**ITS Running**



# 

# 

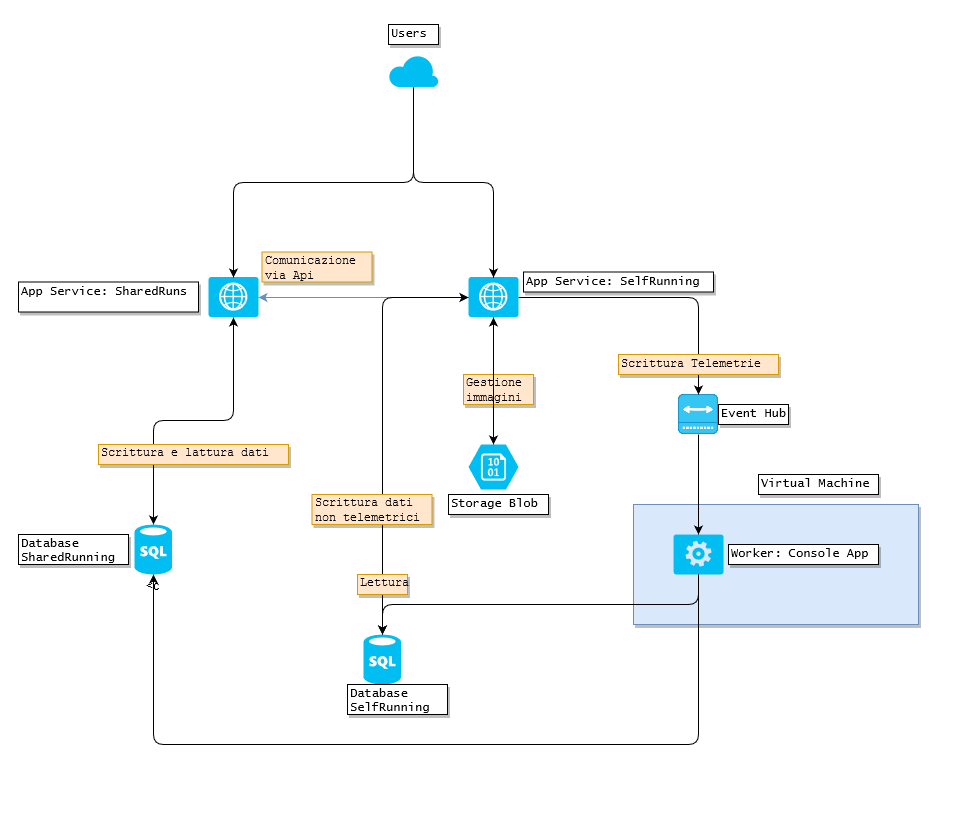
# 

# 1. Descrizione contesto

# 

L’applicativo è una soluzione per gestire l’attività di un Runner. Permette di tracciare i suoi dati telemetrici durante un suo allenamento o una sua gara, offrendone una visualizzazione dettagliata ad attività conclusa. Permette anche ad un organizzatore di creare delle gare, a cui i Runner possono partecipare.

# *2. Diagramma di infrastruttura*

****

***N.B.*** *Un eventuale sistema di scalabilità e repliche come availability zones, geo replication, regions, ecc…, non è rappresentato dal diagramma.*

# *3.Politica approcci*

**Generale:** principalmente ho fatto uso di tecnologia Microsoft, poichè per il deploy finale si utilizzano i servizi di Microsoft Azure.

**Database:** ho scelto un relazionale invece di un documentale per la natura dei dati dei nostri applicativi. Runner, Attività, Telemetry ad esempio sono entità legate fra loro da diverse chiavi esterne e per la loro struttura un database documentale avrebbe reso gestire la loro integrità molto più complesso. La mia scelta finale è stata MSSQL.

**Scrittura dei dati di telemetria:** ho utilizzato il servizio EventHub di Azure, in quanto si tratta proprio di un servizio di inserimento di dati di telemetria su vastissima scala. Può registrare infatti milioni di eventi al secondo quasi in tempo reale. Per la lettura dei dati in esso presente e la loro scrittura sul DB ho scelto una Console Application hostata su una macchina virtuale, invece di una Function di Azure, in quanto meno costosa. Nell’eventualità di una futura trasformazione dell’applicativo in prodotto la console application può benissimo essere trasformata in un servizio Microsoft da far operare sempre su una macchina virtuale.

**Scrittura degli altri tipi di dati e lettura:** le scritture di altro tipo rimanenti sono principalmente la creazione di nuovi utenti, la loro modifica e la creazione di eventi. In quanto queste scritture avverranno in un numero notevolmente minore rispetto alle scritture dei dati telemetrici, non ho reputato necessario una loro gestione tramite una coda.

Per quanto riguarda le letture di qualsiasi tipo di dato possiamo applicare lo stesso discorso, oltre anche al fatto che una lettura per un DB relazionale presenta un carico ridotto rispetto ad una scrittura.

# *4.Tecnologia e servizi*

**Basi salvataggio dei dati:**

* *Database scelto:* Microsoft SQL in un Database Server su Azure;
* *Storage per foto:* Archiviazione Blob dello Storage Service di Azure.

**Applicativi web:**

* *ITSSelfRunning e ITSSharedRuns:* Web Applications in Net Core 2.0 che adottano il pattern MVC e pubblicate tramite Application Service di Azure. In esse i linguaggi utilizzati sono principalmente C# e, nella parte client, Javascript.

**Gestione della coda:**

* *Coda:* i messaggi di scrittura delle telemetrie vengono inviati sul servizio EventHub di Azure;
* *Worker:* Console Application in Net Core pubblicata in una macchina virtuale su Azure, impostata per partire all’avvio del sistema operativo. Va a leggere i messaggi sull’EventHub per poi scriverne il contenuto sul DB.