# Test di valutazione – Microsoft Azure

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Alessia |
|  |  | Cognome | Lionetto |
|  |  | Data | 11/03/2022 |

Leggete attentamente ogni domanda e argomentare quanto più possibile fornendo anche degli esempi.  
ATTENZIONE: Le domande a risposta multipla possono contenere più risposte corrette.

1. Qual è il motivo principale per cui le aziende dovrebbero pensare di portare le loro risorse in cloud?

Perché potrebbero risparmiare sui costi di gestione dell’hardware

Perché il livello di sicurezza è “by design” superiore a quello medio che si può trovare in situazioni “on premise”

Perché le applicazioni sviluppate per il cloud sono più veloci

Perché è possibile scalare in maniera semplice e adeguare le risorse alle richieste del business

1. Quali tipologie di servizi sono disponibili nell’ambiente Cloud Microsoft Azure?

IAAS

PAAS

SAAS

Serverless

1. Fornire un elenco delle tipologie di servizio erogabile in un ambiente cloud (IAAS, PAAS, ecc) e fornire una descrizione di ciascuna di esse, spiegando quando è conveniente utilizzare una tipologia o l’altra.

-IAAS: contiene gli elementi di base dell’infrastruttura IT del Cloud(es: GoogleCloud, MicrosoftAzure), Generalmente, fornisce accesso a funzionalità di rete, computer (virtuali o su hardware dedicati) e spazi di archiviazione di dati.

-PAAS: fornisce all’utente una **piattaforma tramite la quale erogare la propria gamma di servizi. Non si deve più gestire l’infrastruttura sottostante (in genere hardware e sistemi operativi).** In questo modo sarà possibile concentrarsi sulla distribuzione e sulla gestione delle applicazioni.( es: GitHub,Kubernates)

-SAAS: si fa riferimento alle applicazioni per utenti finali. **Con il servizio SaaS, non devi preoccuparti della manutenzione o della gestione dell’infrastruttura sottostante su cui il servizio stesso si basa**. Dovrai solo pensare a come utilizzare il software specific(es:Netflix,facebook)

1. Quali sono le differenze principali tra un database SQL Azure e Azure CosmosDb e, in generale, tra un database relazionale e un database documentale NoSQL?

La rappresentazione che i dati hanno in essi; Nei database relazionali, i dati vengono salvati in delle tabelle seguendo un preciso schema che ogni dato deve seguire per poter essere salvato; Un database documentale è infatti basato non su record ma su documenti, ogni documento è indipendente e privo di schema.

**Cosmos DB**è un servizio di database NoSQL completamente gestito creato per prestazioni rapide e prevedibili, disponibilità elevata, scalabilità elastica, distribuzione globale e facilità di sviluppo.

**Database SQL Azure**È un servizio di database in grado di scalare in modo rapido ed efficiente per soddisfare la domanda, è automaticamente altamente disponibile e supporta una varietà di software di terze parti.

1. Descrivere alcune delle funzionalità dell’Azure Management Portal, per come è stato utilizzato durante le lezioni

Creazione Virtual Machine,creazione App services, creazione Eveny Hubs

1. Dare una descrizione delle principali differenze che ci sono tra un deploy (una pubblicazione) di una applicazione su Azure Virtual Machine e una su Azure AppService. Classificare i due servizi tra IaaS, PaaS o SaaS.

Si utilizza una macchina virtuale(servizio IAAS) invece che un servizio «managed»(azure app service,servizio PAAS) solitamente per la necessità di avere un maggior controllo sull’ambiente, in termine di configurazione e monitoraggio. Una VM di Azure offre la flessibilità della virtualizzazione senza dover acquistare e gestire l'hardware fisico su cui è in esecuzione la macchina virtuale.

1. Qual è lo scopo di servizi di messaging come può essere Azure Event Hub? Si dia una semplice descrizione di uno scenario di utilizzo.

Azure Event Hub è un servizio di inserimento di eventi. Può ricevere ed elaborare milioni di eventi al secondo. I dati inviati a un hub eventi possono essere trasformati e archiviati utilizzando qualsiasi provider di analisi in tempo reale.

Gli scenari seguenti sono alcuni degli scenari in cui è possibile utilizzare Hub eventi:

Registrazione delle applicazioni, Pipeline di analisi, Dashboard in tempo reale, Archiviazione dei dati

1. Per cosa possono essere utilizzati i servizi di Storage Account di Azure? Quali sono le principali 3 tipologie di servizio, all’interno di Storage Account, e quali sono le loro principali caratteristiche?

Possono essere utilizzati per l’archiviazione di dati,condivisione di file, code e tabelle. I 3 principali tipi di servizio che sono contenuti all’interno di uno Storage Account: Blob Storage: dedicati all’archiviazione di dati non strutturati e di larga scala; File Storage: simili ai Blob Storage, ma possono essere “montati” come risorse di rete (share) su machine virtuali; Disk Storage: vengono utilizzati come dischi per le machine virtuali e offrono un eccellente bilanciamento di velocità di lettura/scrittura.

1. Dare una descrizione del concetto di Container; poi fornire la motivazione per cui la “containerizzazione” di una applicazione può essere più efficiente del deploy della stessa applicazione in uno scenario IaaS o PaaS. Se le si conoscono, elencare le maggiori differenze che esistono tra una macchina virtuale e un container.

I «containers» offrono un meccanismo di pacchettizzazione logico in cui le applicazioni possono essere astratte dall'ambiente in cui vengono effettivamente eseguite. La containerizzazione, invece che virtualizzazione, ottimizza le risorse hardware a disposizione.

Il quantitativo di risorse utilizzate in un approccio tradizionale come quello delle macchine virtuali è molto elevato. Con l’adozione di container ogni singolo “box” è auto-consistente e comprensivo non solo del modulo applicativo, ma di tutte I requisiti tecnici necessari al suo corretto funzionamento.

1. Realizzare una applicazione ASP.NET utilizzando, .NET Core 5.0, che sia in grado di adempiere al seguente scenario di business. Si tenga presente che la prova è definita a “step”, e non è necessario completare tutti i punti, ma realizzare almeno il primo; gli step aggiuntivi completati saranno valutati positivamente nel giudizio finale.
   * SCENARIO:
     1. E' data una azienda di consulenza informatica di piccole medie-dimensioni (circa 30 dipendenti). Il personale tecnico (il team di sviluppo) svolge lavoro di implementazione su commessa, nella modalità "time & material". Il CFO (Chief Financial Officer) dell'azienda vorrebbe avere un sistema per eseguire in maniera semplice la fatturazione delle “ore lavorate” ai clienti finali, semplicemente ragionando a "Commessa" (a Progetto). Al completamento dell'attività di sviluppo (più le annesse attività di deploy, training o integrazione) vuole selezionare la commessa di pertinenza ed ottenere la cifra da fatturare al cliente finale, senza dover ricostruire tutto in maniera postuma. I dipendenti dovranno tracciare ogni singola ora spesa sui vari progetti allocando le 8 ore giornaliere di lavoro (per singolo giorno lavorativo) su uno o più progetti, in modo da semplificare il calcolo della fatturazione al CFO.
   * TECNICO: A seguire le caratteristiche tecniche funzionali del sistema:
     1. DATI: Definire la struttura dei progetti (id, nome progetto, nome cliente), la struttura dei dipendenti (id, nome, cognome), la struttura del timesheet (id, numero di ore, data del timesheet, id del progetto, id del dipendente). Il database da utilizzare deve essere SQL Azure
     2. APPLICAZIONE: E’ richiesta solo una struttura API (REST) che esponga i metodi per ottenere la lista dei progetti, la lista dei dipendenti, la lista dei timesheet sul singolo progetto (niente UI).
     3. DEPLOY: La soluzione dovrà essere rilasciata su Azure Virtual Machine
     4. METODOLOGIA: Il codice sorgente deve essere posizionato su Azure DevOps GIT Repo. Si dovranno utilizzare le tecniche di Git Feature Branch e Pull Request per l’integrazione del codice. Il tracciamento delle Features e User Story non è obbligatorio, ma è consigliato.
   * DELIVERABLES (STEPS):
     1. (STEP 1) Creazione della struttura della soluzione, usando repository pattern, definendo le Entità di business, il Business Layer, il Data Access Layer, e la struttura per le erogazioni dei dati dalle API.
     2. (STEP 2) Creare la funzione di erogazione della lista delle persone (es. “FetchAllEmployees”)
     3. (STEP 3) Creare la funzione di erogazione della lista dei progetti (es. “FetchAllProjects”)
     4. (STEP 4) Creare la funzione di erogazione dei timesheets per progetto (es. “FetchTimesheetsByProject”, passando l’id del progetto)
     5. (STEP 5) Creare la funzione di erogazione dei timesheets per impiegato (es. FetchTimesheetsByEmployee, passando l’id dell’impiegato)
     6. (STEP 6) Aggiungere i test funzionali per la verifica dei requisiti. Aggiungere i provider “mock” con dati di esempio
     7. (STEP 7) Configurare le pipelines di Build, Tests, Publish e Release (con YAML)
     8. (STEP 8) Configurare la pipeline di Release su Azure Virtual Machine
     9. (STEP 9) Se possibile, configurare un secondo rilascio (parallelo) su Azure AppService (usando la modalità di Pipeline Release classica/grafica di Azure DevOps, usando come Artifact il risultato della pipeline di Publish definita negli step precedenti)
   * DELIVERABLES (EXTRA STEPS):
     1. (STEP 10) Fornire un metodo di autenticazione alla piattaforma. Usando delle credenziali (username e password) e chiamando il metodo “api/Authentication/SignIn” si dovrà ottenere una valida autenticazione (200-OK) basandosi sui dati recuperati da una tabella “Users” presente nel database che contiene la username e la password di accesso.
     2. (STEP 11) Creare l’entità “Activity” (Attività) associata al Progetto. L’attività rappresenta una sottofase del progetto (es. “Analisi”, “Sviluppo”, “Testing”, ecc) e contiene il riferimento al progetto e all’elenco degli impiegati assegnati a quella attività. Esporre il metodo per erogare l’elenco delle attività di ogni progetto “FetchActivitiesByProject” e l’elenco degli impiegati assegnati ad ogni attività “FetchEmployeesByActivity”
     3. (STEP 12) Modificare l’oggetto TimeSheet per legare il timesheet non più al progetto, ma alla singola attività. I metodi degli step precedenti devono continuare a funzionare (“FetchTimesheetsByEmployee” e “FetchTimesheetsByProject”).
     4. (STEP 13) Esporre il metodo per creare nuovi Timesheet “api/Timesheets/Create” passando l’attività, l’impiegato e tutti i dati necessari alla creazione dell’attività