EBook Gratis

APRENDIZAJE rethinkdb

Free unaffiliated eBook created from
Stack Overflow contributors.

#rethinkdb

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: rethinkdb

It is an unofficial and free rethinkdb ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official rethinkdb.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con rethinkdb

Observaciones

¿Qué es RethinkDB?

- Base de datos de **código abierto** para construir aplicaciones web en tiempo real
- Base de datos **NoSQL** que almacena documentos JSON sin esquemas
- Base de datos **distribuida** que es fácil de escalar
- Base de datos de **alta disponibilidad** con conmutación por error automática y robusta tolerancia a fallos

RethinkDB es la primera base de datos escalable de código abierto creada para aplicaciones en tiempo real. Expone un nuevo modelo de acceso a la base de datos: en lugar de sondear los cambios, el desarrollador puede indicar a la base de datos que envíe continuamente los resultados de las consultas actualizadas a las aplicaciones en tiempo real. RethinkDB permite a los desarrolladores crear aplicaciones escalables en tiempo real en una fracción del tiempo con menos esfuerzo.

Versiones

Versión	Fecha de lanzamiento
2.3.5	2016-08-27
2.3.4	2016-06-03
2.3.3	2016-05-31
2.3.2	2016-05-05
2.3.1	2016-04-22
2.3.0	2016-04-05

Examples

Instalación en OS X

Usando Homebrew

Requisitos previos: asegúrese de estar en OS X 10.9 (Mavericks) o superior, y tenga Homebrew instalado.

Ejecuta lo siguiente en tu terminal:

```
brew update && brew install rethinkdb
```

Compilar desde la fuente

La construcción de RethinkDB desde la fuente requiere OS X 10.9 (Mavericks) o superior. Se requiere Xcode para construir desde la fuente.

Obtener el código fuente

Descarga y extrae el archivo:

```
wget https://download.rethinkdb.com/dist/rethinkdb-2.3.4.tgz
tar xf rethinkdb-2.3.4.tgz
```

Construir RethinkDB

Iniciar el proceso de construcción:

```
cd rethinkdb-2.3
./configure --allow-fetch --fetch openssl
make
```

Encontrará el binario `rethinkdb` en la `build/release/` subcarpeta.

Instalación en Ubuntu

Los binarios de Ubuntu están disponibles para arquitecturas de 32 y 64 bits.

```
source /etc/lsb-release && echo "deb http://download.rethinkdb.com/apt $DISTRIB_CODENAME main"
| sudo tee /etc/apt/sources.list.d/rethinkdb.list
wget -qO- https://download.rethinkdb.com/apt/pubkey.gpg | sudo apt-key add -
sudo apt-get update
sudo apt-get install rethinkdb
```

Instalación en Windows

Descargando

Requisitos previos: Proporcionamos binarios nativos de 64 bits para Windows 7 y superiores. Se requiere una versión de 64 bits de Windows.

Descargue el archivo ZIP y descomprímalo en un directorio de su elección.

El puerto de Windows de RethinkDB es una adición reciente y aún no ha recibido tantos ajustes como las versiones de Linux y OS X. Por favor, informe de cualquier problema de rendimiento en

GitHub .

Ejecutando RethinkDB

La versión de Windows de RethinkDB, al igual que las versiones de Linux / OS X, se ejecuta desde la línea de comandos. Tendrá que iniciar el shell de comandos de Windows.

- Presione `Win` + `X` y haga clic en "Símbolo del sistema"; o
- Abra el menú Inicio, haga clic en "Ejecutar" y escriba "cmd" `ENTER`

Use el comando `cd` para ir al directorio en el que desempaquetó `rethinkdb.exe` .

```
C:\Users\Slava\>cd RethinkDB
C:\Users\Slava\RethinkDB\>
```

A continuación, puede iniciar RethinkDB con sus opciones predeterminadas.

```
C:\Users\Slava\RethinkDB\>rethinkdb.exe
```

También puede usar cualquiera de las opciones de la línea de comandos para controlar la configuración (así como especificar un archivo de configuración).

Para comenzar con un directorio de datos específico:

```
rethinkdb.exe -d c:\RethinkDB\data\
```

Para especificar un nombre de servidor y otro grupo para unirse:

```
rethinkdb.exe -n jarvis -j cluster.example.com
```

Lea Empezando con rethinkdb en línea:
https://riptutorial.com/es/rethinkdb/topic/2498/empezando-con-rethinkdb

Capítulo 2: Empezando con Node

Examples

Instalando el paquete RethinkDB desde NPM

```
npm install -g rethinkdb
```

Haciendo una conexión a RethinkDB

const r = require ("rethinkdb");

```
r.connect({host: 'localhost', port: 28015}, (conn) => console.log(conn))

// Or as a promise

let rdb_conn;
r.connect({host: 'localhost', port: 28015}).then((conn) => {
  rdb_conn = conn;
}).then(() => {
  // Continue to use rdb_conn
});
```

Listado de todas las bases de datos

```
r.connect({host: 'localhost', port: 28015})
.then((conn) => {
    return r.dbList().run(conn);
}).then((result) => {
    // Prints out list of databases on the RethinkDB instance
    console.log(result);
});
```

Crear una nueva base de datos

```
r.connect({host: 'localhost', port: 28015})
.then((conn) => {
    return r.dbCreate("stackoverflow").run(conn);
}).then((result) => {
    console.log(result);
});
```

Crear una nueva tabla en una base de datos

```
r.connect({host: 'localhost', port: 28015})
.then((conn) => {
    return r.db("stackoverflow").tableCreate("examples").run(conn);
}).then((result) => {
    console.log(result);
```

```
});
```

Insertar un documento en una tabla

```
r.connect({host: 'localhost', port: 28015})
.then((conn) => {
    return r.db("stackoverflow").table("examples")
        .insert({
            // If `id` is not set, will automatically generate a UUID
            id: 1,

            name: 'Thinker',

            // Will translate Date types.
            creationDate: new Date(),

            // Embedded array
            tags: ['rethinkdb', 'rethinkdb-javascript', 'rethinkdb-python'],

            // Will evaluate `r.now()` using server time
            dateWithServerTime: r.now(),

            // Embedded document
            location: {
                // Using geospatial example
                coordinates: r.point(-122.423246,37.779388),
                name: 'San Francisco'
            },
        }).run(conn);
}).then((result) => {
    // Returns results object which includes array of generated UUIDs
    //  for inserted documents
    console.log(result);
});
```

Consultar un documento de una tabla

```
r.connect({host: 'localhost', port: 28015})
.then((conn) => {
    // Can also use .get({id: 1})
    return r.db("stackoverflow").table("examples").get(1).run(conn)
}).then((result) => {
    console.log(result);
})
```

Lea Empezando con Node en línea: https://riptutorial.com/es/rethinkdb/topic/6475/empezando-con-node

Capítulo 3: Usando RethinkDB con Docker

Examples

Uso básico

De forma predeterminada, RethinkDB enlaza todos los servicios a `127.0.0.1` . Por lo tanto, este ejemplo siguiente conservará los datos en la `host_data_path` de `host_data_path` en la máquina host del contenedor y estará disponible para `127.0.0.1` en los puertos estándar.

Servicio	Bandera	Puerto predeterminado
Conductor	`--driver-port`	28015
Racimo	`--cluster-port`	29015
HTTP WebUI	`--http-port`	8080

```
docker run -d -v host_data_path:/data rethinkdb
```

Para abrir el controlador y el puerto del clúster al tráfico externo, debe especificar la dirección de las interfaces locales o proporcionar `all` .

```
docker run -d -v host_data_path:/data rethinkdb --bind all
```

Enlace de WebUI a localhost o inhabilitación

Al implementar RethinkDB en producción, desea desactivar o bloquear la WebUI. Esto solo responderá a `localhost` para acceder a la WebUI, lo que le permitirá acceder al SSH Tunnel al equipo host y acceder a él para realizar diagnósticos y solucionar problemas.

```
docker run -d \
   -v host_data_path:/data \
   rethinkdb \
   rethinkdb --bind-cluster all --bind-driver all --bind-http 127.0.0.1 -d /data
```

Si desea desactivar completamente la WebUI:

```
docker run -d \
  -v host_data_path:/data \
  rethinkdb \
  rethinkdb --bind-cluster all --bind-driver all --no-http-admin -d /data
```

Lea Usando RethinkDB con Docker en línea:
https://riptutorial.com/es/rethinkdb/topic/4499/usando-rethinkdb-con-docker

Capítulo 4: utilizando thinky.io con RethinkDB

Examples

Iniciando thinky en node.js

thinky es un ORM de node.js ligero para RethinkDB.

Primero necesitas tener RethinkDB corriendo en tu servidor.

Luego instale el paquete thinky.io npm en su proyecto.

```
npm install --save thinky
```

Ahora importa thinky en tu archivo de modelo.

```
const thinky = require('thinky')();
const type = thinky.type
```

A continuación crear un modelo.

```
const User = thinky.createModel('User' {
    email:type.string(),
    password: type.string()
});
```

Ahora puedes crear y guardar un usuario.

```
const user = new User({
    email: 'test@email.com',
    password: 'password'
});

user.save();
```

Al usuario se le dará una identificación única.

Puede encadenar una promesa a la función de guardar para usar el usuario resultante, o detectar un error.

```
user.save()
    .then(function(result) {
        console.log(result);
    })
    .catch(function(error) {
        console.log(error);
    });
```

Encuentre a su usuario usando funciones como el filtro y use promesas para usar los resultados.

```
User.filter({ email: 'test@email.com }).run()
    .then(function(result) {
        console.log(result);
    })
    .catch(function(error) {
        console.log(error);
    });
```

O busque un usuario específico por el id único.

```
User.get(id)
    .then(function(user) {
        console.log(user);
    })
    .catch(function(error) {
        console.log(error);
    });
```

Lea utilizando thinky.io con RethinkDB en línea:
https://riptutorial.com/es/rethinkdb/topic/4068/utilizando-thinky-io-con-rethinkdb

Creditos

S. No	Capítulos	Contributors
1	Empezando con rethinkdb	Community, dalanmiller, Matteo
2	Empezando con Node	dalanmiller
3	Usando RethinkDB con Docker	dalanmiller
4	utilizando thinky.io con RethinkDB	alexi2