

**Containerizzazione e deployment su Cloud
di un simulatore di Ethereum con Docker e AWS**

Antonio Gravino - 05225 01502
Sicurezza dei Dati (a.a. 2022-2023)

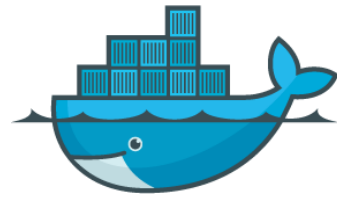
Il problema

- **Lo sviluppo di applicativi e blockchain Ethereum-based è complesso**
- Si tratta di un'**attività di gruppo**, partecipata e condivisa, che impegna in genere **team numerosi**
- Inevitabilmente, una mole di lavoro più o meno corposa può far emergere **problematiche relative ai processi di sviluppo**
- In genere, lavorare nella cornice di Ethereum implica l'impiego di development kits e **tool di simulazione**
 - Truffle, Ganache sono sicuramente fra i più noti
- Tuttavia, l'utilizzo «*in locale*» di questi tool potrebbe non essere sufficiente per numerosi motivi
- **E' necessario predisporre ed elargire la simulazione attraverso tecnologie flessibili**
in modo da rendere la fruizione della codebase e della blockchain pratica ed immediata

La soluzione

- L'obiettivo è permettere a tutti gli sviluppatori di avere un **accesso comune, potenzialmente distribuito, facilmente scalabile** e con elevati requisiti di **observability** alla **simulazione della Blockchain**, per la quale s'intende utilizza **Ganache**.
- L'ambiente simulato dev'essere **portabile**, quanto più **light-weight** possibile e **replicabile** in base alla necessità. Risulta dunque immediato l'impiego di tecnologie di containerizzazione via **Docker**.
- Per minimizzare i costi ed ottenere elevati standard di qualità, si preferisce ovviamente una **infrastruttura cloud native** piuttosto che on-premise. Si procederà con **AWS** come cloud provider.
- Vogliamo anche essere in grado di prevedere potenziali sviluppi futuri di questa architettura cloud, arrivando ad **orchestrare n simulazioni di blockchain Ethereum** distribuite orizzontalmente.

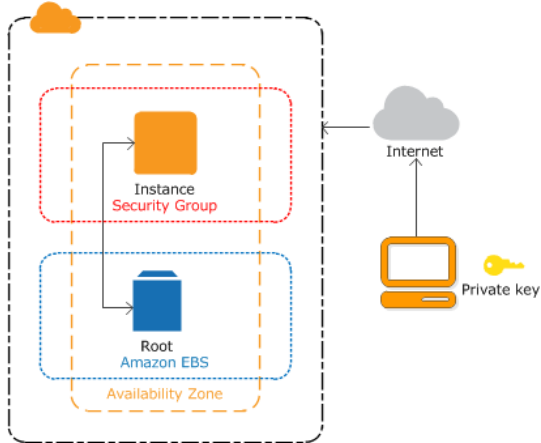
Le tecnologie



docker

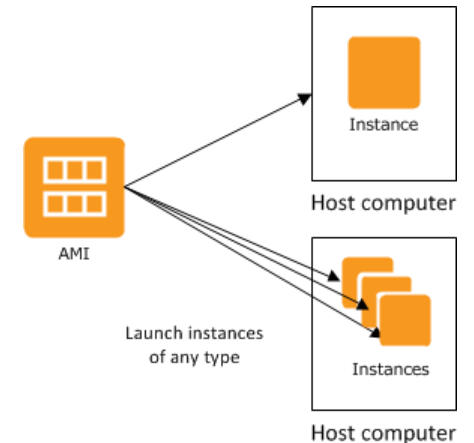


Elastic Compute Cloud e AMI

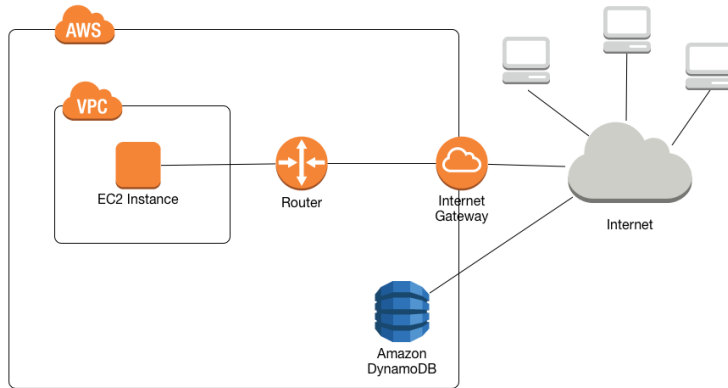


- **AWS EC2** è un servizio di cloud computing di **Amazon Web Services** che consente di utilizzare le risorse di elaborazione in modo flessibile.
- **Il nostro obiettivo è containerizzare la simulazione di un sistema distribuito:** non è un compito poco oneroso.
- Possiamo scegliere tra diverse opzioni di configurazione delle istanze, tra cui tipo di macchina virtuale, quantità di memoria, potenza di elaborazione e archiviazione.

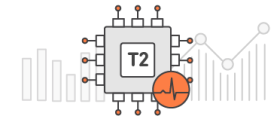
- Un **Amazon Machine Image** (AMI) è un modello preconfigurato per un'istanza EC2, che include sistema operativo, applicazioni e configurazioni.
- Gli utenti possono utilizzare un'immagine predefinita di AWS o creare una propria immagine personalizzata per utilizzare come base per le istanze EC2.
- Gli AMI consentono agli utenti di avviare rapidamente istanze con le configurazioni desiderate, senza la necessità di configurare manualmente ogni istanza a mano.



Creare l'istanza **EC2** per **Ganache**

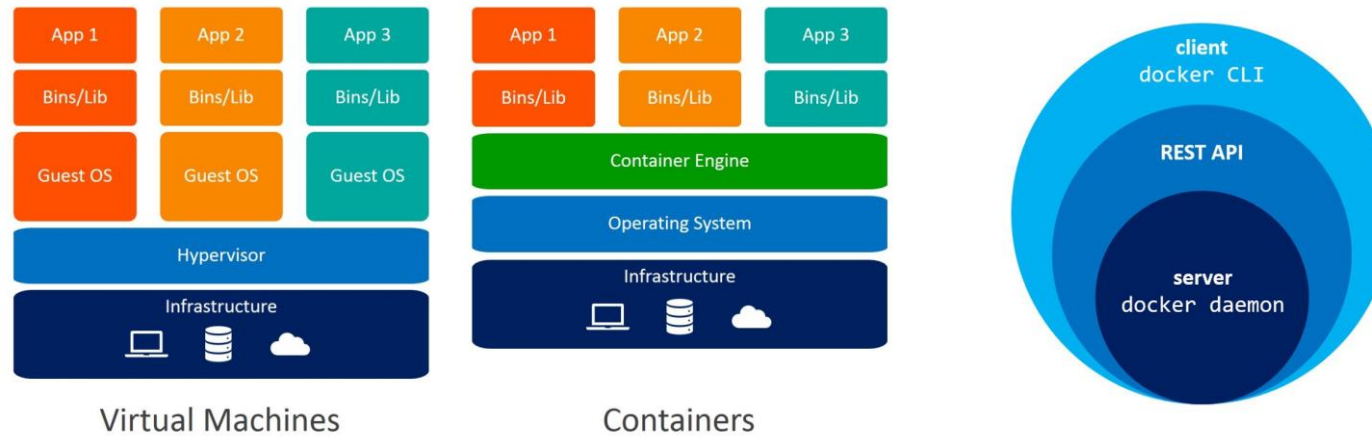


- Questo progetto è fondamentalmente un «proof of concept», quindi utilizzeremo un tier semi-gratuito di EC2: la **t2.micro**
 - Intel Xeon 3,3 GHz (vCPU) – 1GB RAM
- Per prevenire accessi da utenti non autorizzati, si realizza una **VPC** (Virtual Private Cloud) per servire richieste originate da fonti specifiche e indirizzate a porte ben definite.
- L'accesso via remote host del server è consentito esclusivamente via SSH con chiave PEM (RSA). Accedere alla macchina ci permette di configurarla secondo le nostre necessità, cioè costruire un container Docker con Ganache ed impostarlo in listening su :8545.



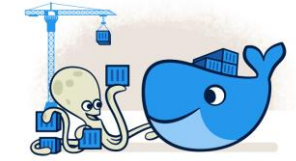
Instances (1) Info									
Find instance by attribute or tag (case-sensitive)									
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	ganache-ec2	i-0c820d0c8a02981df	Running	t2.micro	2/2 checks passed	No alarms	eu-central-1a	ec2-3-71-206-133.eu-c...	3.71.206.133

Il ruolo della containerizzazione e di Docker



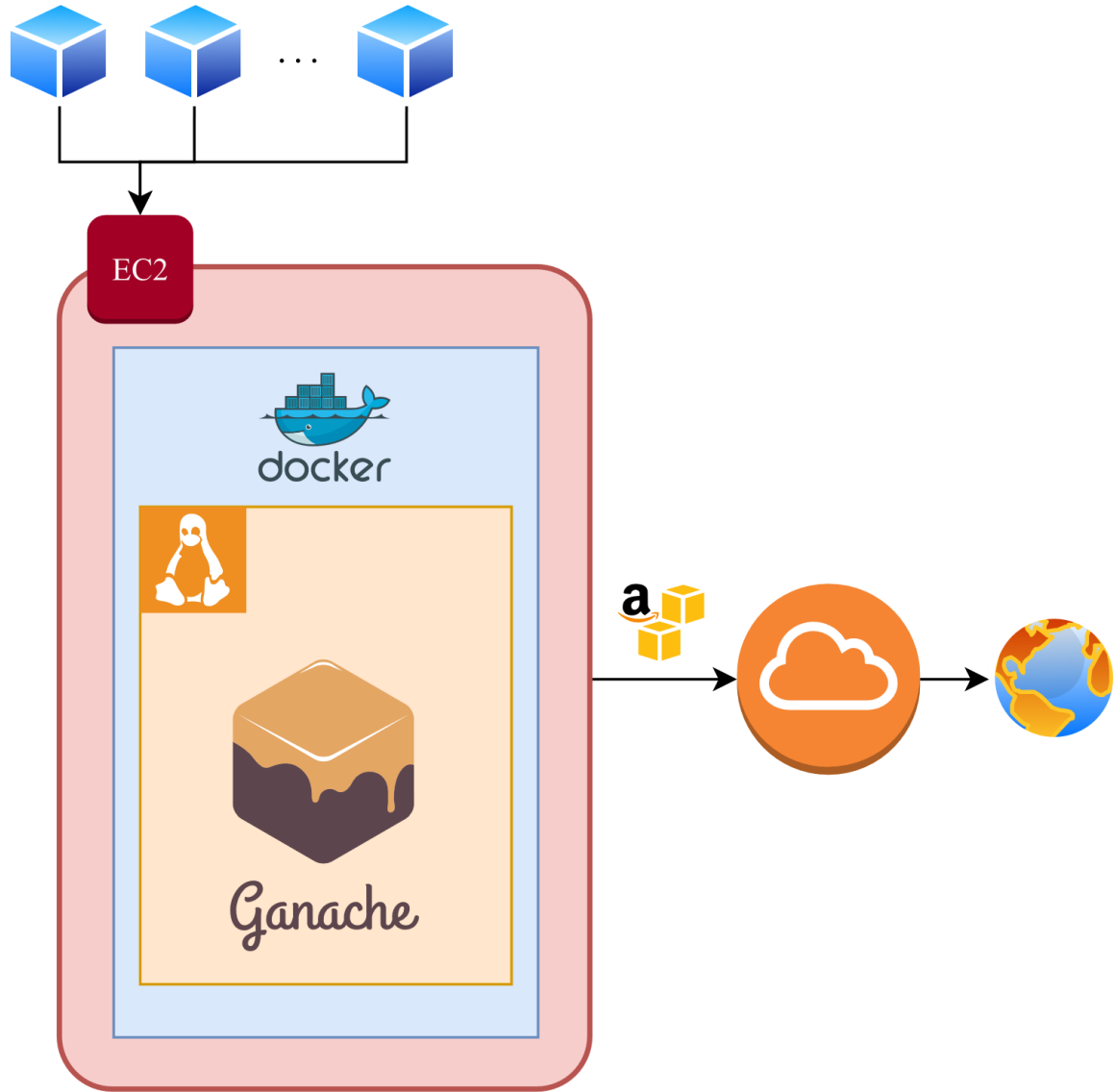
- La containerizzazione è una tecnologia che permette di **isolare le applicazioni** all'interno di containers, che condividono il kernel del sistema operativo ospite.
- Ciò consente di avere immagini più leggere rispetto alle macchine virtuali e maggiore flessibilità nella distribuzione e gestione delle applicazioni, con tempi di avvio più brevi e maggiore densità di esecuzione.
- Ci permette di realizzare un **Trusted Execution Environment**.

Docker e Ganache – I vantaggi principali



Containerizzare Ganache con Docker ha senso per diverse ragioni:

- **Isolamento:** utilizzando Docker, è possibile isolare l'ambiente di esecuzione di Ganache dal sistema operativo ospite e garantire che le impostazioni e le dipendenze siano coerenti per ogni esecuzione (es. web3, npm/nvm).
- **Portabilità e Accessibilità:** utilizzando un'immagine Docker, è possibile distribuire facilmente Ganache su diverse piattaforme, semplificando il processo di test e sviluppo su diverse configurazioni di sistema (AMI via Dockerfile).
- **Facilità di gestione:** utilizzando docker-compose o altri strumenti di gestione dei container, è possibile automatizzare il processo di avvio e arresto di Ganache, semplificando la gestione dell'ambiente di sviluppo.
- **Replicabilità:** utilizzando un'immagine Docker, è possibile replicare esattamente l'ambiente utilizzato in un determinato momento. Questo permette di riprodurre esattamente gli stessi risultati in una successiva esecuzione del progetto.



L'architettura

A breve la vedremo in funzione. In sintesi:

- Immagine **Docker** con **Ganache** e **Linux Alpine**
- Container **Docker** su **AWS EC2**
- **EC2** con **AMI Ubuntu 20.04 LTS**
- Potenziale replicabilità su n nodi
- Il container **Docker** espone **:8545** per rendere fruibile all'esterno la simulazione della blockchain via Node
- L'accesso è granulare in funzione della **VPC**

Hands-on via SSH e AWS

Possibili sviluppi futuri



Grazie per l'attenzione!

Antonio Gravino - 05225 01502

Sicurezza dei Dati (a.a. 2022-2023)