



Skarab Group

Analisi AWS

skarabswegroup@gmail.com

Versionamento e changelog

Data Modifica	Versione	Descrizione Modifica	Redattore	Verificatore
2025-11-21	1.0.1	Fix formato data	Basso Kevin	/
2025-11-20	1.0.0	Creazione documento	Basso Kevin	/

Indice

Cos'è AWS	4
Profili IAM	4
Gestione di un profilo	4
Servizi offerti utili al progetto Code Guardian	5
Storage	5
Amazon S3	5
Amazon DynamoDB	5
Amazon RDS	5
Execution	5
AWS Lambda	5
AWS Fargate / ECS	5
Amazon SQS (Simple Queue Service)	5
Monitoraggio e audit	5
CloudWatch	5
CloudTrail	5
Bedrock	5
BedRock AgentCore	5
Bedrock AgentCore VS AWS Lambda	6
Recap	6
Fonti	6

Cos'è AWS

Amazon Web Services (**AWS**) è una piattaforma di servizi di cloud computing offerta da Amazon, che fornisce una vasta gamma di risorse scalabili e on-demand, tra cui elaborazione, archiviazione, database, reti, analisi dei dati e intelligenza artificiale. AWS permette alle organizzazioni di distribuire applicazioni e servizi senza dover gestire infrastrutture fisiche, riducendo i costi operativi e aumentando la flessibilità. Grazie al modello pay-as-you-go, le aziende possono adattare dinamicamente le risorse alle proprie esigenze, garantendo alta disponibilità, sicurezza e scalabilità a livello globale.

Profili IAM

In AWS, **IAM** (Identity and Access Management) è il servizio che permette di gestire utenti, gruppi e permessi per controllare l'accesso alle risorse. Un profilo **IAM** è un'entità associata a un ruolo o a un utente che definisce quali azioni possono essere eseguite su quali risorse. I profili **IAM** consentono di applicare il principio del least privilege, garantendo che ogni utente o servizio abbia solo i permessi strettamente necessari per svolgere le proprie funzioni, aumentando la sicurezza e il controllo dell'ambiente cloud. Un profilo **IAM** collega un ruolo AWS alle varie risorse, dove un ruolo è un'entità di AWS che definisce insieme di permessi da applicare a utenti, servizi o applicazioni.

Gestione di un profilo

I vari profili **IAM** vengono creati tramite la console AWS. Una volta creato il profilo, le credenziali vanno inserite in file di configurazione locali come `~/.aws/credentials` (AWS **CLI** (Command Line Interface)) o veriabili d'ambiente. Ovviamente, essendo credenziali, i file di configurazione vanno inseriti nel `.gitignore` nel caso di un progetto GitHub come Code Guardian. L'utilizzo delle credenziali può essere monitorato tramite **AWS CloudTrail**.

Servizi offerti utili al progetto Code Guardian

Storage

Amazon S3

Utile per conservare i repository già scaricati, log delle valutazioni e report dei risultati, inoltre permette una scalabilità e durata praticamente illimitata.

Amazon DynamoDB

Database NoSQL ottimo per salvare i risultati di valutazioni, metriche e metadati

Amazon RDS

Database SQL, utile per query complesse sui risultati

Execution

AWS Lambda

Servizio di serverless computing, utile per evitare di gestire server fissi come con EC2, scalabilità automatica

AWS Fargate / ECS

Gestore di containers docker serverless, utile per fornire ai vari agenti ambienti isolati complessi

Amazon SQS (Simple Queue Service)

Servizio che permette la gestione FIFO di messaggi e task tra servizi o istanze diverse

Monitoraggio e audit

CloudWatch

Servizio per il monitoraggio delle risorse come memoria CPU, log etc.. e applicazioni di un account AWS, utile per comprendere le performances e ottimizzare le risorse

CloudTrail

Servizio di audit e logging di un account AWS

Bedrock

Servizio indispensabile per l'utilizzo di AI generativa tramite AWS, offre un grande numero di modelli, come default usa Claude (ultima versione). Utilizzabile tramite AWS Strand, una libreria open source creata da Amazon per la creazione di agenti AI.

BedRock AgentCore

Ambiente di serverless computing studiato appositamente per agenti AI, funziona a virtual machines con sessioni isolate fino a 8 ore; gestisce sia la short-time memory dell'agente durante la sessione che una long-time-memory come le preferenze utente, inoltre permette di gestire le funzioni tramite server MCP

Bedrock AgentCore VS AWS Lambda

Funzione	AgentCore	Lambda
Scopo	Gestire agenti AI, stati, memoria e workflow	Eseguire codice in risposta ad eventi come S3 e DynamoDB in modo scalabile
Durata esecuzione	Fino ad 8 ore	Fino a 15 minuti
Stato	mantiene il contesto per tutta la sessione	Stateless, ogni invocazione è indipendente

Recap

- Pipeline serverless automaticamente scalabile tramite Lambda, BedRock AgentCore, S3, DynamoDB e SQS
- Distinzioni tra ambienti diversi con Fargate e ECS
- Monitoraggio e audit tramite CloudTrail

Fonti

Documentazione ufficiale AWS

Amazon Bedrock AgentCore

AWS SQS

Lista di servizi AWS

AgentCore