

FantaNBA

Progetto di Scalable and reliable services

Alberto Garagnani

Leonardo DiFabio

Tommaso Muzzi

Thomas Bianconcini

OBIETTIVI



API che fornisce il
pronostico di una partita
di NBA



Scalabilità e Affidabilità
del servizio



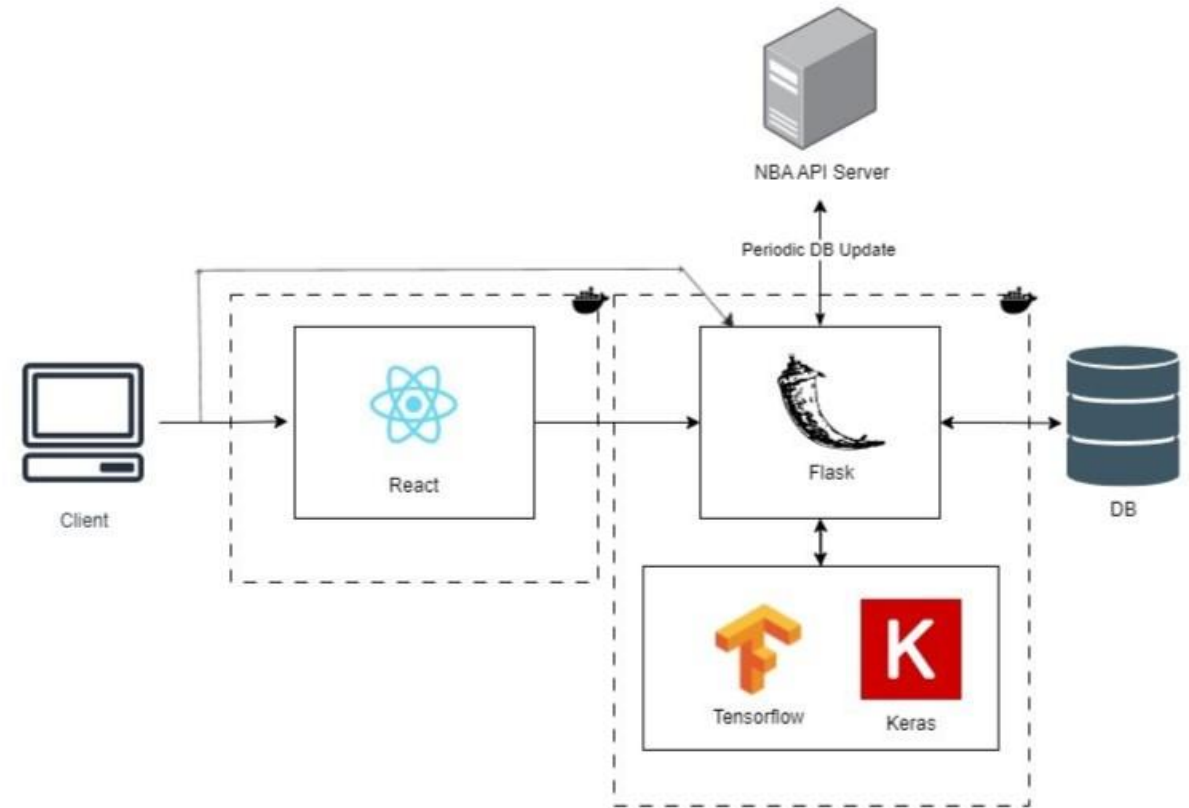
Sicurezza del servizio



Servizio di Frontend per
applicazione Web

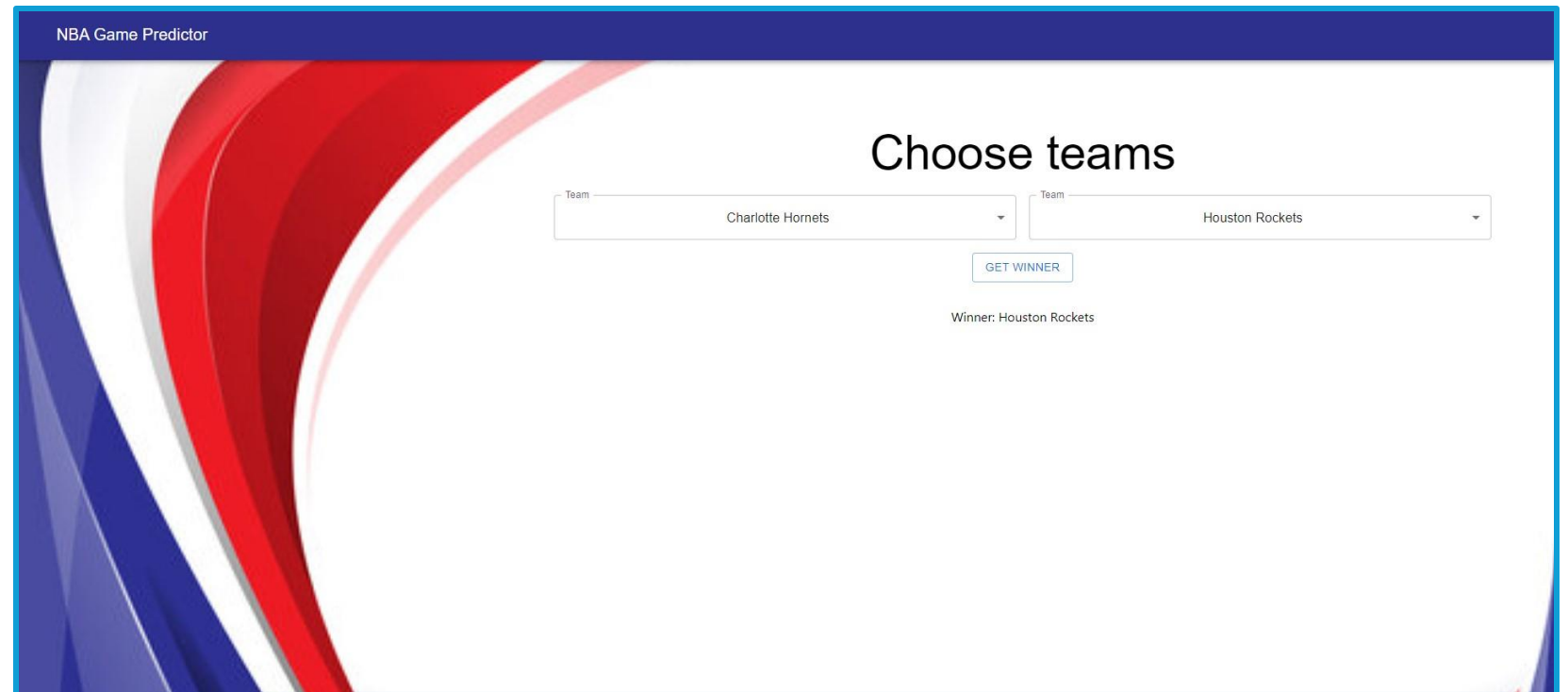
STRUTTURA APPLICAZIONE

- Frontend React
- Backend Flask
- Tensorflow & Keras
- Database



FRONTEND

- React – MUI
- Scelta squadre per partita da pronosticare
- Chiamata API verso backend



BACKEND



App Flask



Esposizione API su
indirizzo IP pubblico



Recupero dati dal
Database



Gestione predizione
risultato tramite Rete
Neurale

API

Esposizione API route /predict

01

Ricezione squadre

02

Recupero dati tramite il database

03

Predizione tramite intelligenza artificiale

04

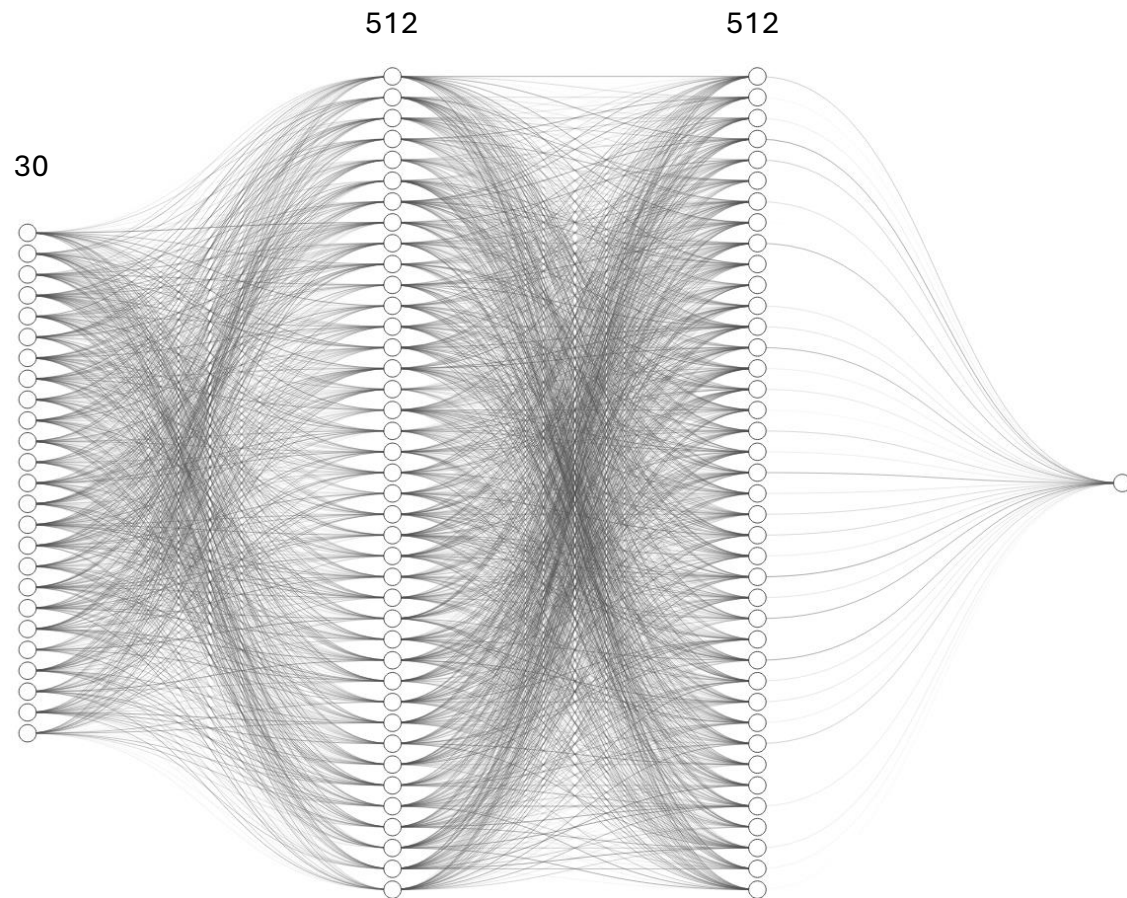
Generazione della risposta

RICHIESTE API



- Tramite richiesta dal frontend.
- Tramite curl -H "X-API-KEY: *****"
"https://172.213.194.23/api/predict?squadra1=Chicago%20Bulls&squadra2=Los%20Angeles%20Lakers"

RETE NEURALE



Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	15,872
dense_1 (Dense)	(None, 512)	262,656
dense_2 (Dense)	(None, 1)	513

Total params: 279,041 (1.06 MB)

Trainable params: 279,041 (1.06 MB)

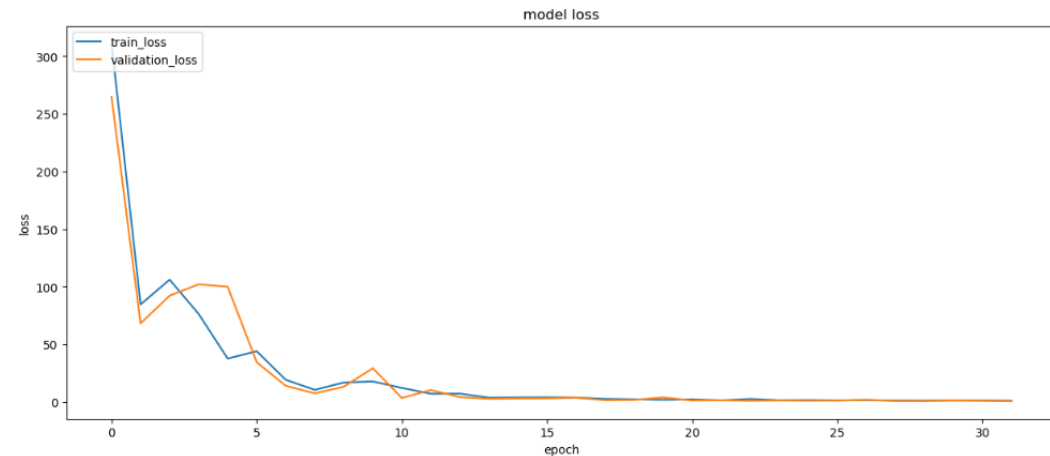
PREPARAZIONE DATI

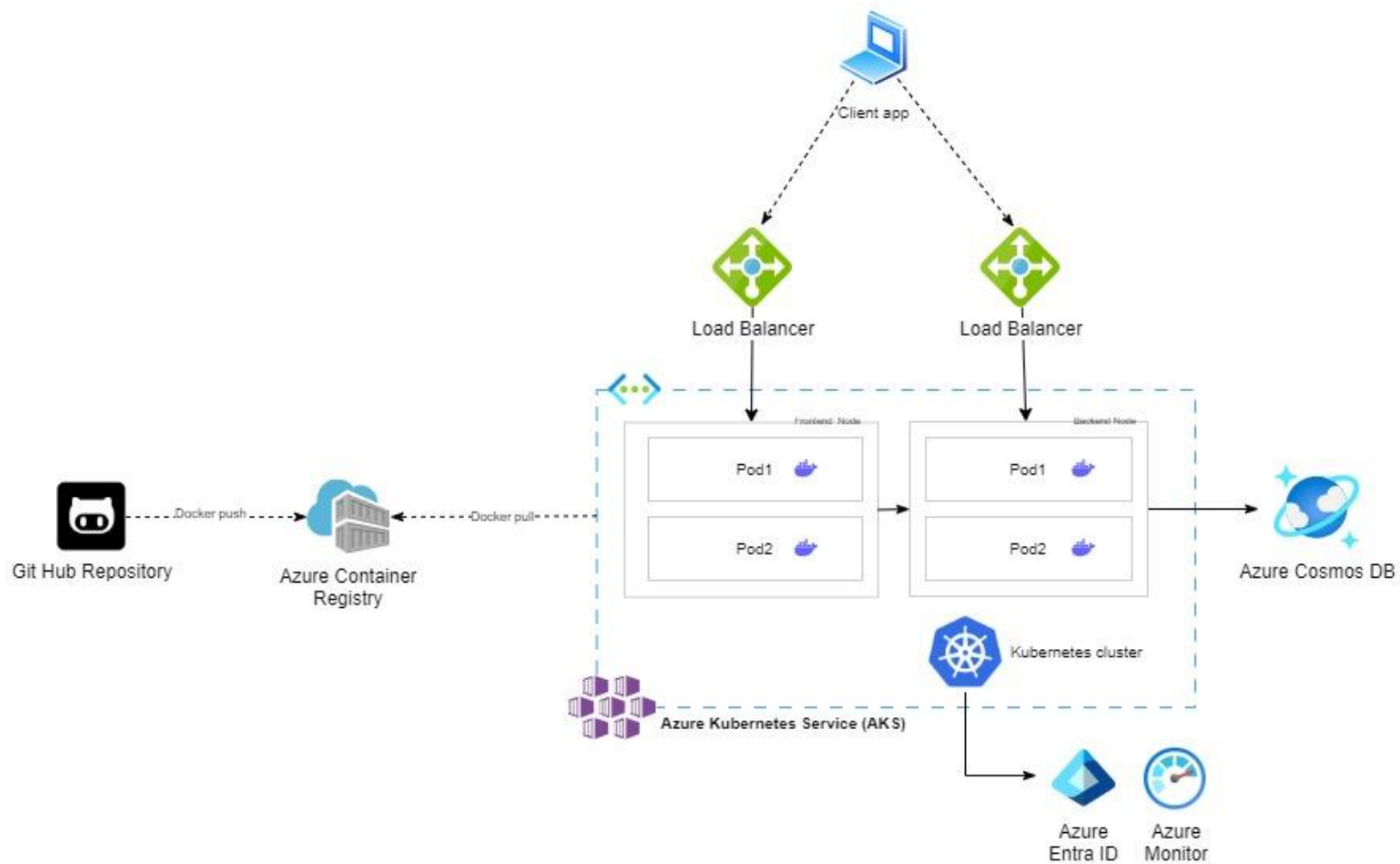
- Richiesta a nba_api dei dati per ogni singolo giocatore
- Parsing dati e unione in un unico array per squadra
- Standardizzazione dei dati

	Home_Stat_GP	Away_Stat_GP	Home_Stat_GS	Away_Stat_GS	Home_Stat_MIN	Away_Stat_MIN	Home_Stat_FG_PCT	Away_Stat_FG_PCT	Home_Stat_FG3
0	0.003216	0.779291	0.213389	-1.426351	-0.091494	0.131449	-1.072076	2.255496	-0.92
1	0.003216	0.560785	0.213389	-0.613040	-0.091494	0.034503	-1.072076	1.788833	-0.92
2	0.003216	-0.373519	0.213389	-1.531295	-0.091494	-1.404675	-1.072076	-0.457243	-0.92
3	0.003216	-0.373519	0.213389	-1.531295	-0.091494	-1.404675	-1.072076	-0.457243	-0.92
4	-0.237895	0.003216	-1.780535	0.213389	-0.786112	-0.091494	-0.493940	-1.072076	-0.98
...
3535	-0.026923	-0.682443	0.174035	0.108445	-0.209005	-0.546521	-0.032123	0.062733	0.18
3536	-0.682443	-1.066713	0.108445	-0.849163	-0.546521	-1.203275	0.062733	-0.569409	0.17
3537	-1.066713	-0.682443	-0.849163	0.108445	-1.203275	-0.546521	-0.569409	0.062733	-1.18
3538	-0.682443	-1.066713	0.108445	-0.849163	-0.546521	-1.203275	0.062733	-0.569409	0.17
3539	-1.066713	-0.682443	-0.849163	0.108445	-1.203275	-0.546521	-0.569409	0.062733	-1.18

TRAINING RETE

- Acquisizione storico delle partite negli ultimi 3 anni
- Preparazione Input
- Allenamento rete con Callback EarlyStopping e ModelCheckpoint





INFRASTRUTTURA
AZURE

Azure Kubernetes Service (AKS)

Servizio Azure per la gestione e il deploy di applicazioni multi-container tramite Kubernetes

- 2 nodi
 - Frontend
 - Backend
- 2 pod auto scalabili per ogni nodo
- Load balancer in ingresso al cluster
- ACR



Azure Container Registry (ACR)

- Servizio Azure per lo storage sicuro di immagini Docker sul cloud
- Funge da step intermedio tra la repository e il cluster Kubernetes su cui girano i servizi
- AKS pulla le immagini dal registry per deployarle



Cosmos DB

Servizio Azure per database gestiti:

- API per MongoDB NoSQL
- Scalabilità automatica
- Backup periodico
- Distribuibile tra più data center



AFFIDABILITA E SCALABILITA



- Load Balancer
 - Repliche dei deployment nei nodi
 - Autoscaler per entrambi i servizi
 - KEDA Scaler per servizio Backend
-

Sicurezza

- Protocollo HTTPS su frontend e backend
- Certificati autofirmati
- API Key
- Azure Virtual Network



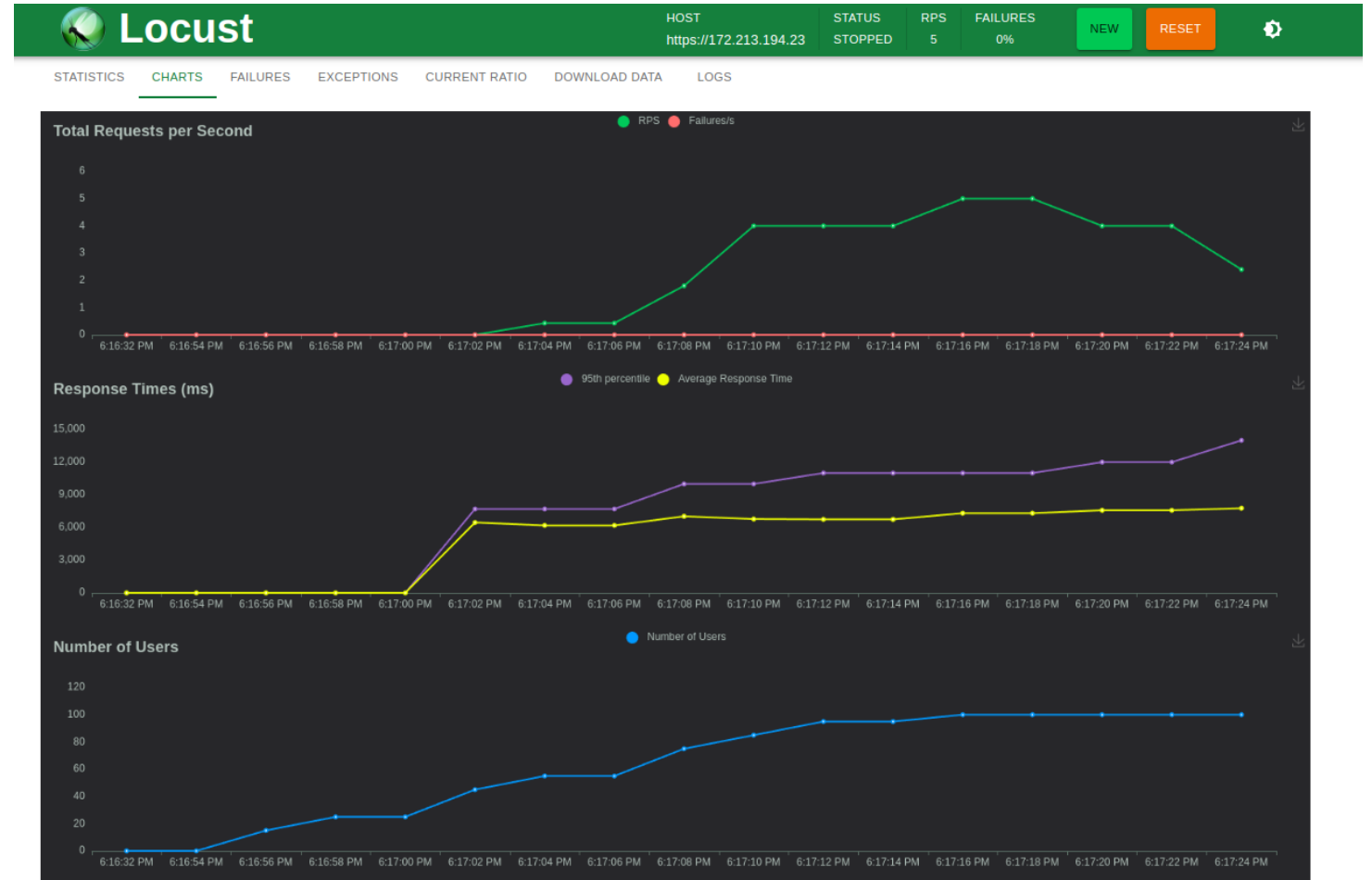
TEST FRONTEND

- Test di carico sul Frontend
- Parametri:
 - Max user: 100
 - Ramp-up: 10 user/sec
- Risultati:
 - Average RT: 80 ms

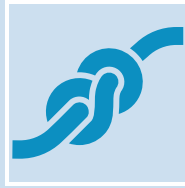


TEST BACKEND

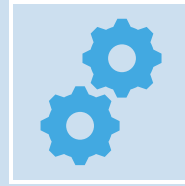
- Test di carico sul Backend
- Parametri:
 - Squadre casuali
 - Max user: 100
 - Ramp-up: 10 user/sec
- Risultati:
 - Average RT: 10 s



SVILUPPI FUTURI



Aggiungere nodi



Azure pipeline CI/CD per
DevOps



Live scoreboard frontend

GRAZIE DELL'ATTENZIONE