DelphiDay
italian conference
PADOVA EDITION
Edizione 2019

SMONTARE IL MONOLITE

Delphi ❤ Redis

wintech italia

Luca Minuti
Developer

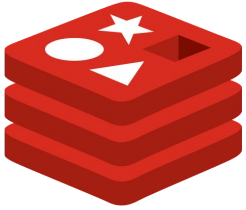
email
luca.minuti@gmail.com
GitHub
https://github.com/lminuti/

DelphiDay
italian conference
PADOVA EDITION
Edizione 2019

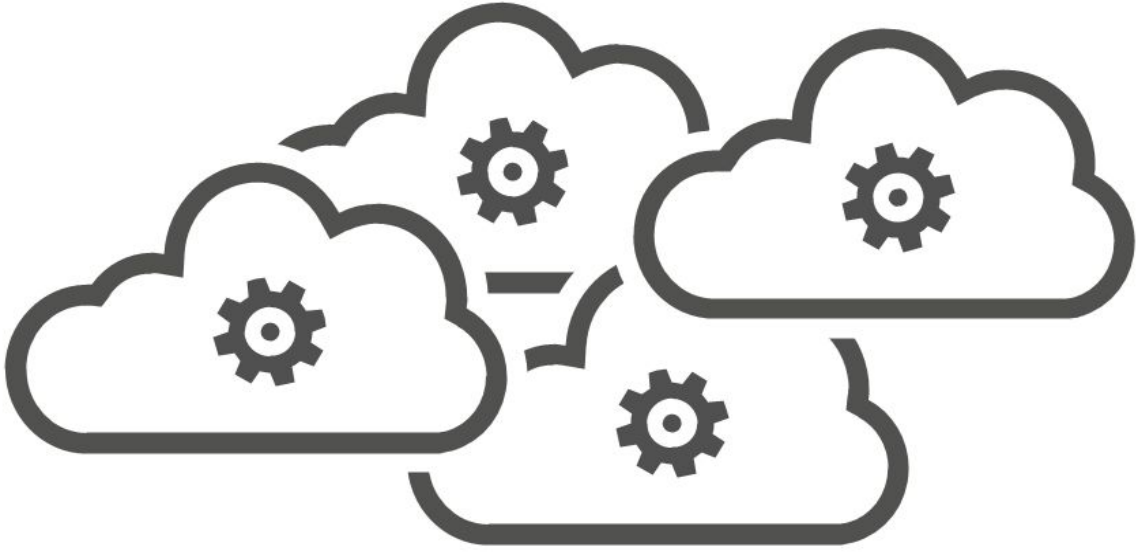
wintech italia

AGENDA

- → Microservices
- → Cos'è REDIS
- → Casi d'uso
- → Comandi principali
- → Esempi



DelphiDay italian conference



Microservices

COSA SONO

- ➔ Tanti servizi che lavorano insieme
- ➔ Un servizio si occupa di una sola cosa (una cosa ≠ una API)
- ➔ Single responsibility principle
- ➔ Filosofia unix: una solo cosa fatta bene

- ➔ Attenzione a non creare “Nanoservice”

NOVITÀ?

- ➔ Shared library (DLL)
- ➔ Moduli
- ➔ Delphi package

PRO

- ➔ Lavoro diviso su più team
- ➔ Applicazioni più piccole
- ➔ Continuous deployment
- ➔ Scalabilità: più copie su più macchine
- ➔ Molti framework, architetture e linguaggi di programmazione

CONTRO

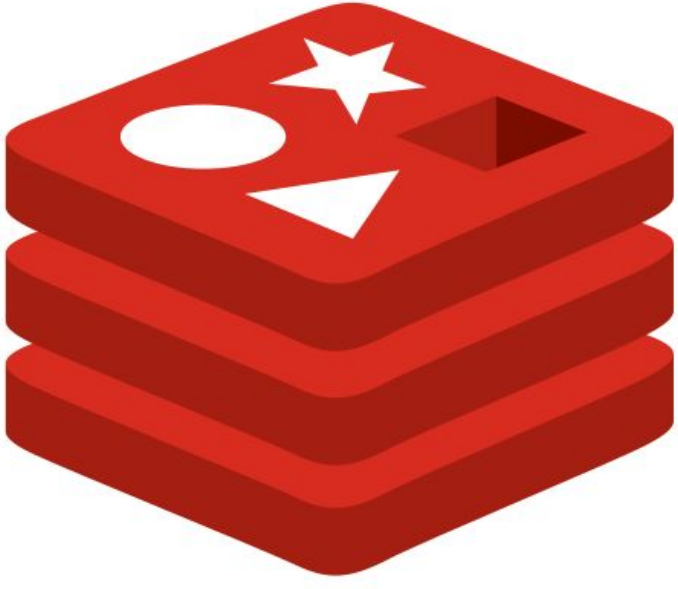
- ➔ Complessità
  - Collaborazione tra i servizi
  - Individuazione dei servizi
  - Consistenza
  - Performance
  - Deployment
  - Autenticazione e sicurezza

QUINDI?

- ➔ Monolitico non è sempre una cattiva idea
  - Permette uno sviluppo più rapido nelle prime fasi del progetto
  - Va bene all'inizio ma può trasformarsi in una giungla inestricabile
  - Non è male quando c'è un singolo team che ci lavora

COM'È FATTO?

- Data store
  - Ogni servizio dovrebbe avere il suo DB/storage
- Logica applicativa
- Comunicazione con l'esterno
  - Sincrona (ReST, SOAP, RPC, ...)
  - Asincrona (code)



redis

DelphiDay
italian conference

REDIS

- REmote DIctionary Server
- Sviluppato da Salvatore Sanfilippo (aka antirez) nel 2009
- Licenza BSD
- Key-Value database con persistenza opzionale (lavora in memoria)
- Performance

TIPI DI DATI

- Strings
- Lists
- Sets
- Sorted Sets (value + score)
- Hashes

FUNZIONALITÀ

- ➔ Performace (100k SET/s - >80k GET/s)
- ➔ Built-in replication
- ➔ Lua scripting
- ➔ LRU (Least recently used) eviction*
- ➔ Transazioni
- ➔ Diversi livelli di persistenza su disco
- ➔ High availability con Redis Sentinel

* Disponibile anche LFU da Redis 4.0

PERSISTENZA 1/3

- In “**Snapshotting mode**” salva un dump della memoria ogni:
  - **X** secondi
  - **Y** operazioni
- Quello che viene salvato è uno snapshot della memoria
- **Se il server muore tra uno snapshot e l’altro i dati vengono persi**

PERSISTENZA 2/3

- Modalità **Append-Only File (AOF)**
- Ogni comando viene salvato su file
- Diverse modalità
  - fsync() per ogni comando
  - fsync() ogni secondo
  - Lascia decidere l'OS

PERSISTENZA 3/3

➔ Nessuna persistenza...

CASI D'USO

- ➔ Cache lato server
- ➔ Invio di messaggi asincroni tra vari moduli dell'applicazione
- ➔ Gestione dello stato dell'applicazione
- ➔ Sincronizzazione tra microservizi
- ➔ Database temporaneo

COMANDI

Keys: inserimento di coppie chiave/valore

Lists: gestione di liste, code, pile, ecc.

Pub/Sub: messaggi Publish/Subscribe

Scripting: Esecuzione di script server side

Sets: gestione di set di valori

Strings: permette di manipolare chiavi di tipo stringa

Transactions: permette di eseguire più comandi garantendo l'atomicità

demo time

DELPHI

- Delphi Redis Client
- Sviluppata da Daniele Teti (Apache license)
- Due versioni
  - Redis Client versione 2: Delphi Berlin o superiore
  - Redis Client versione 1: Delphi XE5 o superiore
- Piattaforme
  - Win32/64
  - Mobile
  - Linux (solo versione 2)

https://github.com/danieleteti/delphiredisclient

Esempio

```
uses
  System.SysUtils,
  Redis.Commons, // Interfaces and types
  Redis.Client, // The client itself
  Redis.NetLib.INDY, // The tcp library used
  Redis.Values; // nullable types for redis commands
var
  lRedis: IRedisClient;
begin
  lRedis := TRedisClient.Create;
  lRedis.Connect;
  lRedis.&SET('firstname', 'Daniele');
end;
```

demo time

THANK YOU

wintech italia