

Quiz 01 - Spiegazioni Dettagliate

1. Quale comando crea un utente Linux con home directory e shell bash?

Risposta corretta: b) `sudo useradd -m -s /bin/bash lab01_student`

`useradd` è il comando standard Linux per creare utenti. Il flag `-m` (abbreviazione di `--create-home`) crea automaticamente la home directory dell'utente in `/home/lab01_student`. Il flag `-s /bin/bash` (abbreviazione di `--shell`) imposta la shell di login a bash. Questo è il comando usato nel tutorial del Lab 01 per simulare la creazione di un utente IAM.

Perché le altre risposte sono errate:

- a) `sudo adduser -home -shell lab01_student` — `adduser` è un wrapper interattivo di alto livello (presente su Debian/Ubuntu), non un comando universale. Inoltre i flag `-home` e `-shell` non sono sintassi valida per `adduser`; il formato corretto userebbe `--home` e `--shell` con il doppio trattino.
- c) `sudo newuser --create-home lab01_student` — Il comando `newuser` non esiste nei sistemi Linux standard. Non fa parte di nessun pacchetto base delle distribuzioni.

2. In Docker Compose, quale flag rende una rete isolata da Internet?

Risposta corretta: b) `internal: true`

Nel file `docker-compose.yml` del Lab 02, la rete privata `lab02-vpc-private` è definita con `internal: true`. Questo flag impedisce ai container connessi di raggiungere reti esterne (Internet), simulando una subnet privata AWS senza route verso un Internet Gateway.

Perché le altre risposte sono errate:

- a) `isolated: true` — Non è una proprietà valida nella specifica Docker Compose per le reti. Non esiste nella documentazione ufficiale.
- c) `private: true` — Non è una proprietà valida nella specifica Docker Compose. È un nome inventato che non corrisponde a nessuna opzione reale.

3. Qual è il CIDR della subnet pubblica nel Lab 02?

Risposta corretta: c) `10.0.1.0/24`

Nel `docker-compose.yml` del Lab 02, la rete `vpc-public` è configurata con `subnet: 10.0.1.0/24` e `gateway: 10.0.1.1`. Questo CIDR fornisce 254 indirizzi IP utilizzabili (da 10.0.1.1 a 10.0.1.254).

Perché le altre risposte sono errate:

- **a) 10.0.0.0/16** — Questo è un CIDR molto più ampio (65.534 host) che rappresenterebbe l'intera VPC, non una singola subnet. Nel lab non viene usato questo range.
 - **b) 10.0.2.0/24** — Questo è il CIDR della subnet **privata** (**vpc-private**), non di quella pubblica. I due CIDR si distinguono per il terzo ottetto: **.1.** = pubblica, **.2.** = privata.
-

4. Nel formato Docker Compose, come si specificano i limiti CPU di un servizio?

Risposta corretta: b) Nella sezione `deploy.resources.limits`

Come visibile nel `docker-compose.yml` del Lab 03, i limiti di risorse sono definiti sotto `deploy:` → `resources:` → `limits:` con le chiavi `cpus` e `memory`. Questa è la sintassi ufficiale Docker Compose V2.

Perché le altre risposte sono errate:

- **a) Nella sezione `resources` direttamente sotto il servizio** — La sezione `resources` non va direttamente sotto il servizio; deve essere annidata sotto `deploy`. Senza `deploy:` come livello intermedio, Docker Compose non riconosce la configurazione.
 - **c) Nella sezione `cpu_config` del servizio** — `cpu_config` non esiste nella specifica Docker Compose. È un nome inventato.
-

5. Quale porta espone MinIO per l'API S3-compatibile?

Risposta corretta: a) 9000

Nel `docker-compose.yml` del Lab 04, MinIO espone la porta 9000 per l'API S3 ("`127.0.0.1:9000:9000`") e la porta 9001 per la console web UI. L'healthcheck verifica `http://localhost:9000/minio/health/live`, confermando che 9000 è la porta API.

Perché le altre risposte sono errate:

- **b) 9001** — Questa è la porta della **console web UI** di MinIO (interfaccia grafica di amministrazione), non dell'API S3. È configurata con `--console-address ":9001"`.
 - **c) 8080** — Non è una porta usata da MinIO. È comunemente associata a web server HTTP alternativi o proxy, ma non ha relazione con MinIO.
-

6. A quale servizio AWS corrisponde un Docker bridge network?

Risposta corretta: b) AWS VPC

Nel materiale del corso, le Docker bridge networks sono utilizzate per simulare le VPC (Virtual Private Cloud) di AWS. Una bridge network crea un segmento di rete isolato con subnet, gateway e DNS interno — esattamente come fa una VPC AWS.

Perché le altre risposte sono errate:

- **a) AWS S3** — S3 è un servizio di object storage, non un servizio di rete. Nel corso, S3 è simulato da MinIO (Lab 04).

- **c) AWS Lambda** — Lambda è un servizio di computing serverless (esecuzione di funzioni senza server). Non ha relazione con il networking.
-

7. Qual è il percorso del Docker socket su Linux?

Risposta corretta: b) `/var/run/docker.sock`

Come documentato nel file `docker-socket-permissions.md` del Lab 01, il Docker daemon espone un Unix domain socket a `/var/run/docker.sock` con permessi `srw-rw---- root docker` (660 octal). Questo è il punto di accesso per comunicare con il Docker Engine.

Perché le altre risposte sono errate:

- **a) `/etc/docker/docker.sock`** — `/etc/docker/` contiene file di configurazione del daemon (come `daemon.json`), non il socket di runtime.
 - **c) `/tmp/docker.sock`** — `/tmp/` è una directory temporanea. Il Docker socket non viene mai posizionato qui nell'installazione standard.
-

8. In Docker Compose, quale condizione di `depends_on` garantisce che un servizio sia effettivamente pronto?

Risposta corretta: b) `condition: service_healthy`

Nel `docker-compose.yml` del Lab 03, i servizi usano `depends_on` con `condition: service_healthy` per garantire che il servizio dipendente abbia superato il suo healthcheck prima di avviare il servizio che ne dipende. Questo garantisce che il servizio sia realmente operativo, non solo avviato.

Perché le altre risposte sono errate:

- **a) `condition: service_started`** — Questa condizione verifica solo che il container sia stato **avviato**, non che sia pronto. Un database potrebbe essere avviato ma ancora in fase di inizializzazione, causando errori di connessione.
 - **c) `condition: service_running`** — Non è una condizione valida in Docker Compose. Le condizioni supportate sono: `service_started`, `service_healthy` e `service_completed_successfully`.
-

9. Quale immagine base viene utilizzata per PostgreSQL nel Lab 05?

Risposta corretta: c) `postgres:16-alpine`

Nel `docker-compose.yml` del Lab 05, il servizio `db` usa `image: postgres:16-alpine`. La variante Alpine è scelta per la dimensione ridotta dell'immagine e la superficie di attacco minore.

Perché le altre risposte sono errate:

- **a) `postgres:latest`** — Il tag `latest` non è mai usato nei lab del corso. Usare `latest` è una bad practice perché non garantisce la riproducibilità dell'ambiente.

- **b) postgres:15-alpine** — La versione 15 non è quella usata nel corso. Il lab specifica esplicitamente la versione 16.

10. Quale comando verifica che un container NON sia eseguito come root?

Risposta corretta: a) `docker exec container whoami`

`docker exec` esegue un comando all'interno del container in esecuzione. `whoami` restituisce il nome dell'utente corrente. Se il risultato è diverso da "root" (ad esempio "labuser" o "postgres"), il container rispetta il requisito INF-01.

Perché le altre risposte sono errate:

- **b) `docker inspect container --format root`** — `--format root` non è una sintassi valida per `docker inspect`. Il flag `--format` richiede un template Go (es. `{{.Config.User}}`), non un valore arbitrario.
- **c) `docker logs container | grep root`** — I log del container mostrano l'output dell'applicazione, non informazioni sull'utente di esecuzione. Anche se la parola "root" apparisse nei log, non indicherebbe necessariamente che il processo gira come root.

11. Quanti indirizzi IP utilizzabili offre una subnet /24?

Risposta corretta: b) 254

Una subnet /24 ha 256 indirizzi totali (2^8). Di questi, 2 sono riservati: il primo (es. 10.0.1.0) è l'indirizzo di rete e l'ultimo (es. 10.0.1.255) è l'indirizzo di broadcast. Restano $256 - 2 = 254$ indirizzi utilizzabili.

Perché le altre risposte sono errate:

- **a) 256** — Questo è il numero totale di indirizzi, non quelli utilizzabili. Include l'indirizzo di rete e di broadcast che non possono essere assegnati agli host.
- **c) 252** — Questo valore sottrae 4 indirizzi (come fanno le subnet AWS che riservano 5 indirizzi), ma in una rete Docker standard si sottraggono solo 2 (rete e broadcast).

12. Quale variabile d'ambiente imposta il nome del database in un container PostgreSQL?

Risposta corretta: c) `POSTGRES_DB`

L'immagine ufficiale PostgreSQL Docker utilizza `POSTGRES_DB` per creare automaticamente un database al primo avvio. Nel Lab 05: `POSTGRES_DB: lab05_db`.

Perché le altre risposte sono errate:

- **a) `DB_NAME`** — Non è una variabile riconosciuta dall'immagine ufficiale PostgreSQL Docker. Potrebbe essere usata in applicazioni custom, ma non dall'immagine base.

- **b) PG_DATABASE** — Non è una variabile dell'immagine ufficiale PostgreSQL Docker. Le variabili ufficiali iniziano con **POSTGRES_**, non con **PG_**.
-

13. Qual è il formato corretto per esporre una porta solo su localhost in Docker Compose?

Risposta corretta: c) "127.0.0.1:8080:80"

Il formato **host_ip:host_port:container_port** permette di specificare l'interfaccia di bind. **127.0.0.1** limita l'accesso al solo localhost, soddisfacendo il requisito INF-02 del corso. Questo è il formato usato in tutti i lab.

Perché le altre risposte sono errate:

- **a) "8080:80"** — Senza specificare l'IP, Docker fa bind su **0.0.0.0** (tutte le interfacce), rendendo il servizio accessibile da qualsiasi indirizzo di rete. Questo viola il requisito INF-02.
 - **b) "0.0.0.0:8080:80"** — Specifica esplicitamente il bind su tutte le interfacce, rendendo il servizio accessibile dall'esterno. È l'opposto della sicurezza richiesta.
-

14. Quale valore di **cpus** in Docker Compose corrisponde a un'istanza t2.nano?

Risposta corretta: b) '0.5'

Dalla tabella di mapping EC2 nel file **ec2-instance-mapping.md** del Lab 03: t2.nano = 0.5 vCPU, 512 MB RAM. In Docker Compose si esprime come **cpus: '0.5'** e **memory: 512M**.

Perché le altre risposte sono errate:

- **a) '1'** — 1 vCPU corrisponde a un'istanza **t2.micro**, non t2.nano. Il t2.micro ha il doppio delle risorse CPU.
 - **c) '0.25'** — Non corrisponde a nessuna istanza EC2 standard nella tabella di mapping del corso. È un valore inferiore anche al t2.nano.
-

15. Dove vengono memorizzati i dati di un named volume Docker sul filesystem host?

Risposta corretta: a) /var/lib/docker/volumes/[nome-volume]/_data/

Docker memorizza tutti i named volumes sotto **/var/lib/docker/volumes/**. Ogni volume ha una sottodirectory con il suo nome, e i dati effettivi sono nella cartella **_data/**. Verificabile con **docker volume inspect**.

Perché le altre risposte sono errate:

- **b) /var/run/docker/volumes/[nome-volume]/** — **/var/run/** contiene file di runtime (come il socket), non dati persistenti dei volumi.

- **c) `/etc/docker/data/[nome-volume]/` — `/etc/docker/`** contiene la configurazione del daemon Docker, non i dati dei volumi.
-

16. Quale comando rimuove i volumi insieme ai container in Docker Compose?

Risposta corretta: **b) `docker compose down -v`**

Il flag `-v` (abbreviazione di `--volumes`) dice a `docker compose down` di rimuovere anche i named volumes definiti nel file compose, oltre a fermare e rimuovere i container e le reti.

Perché le altre risposte sono errate:

- **a) `docker compose down --remove-volumes` — `--remove-volumes`** non è un'opzione valida di `docker compose down`. Il flag corretto è `--volumes` o `-v`.
 - **c) `docker compose rm --volumes` — `docker compose rm`** rimuove solo i container fermi, non gestisce i volumi. Il flag `--volumes` non è disponibile per `rm` nel contesto dei named volumes.
-

17. Quale UID ha l'utente postgres all'interno del container PostgreSQL Alpine?

Risposta corretta: **c) 70**

Nell'immagine `postgres:16-alpine`, l'utente `postgres` è creato con UID 70. Questo è un valore specifico dell'immagine Alpine di PostgreSQL (nelle immagini Debian-based l'UID è 999). Il processo PostgreSQL gira come questo utente non-root, rispettando INF-01.

Perché le altre risposte sono errate:

- **a) 0** — UID 0 corrisponde all'utente `root`. Se PostgreSQL girasse come UID 0, violerebbe il requisito INF-01 (non-root execution).
 - **b) 1000** — UID 1000 è quello usato per l'utente `labuser` nel Lab 01 (creato con `adduser -D -u 1000`). L'utente postgres nell'immagine Alpine ha un UID diverso.
-

18. A quale servizio AWS corrisponde un Docker named volume?

Risposta corretta: **b) AWS EBS**

Nel materiale del corso, i Docker named volumes sono il parallelo di AWS EBS (Elastic Block Store). Entrambi forniscono storage persistente a livello di blocco che sopravvive al ciclo di vita del container/istanza.

Perché le altre risposte sono errate:

- **a) AWS S3** — S3 è object storage (accesso tramite API HTTP), non block storage. Nel corso, S3 è simulato da MinIO, non dai volumi Docker.
 - **c) AWS EFS** — EFS (Elastic File System) è un file system condiviso tra più istanze. Non è il parallelo diretto di un named volume locale, che è legato a un singolo host.
-

19. Qual è il gateway della subnet privata nel Lab 02?

Risposta corretta: b) **10.0.2.1**

Nel `docker-compose.yml` del Lab 02, la rete `vpc-private` è configurata con `gateway: 10.0.2.1` nella subnet `10.0.2.0/24`.

Perché le altre risposte sono errate:

- a) **10.0.1.1** — Questo è il gateway della subnet **pubblica** (`vpc-public`), non di quella privata.
 - c) **10.0.0.1** — Questo indirizzo non è usato in nessuna configurazione del corso. Apparterrebbe a una subnet `10.0.0.0/24` che non esiste nei lab.
-

20. In un healthcheck Docker, quale parametro definisce il periodo di grazia iniziale prima di contare i tentativi falliti?

Risposta corretta: c) **start_period**

`start_period` è il parametro che definisce il periodo di grazia durante il quale i fallimenti dell'healthcheck non vengono contati verso il limite di `retries`. È utile per servizi che richiedono tempo per l'inizializzazione (es. database). Nel Lab 01: `start_period: 5s`.

Perché le altre risposte sono errate:

- a) **grace_period** — Non è un parametro valido dell'healthcheck Docker. Non esiste nella specifica.
- b) **initial_delay** — Non è un parametro Docker. È un nome che ricorda la configurazione di Kubernetes (`initialDelaySeconds`), ma non è usato in Docker Compose.