

Quiz 03 - Spiegazioni Dettagliate

1. Quale permesso octal ha il Docker socket `/var/run/docker.sock`?

Risposta corretta: c) 660

Dal file `docker-socket-permissions.md` del Lab 01: il socket ha permessi `srw-rw----`, che in notazione octal corrisponde a 660 (owner: rw = 6, group: rw = 6, others: --- = 0). Owner `root` e gruppo `docker` possono leggere e scrivere; nessun altro ha accesso.

Perché le altre risposte sono errate:

- a) 755 — In octal, 755 = `rwxr-xr-x`, che darebbe permessi di lettura e esecuzione a tutti gli utenti del sistema. Sarebbe un grave problema di sicurezza: chiunque potrebbe controllare il Docker daemon.
- b) 644 — In octal, 644 = `rw-r--r--`, che darebbe permessi di sola lettura a tutti. Inoltre, il gruppo avrebbe solo lettura (non scrittura), impedendo ai membri del gruppo `docker` di eseguire comandi.

2. In Docker Compose V2, qual è il comando corretto per avviare i servizi?

Risposta corretta: b) `docker compose up -d`

Docker Compose V2 è integrato come plugin del comando `docker` (senza trattino). Il corso richiede Compose V2 e usa la sintassi `docker compose` (spazio, non trattino) in tutti gli esempi e script.

Perché le altre risposte sono errate:

- a) `docker-compose up -d` — Questa è la sintassi di Docker Compose V1, un binario Python standalone separato. V1 è deprecato dal giugno 2023 e non è supportato nel corso.
- c) `docker run compose up -d` — `docker run` serve per eseguire container da un'immagine, non per gestire servizi Compose. `compose` non è un argomento valido di `docker run`.

3. Quale formula calcola il tempo massimo per marcare un container come unhealthy?

Risposta corretta: b) $(interval + timeout) \times retries$

Ogni ciclo di healthcheck dura al massimo `interval + timeout` (il tempo tra i check più il tempo massimo per un singolo check). Dopo `retries` cicli falliti consecutivi, il container diventa unhealthy. Esempio dal corso: $(10s + 5s) \times 3 = 45$ secondi.

Perché le altre risposte sono errate:

- a) $interval \times retries$ — Questa formula non tiene conto del `timeout`. Se un check impiega il massimo tempo consentito prima di fallire, il tempo totale è maggiore di quanto suggerisce questa formula.

- **c) `timeout × retries + start_period`** — Questa formula non include l'`interval` e aggiunge erroneamente lo `start_period`. Lo `start_period` è un periodo di grazia separato durante il quale i fallimenti non vengono contati; non si somma al tempo per dichiarare unhealthy.
-

4. In un'architettura multi-tier, quale container può comunicare sia con la rete pubblica che privata?

Risposta corretta: b) Il container app (multi-homed, connesso a entrambe le reti)

Nel `docker-compose.yml` del Lab 02, il servizio `app` è configurato con due blocchi di rete: `vpc-public` (IP 10.0.1.20) e `vpc-private` (IP 10.0.2.20). Questo lo rende "multi-homed", capace di comunicare con i container di entrambe le reti.

Perché le altre risposte sono errate:

- **a) Il container web (solo rete pubblica)** — Il container `web` è connesso solo a `vpc-public` (IP 10.0.1.10). Non può raggiungere direttamente i container nella rete privata come il database.
 - **c) Il container db (solo rete privata)** — Il container `db` è connesso solo a `vpc-private` (IP 10.0.2.10). È completamente isolato dalla rete pubblica e da Internet, come previsto da INF-02.
-

5. Quale gruppo Linux deve contenere un utente per accedere al Docker socket?

Risposta corretta: c) `docker`

Il socket `/var/run/docker.sock` appartiene al gruppo `docker` con permessi 660. Solo l'utente root e i membri del gruppo `docker` possono interagire con il Docker daemon. Il comando `usermod -aG docker utente` concede l'accesso.

Perché le altre risposte sono errate:

- **a) `sudo`** — Il gruppo `sudo` concede la possibilità di eseguire comandi come root tramite il comando `sudo`, ma non dà accesso diretto al socket Docker. Un utente nel gruppo `sudo` potrebbe usare `sudo docker`, ma l'accesso non-root richiede il gruppo `docker`.
 - **b) `admin`** — Il gruppo `admin` non ha una relazione specifica con Docker. Non è nemmeno un gruppo standard su tutte le distribuzioni Linux.
-

6. Quale requisito di infrastruttura (INF) garantisce la persistenza dei dati con named volumes?

Risposta corretta: c) INF-04

INF-04 richiede l'uso di named volumes con driver `local` per garantire che i dati persistano oltre il ciclo di vita dei container. Verificabile con `docker volume ls` che mostra i named volumes.

Perché le altre risposte sono errate:

- **a) INF-01** — INF-01 riguarda l'esecuzione **non-root** dei processi. Si verifica con `docker exec container whoami` che deve restituire un utente diverso da root.
 - **b) INF-03** — INF-03 riguarda i **limiti di risorse** (CPU e memoria). Si implementa con `deploy.resources.limits` nel Docker Compose.
-

7. Qual è il valore in nanosecondi restituito da `docker inspect` per un limite di 1 CPU?

Risposta corretta: b) 1000000000

Docker rappresenta internamente i limiti CPU in nanosecondi. 1 CPU = 1 secondo di tempo CPU = 10^9 nanosecondi = 1.000.000.000. Il comando `docker inspect --format '{{.HostConfig.NanoCpus}}'` restituisce questo valore.

Perché le altre risposte sono errate:

- **a) 100000000** — Questo valore (10^8) corrisponderebbe a 0.1 CPU (un decimo di CPU), non a 1 CPU intero.
 - **c) 100000000000** — Questo valore (10^{10}) corrisponderebbe a 10 CPU, un valore tipicamente superiore al numero di core disponibili su macchine standard.
-

8. Quale opzione di `docker compose down` preserva i volumi?

Risposta corretta: a) Nessuna opzione aggiuntiva (i volumi sono preservati per default)

Per default, `docker compose down` ferma e rimuove container e reti, ma **preserva** i named volumes. I volumi vengono rimossi solo aggiungendo esplicitamente il flag `-v` (o `--volumes`).

Perché le altre risposte sono errate:

- **b) -k (keep volumes)** — Il flag `-k` non esiste per `docker compose down`. Non è necessario un flag per preservare i volumi perché è già il comportamento default.
 - **c) --preserve-volumes** — Questo flag non esiste. È un nome inventato che non corrisponde a nessuna opzione reale di `docker compose down`.
-

9. Quale comando testa la connettività di rete tra due container?

Risposta corretta: b) `docker exec container1 ping container2`

`docker exec` esegue un comando all'interno di un container in esecuzione. `ping` testa la raggiungibilità di rete verso un altro host. Quando i container sono nella stessa Docker network, il nome del servizio funge da hostname DNS.

Perché le altre risposte sono errate:

- **a) `docker network test container1 container2`** — Il sotto-comando `test` non esiste per `docker network`. I sotto-comandi validi sono: `connect`, `create`, `disconnect`, `inspect`, `ls`, `prune`, `rm`.

- **c) `docker connect container1 container2`** — `docker connect` non è un comando Docker valido. Per connettere un container a una rete esiste `docker network connect`, ma richiede una rete come argomento, non due container.

10. Quale UID viene tipicamente assegnato a un utente non-root creato con `adduser -D -u 1000`?

Risposta corretta: c) 1000

Il flag `-u 1000` specifica esplicitamente l'UID (User ID) 1000. Nel Dockerfile del Lab 01: `adduser -D -u 1000 -G labuser labuser` crea l'utente `labuser` con UID 1000. Il flag `-D` disabilita la richiesta di password (modalità non interattiva su Alpine).

Perché le altre risposte sono errate:

- **a) 0** — UID 0 è **sempre** l'utente root su sistemi Unix/Linux. Un utente con UID 0 ha pieni privilegi di sistema, violando il principio del minimo privilegio (INF-01).
- **b) 65534** — UID 65534 è convenzionalmente assegnato all'utente `nobody`, un utente speciale con privilegi minimi usato per processi che non devono possedere risorse.

11. In Docker Compose, come si specifica un indirizzo IP statico per un servizio?

Risposta corretta: c) `ipv4_address: 10.0.1.10`

In Docker Compose, sotto la sezione `networks` di un servizio, si usa `ipv4_address` per assegnare un IP fisso. Esempio dal Lab 02: `vpc-public: ipv4_address: 10.0.1.10`.

Perché le altre risposte sono errate:

- **a) `ip_address: 10.0.1.10`** — La chiave `ip_address` non è valida nella specifica Docker Compose. La chiave corretta è `ipv4_address` (per IPv4) o `ipv6_address` (per IPv6).
- **b) `static_ip: 10.0.1.10`** — La chiave `static_ip` non esiste nella specifica Docker Compose. È un nome inventato.

12. A quale servizio AWS corrisponde il healthcheck di Docker?

Risposta corretta: b) ELB Health Check (Elastic Load Balancer)

Nel materiale del corso, gli healthcheck Docker sono mappati agli ELB Health Check di AWS. Entrambi verificano periodicamente lo stato di salute dei servizi attraverso test configurabili (HTTP, TCP, comandi), con parametri di interval, timeout e soglie di fallimento.

Perché le altre risposte sono errate:

- **a) AWS CloudWatch Alarms** — CloudWatch Alarms monitora **metriche** (CPU, memoria, latenza) e invia allarmi quando superano soglie. Non esegue health check attivi su servizi. È più simile a `docker stats` con alerting.

- **c) AWS GuardDuty** — GuardDuty è un servizio di **sicurezza** che rileva minacce e attività sospette analizzando log e traffico di rete. Non ha relazione con la verifica della salute dei servizi.
-

13. Quale comando verifica se PostgreSQL è pronto ad accettare connessioni?

Risposta corretta: c) **pg_isready -U user -d database**

pg_isready è un'utilità inclusa nell'installazione PostgreSQL che verifica se il server accetta connessioni. **-U** specifica l'utente e **-d** il database. È usato negli healthcheck dei lab: **pg_isready -U lab05_user -d lab05_db**.

Perché le altre risposte sono errate:

- **a) psql --check-ready** — **--check-ready** non è un'opzione valida di **psql**. **psql** è un client interattivo per eseguire query, non uno strumento di health check.
 - **b) postgres --health-check** — **postgres** è il binario del server PostgreSQL. **--health-check** non è un'opzione valida. Il server si avvia con **postgres -D datadir**, non ha modalità di health check integrata.
-

14. Nel Lab 02, quale container funge da "ponte" tra rete pubblica e privata?

Risposta corretta: b) **lab02-app**

Nel **docker-compose.yml** del Lab 02, il servizio **app** è connesso sia a **vpc-public** (10.0.1.20) che a **vpc-private** (10.0.2.20). Questo gli permette di ricevere richieste dalla rete pubblica e inoltrare verso servizi nella rete privata (come il database).

Perché le altre risposte sono errate:

- **a) lab02-web** — Il container **web** è connesso **solo** a **vpc-public** (IP 10.0.1.10). Non ha interfaccia nella rete privata e non può raggiungere il database.
 - **c) lab02-db** — Il container **db** è connesso **solo** a **vpc-private** (IP 10.0.2.10). È completamente isolato dalla rete pubblica per garantire la sicurezza dei dati.
-

15. Quale valore in byte restituisce **docker inspect** per un limite di memoria di 4 GB?

Risposta corretta: b) **4294967296**

4 GB in byte = $4 \times 1024 \times 1024 \times 1024 = 4 \times 1.073.741.824 = 4.294.967.296$. Docker usa il sistema binario (potenze di 1024) per le unità di memoria. Il comando **docker inspect --format '{{.HostConfig.Memory}}'** restituisce questo valore esatto.

Perché le altre risposte sono errate:

- **a) 4000000000** — Questo valore (4×10^9) usa il sistema decimale (potenze di 1000), equivalente a circa 3.73 GiB. Docker usa il sistema binario dove $1 \text{ GB} = 1.073.741.824 \text{ byte}$.
- **c) 4096000000** — Questo valore (4096×10^6) mescola sistema binario e decimale: 4096 MB in base decimale, che non è corretto. In binario: $4096 \text{ MB} = 4096 \times 1.048.576 = 4.294.967.296$.

16. Quale versione minima di Docker Engine è richiesta per il corso?

Risposta corretta: c) 24.0

Lo script `check-env.sh` verifica che Docker Engine sia versione ≥ 24.0 . Anche il `STUDENT-GUIDE.md` e il README principale specificano questo requisito. Docker 24.0+ include Compose V2 integrato e le funzionalità necessarie per i lab.

Perché le altre risposte sono errate:

- **a) 20.0** — La versione 20.x è troppo datata e non include molte funzionalità usate nei lab, come la piena integrazione di Compose V2 come plugin del comando `docker`.
- **b) 23.0** — Sebbene Docker 23.0 includa Compose V2, il corso richiede specificamente la versione 24.0 o superiore come documentato nei prerequisiti e nello script di verifica.

17. Qual è la credenziale di default per l'utente root di MinIO nel Lab 04?

Risposta corretta: b) minioadmin / minioadmin123

Nel `docker-compose.yml` del Lab 04, le variabili d'ambiente sono: `MINIO_ROOT_USER: minioadmin` e `MINIO_ROOT_PASSWORD: minioadmin123`.

Perché le altre risposte sono errate:

- **a) admin / admin123** — Queste credenziali non corrispondono a quelle configurate nel lab. Le variabili d'ambiente di MinIO usano il prefisso `minioadmin`, non `admin`.
- **c) minio / minio_password** — Queste credenziali non sono presenti nella configurazione. Il nome utente reale è `minioadmin` (non `minio`) e la password è `minioadmin123` (non `minio_password`).

18. Quale framework documentale segue la struttura dei lab (Tutorial, How-to, Reference, Explanation)?

Risposta corretta: c) Diátaxis

La struttura dei lab del corso segue esplicitamente il framework Diátaxis, ideato da Daniele Procida, che organizza la documentazione tecnica in 4 quadranti: Tutorial (apprendimento guidato), How-to Guide (risoluzione problemi), Reference (informazioni tecniche) e Explanation (comprensione concettuale).

Perché le altre risposte sono errate:

- **a) ITIL** — ITIL (Information Technology Infrastructure Library) è un framework per la gestione dei servizi IT (incident management, change management, ecc.). Non riguarda la struttura della documentazione tecnica.

- **b) Agile Documentation** — Non è un framework documentale specifico e codificato. "Agile" si riferisce a metodologie di sviluppo software. Non definisce una struttura a 4 quadranti come Diátaxis.
-

19. Nella configurazione di rete Docker, cosa definisce `ip_range` nella sezione IPAM?

Risposta corretta: b) Il sottoinsieme di indirizzi assegnabili automaticamente ai container

Nel `docker-compose.yml` del Lab 02, `ip_range: 10.0.1.128/25` limita l'assegnazione automatica degli IP alla metà superiore della subnet (10.0.1.128-10.0.1.255). Gli IP fuori da questo range nella stessa subnet possono essere assegnati solo manualmente con `ipv4_address`.

Perché le altre risposte sono errate:

- **a) L'intervallo di IP riservati al gateway** — Il gateway è un singolo indirizzo IP (es. 10.0.1.1), non un range. `ip_range` non ha relazione con la configurazione del gateway.
 - **c) Gli IP bloccati nella subnet** — `ip_range` definisce gli IP **assegnabili**, non quelli bloccati. È l'opposto: restringe il pool di assegnazione automatica, non blocca indirizzi.
-

20. Quale tipo di istanza EC2 corrisponde a un container con 2 CPU e 4 GB di memoria?

Risposta corretta: b) t2.medium

Dalla tabella di mapping nel file `ec2-instance-mapping.md` del Lab 03: t2.medium = 2 vCPU, 4 GB RAM (`cpus: '2', memory: 4G`). Il servizio `worker` e il servizio `db` del Lab 03 sono configurati con queste risorse.

Perché le altre risposte sono errate:

- **a) t2.small** — t2.small ha 1 vCPU e 2 GB di RAM. Ha la metà delle risorse sia in CPU che in memoria rispetto al t2.medium.
- **c) m5.large** — m5.large ha 2 vCPU e **8 GB** di RAM. Ha le stesse CPU del t2.medium ma il doppio della memoria. Appartiene inoltre alla famiglia "general purpose" (m5), non alla famiglia "burstable" (t2).