface for a bucket named 'test-dimes'. The breadcrumb navigation at the top reads 'Amazon S3 > Bucket > test-dimes'. Below the bucket name, there are tabs for 'Oggetti', 'Proprietà', 'Autorizzazioni', 'Parametri', 'Gestione', and 'Punti di accesso'. The 'Oggetti' tab is selected, showing a list of objects. A search bar is present with the placeholder text 'Trova oggetti per prefisso'. Below the search bar, there is a table with columns for selection (checkbox), name, and type. One object is listed: a folder named 'out/' with the type 'Cartella'.

	Nome	Tipo
<input type="checkbox"/>	out/	Cartella



Amazon AWS Lambda - Esempio 2 (Step 4)

- Ora dobbiamo dare il permessi di scrittura sul bucket S3 alla funzione Lambda.
- Andare su **IAM** > **Ruoli** e selezionare il ruolo creato per l'applicazione Lambda (*CompressFileS3*).
- Clic su **“Crea policy inline”** → **JSON** ed incollare la seguente configurazione (è comunque possibile seguire un wizard automatico)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::test-dimes/*"
    }
  ]
}
```



Amazon AWS Lambda - Esempio 2 (Step 5)

- Ritorniamo su Lambda, apriamo la funzione *CompressFile* creata e aggiungiamo il seguente codice al file *lambda_function.py*:
- Successivamente clic su **File→Salva e Deploy**.

```
import json
import io
import os
import boto3
import tarfile
```

```
s3 = boto3.client('s3')
```

```
def lambda_handler(event, context):
```

```
    #Get bucket and file name
```

```
    bucket_name = event['Records'][0]['s3']['bucket']['name']
```

```
    object_key = event['Records'][0]['s3']['object']['key']
```

```
    if not object_key.startswith("out/"):
```

```
        tmp_file_path = '/tmp/{}'.format(object_key)
```

```
        s3.download_file(bucket_name, object_key, tmp_file_path)
```

```
        tar_filename = '{}.tar.gz'.format(object_key)
```

```
        tar_path = '/tmp/{}'.format(tar_filename)
```

```
        with tarfile.open(tar_path, "w:gz") as tar:
```

```
            tar.add(tmp_file_path, arcname=os.path.basename(tmp_file_path))
```

```
        tar.close()
```

```
        s3.upload_file(tar_path, bucket_name, 'out/{}'.format(tar_filename))
```

Amazon AWS Lambda - Esempio 2 (Step 6)

- Aggiungiamo un trigger associato da S3 per l'evento di creazione di un oggetto.

Configurazione del trigger



S3

aws

asynchronous

storage



Bucket

Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.



arn:aws:s3:::test-dimes



Bucket must be in region us-east-1

Event types

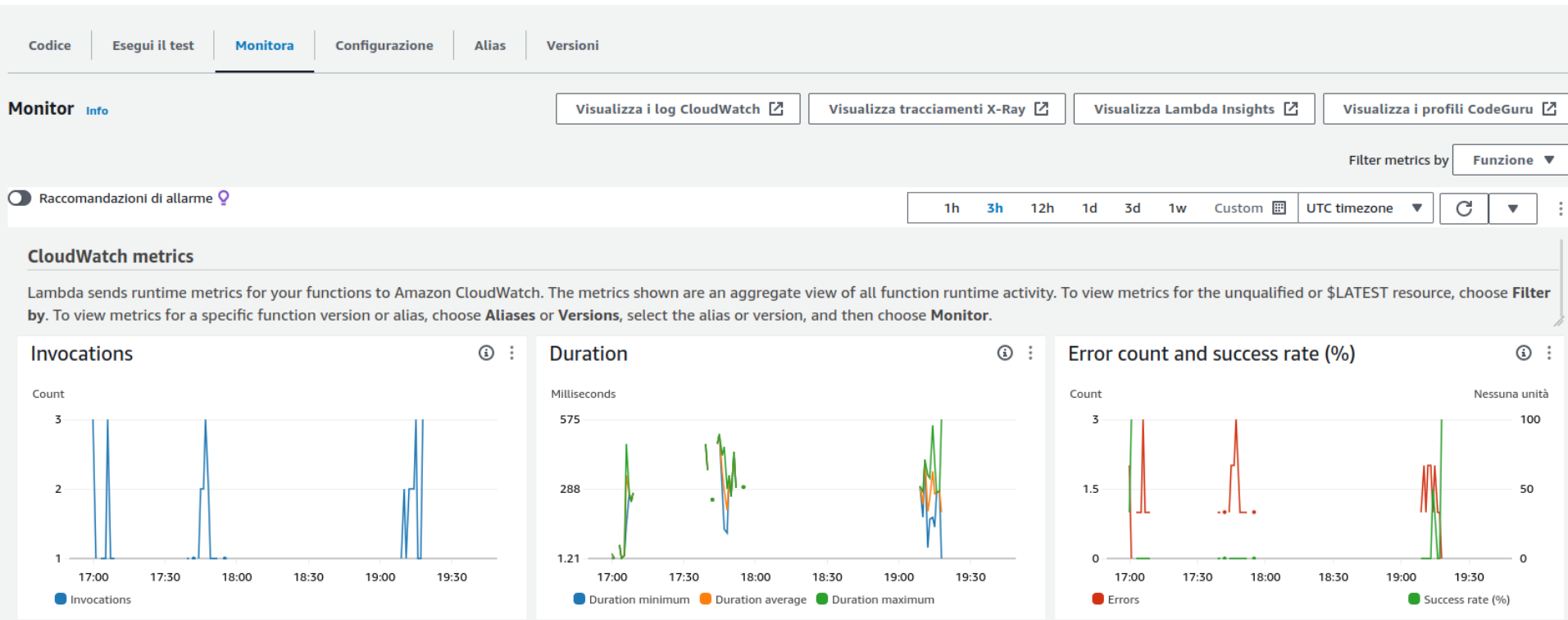
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events ✕



Amazon AWS Lambda - Esempio 2 (Step 7)

- Carichiamo un file su S3 nel bucket *test-dimes* e verifichiamo nella sezione **Monitora** di Lambda (**CloudWatch**) se l'esecuzione è andata a buon fine.



Amazon AWS Lambda - Esempio 2 (Step 7)

- Cliccando su “Visualizza i log CloudWatch” è possibile accedere al log dell’esecuzione, in cui sono riportate anche le risorse usate (e fatturate) per l’esecuzione.

CloudWatch > Gruppi di log > /aws/lambda/CompressFile > 2024/04/16/[\$LATEST]bdad9c05d2de4d26a1f8abbe74d75915

Eventi di log

Puoi utilizzare la barra dei filtri qui sotto per cercare e trovare corrispondenze con termini, frasi o valori negli eventi di log. [Ulteriori informazioni sui pattern di filtro](#)

🔍 Filtra eventi: premi invio per cercare

Annulla

1minuto

30minuto

1ora

12ora

Personalizzato

Fuso orario locale

Display



Timestamp

Messaggio

Nessun evento meno recente al momento. [Riprova](#)

2024-04-16T21:18:12.849+02:00	INIT_START Runtime Version: python:3.10.v32 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:95a2fd9a7518cc85d13fc08854b7e7167a61a5ee15ed...
2024-04-16T21:18:13.305+02:00	START RequestId: 247d0ca9-8df2-4d83-a32b-cc8238568c3c Version: \$LATEST
2024-04-16T21:18:13.307+02:00	END RequestId: 247d0ca9-8df2-4d83-a32b-cc8238568c3c
2024-04-16T21:18:13.307+02:00	REPORT RequestId: 247d0ca9-8df2-4d83-a32b-cc8238568c3c Duration: 2.48 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 76 MB Init ...



Amazon AWS Lambda - Esempio 2 (Step 7)





- Dentro S3 → Bucket *test-dimes/out* sono presenti i file compressi.

Amazon S3 > Bucket > test-dimes

test-dimes Info

Oggetti | Proprietà | Autorizzazioni | Parametri | Gestisci

Oggetti (2) Info

  Copia URI S3  Copia URL  Scarica

Gli oggetti sono le entità fondamentali archiviate in Amazon S3. Per ottenere un elenco oggetti, è necessario concedere loro le autorizzazioni esplicitamente. [Ulteriori informazioni](#)

<input type="checkbox"/>	Nome	Tipo
<input type="checkbox"/>	out/	Cartella
<input type="checkbox"/>	restaurants.json	json







Amazon S3 > Bucket > test-dimes > out/

out/

Oggetti | Proprietà

Oggetti (1) Info

  Copia URI S3  Copia URL  Scarica

Gli oggetti sono le entità fondamentali archiviate in Amazon S3. Per ottenere un elenco oggetti, è necessario concedere loro le autorizzazioni esplicitamente. [Ulteriori informazioni](#)

<input type="checkbox"/>	Nome	Tipo
<input type="checkbox"/>	restaurants.json.tar.gz	gz