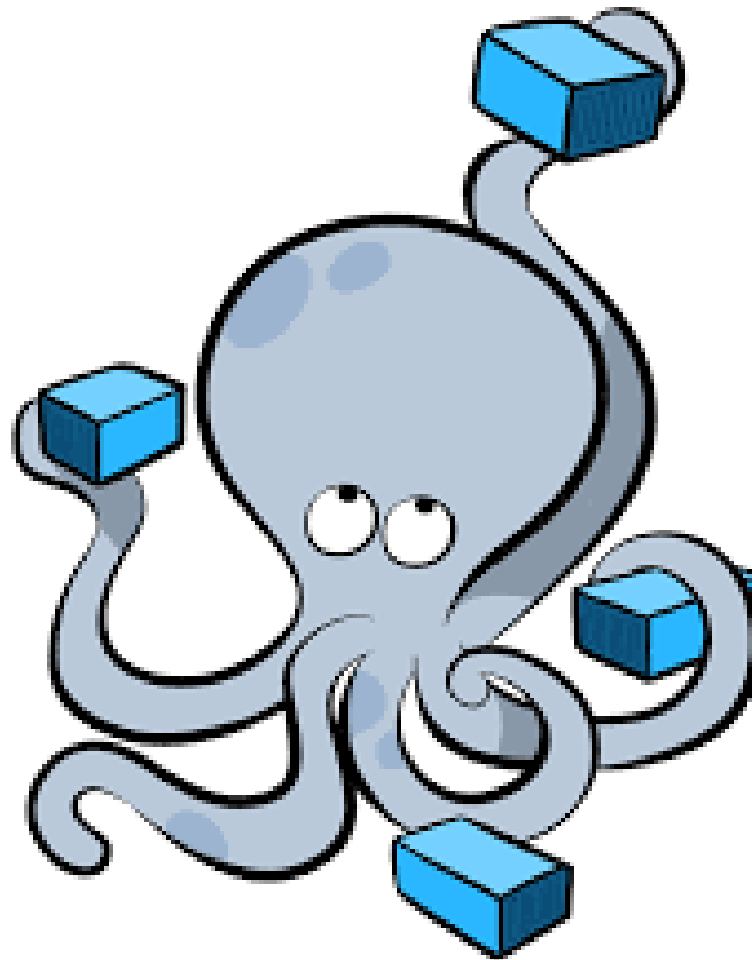


Docker compose, Redis

9 декабря 2016

Владимир Ананьев
АО "Программный Регион"

Docker compose



Что это такое?

- docs.docker.com/compose/overview/ (<https://docs.docker.com/compose/overview/>)
- Утилита для создания и запуска **многоконтейнерных** приложений (систем)
- Используется **один** конфигурационный файл для конфигурации всего приложения
- Используется **одна** команда для создания и запуска такого приложения

Основные варианты использования

- Окружение для разработчика (development environment). Позволяет быстро разворачивать окружение со всеми зависимостями: базы данных, очереди, кеши, веб-сервисы и т.п.

```
$ docker-compose up
```

- Окружение для автоматического тестирования (automated testing environment)

```
$ docker-compose up -d  
$ ./run_tests  
$ docker-compose down
```

Демо: показать преимущество подхода с использованием Compose

Установка

- В первую очередь нужно установить сам **Docker**
- Для **Windows** и **MacOS** устанавливается в рамках **Docker Toolbox**
- Для Linux необходимы следующие шаги
- `$ curl -L "https://github.com/docker/compose/releases/download/1.9.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`
- `$ chmod +x /usr/local/bin/docker-compose`
- И проверить работоспособность
- `$ docker-compose --version`

```
$ docker-compose --version  
docker-compose version 1.9.0, build 2585387
```

- Альтернативный метод - использовать `pip`

```
pip install docker-compose
```



- redis.io/ (<https://redis.io/>)
- Используется как **in-memory** база данных, кеш и сервис обмена сообщениями
- Это **key-value** база данных
- Ключ (**Key**) - любая последовательность байт, в простейшем случае - строка.
- Значение (**Value**) - **string, hash, list, set, sorted set, bitmaps, hyperLogLogs**
- redis.io/topics/data-types-intro (<https://redis.io/topics/data-types-intro>)
- try.redis.io/ (<http://try.redis.io/>)

Демо: Что же такое key-value и как это использовать?

Установка

```
$ wget http://download.redis.io/releases/redis-3.2.6.tar.gz  
$ tar xzf redis-3.2.6.tar.gz  
$ cd redis-3.2.6  
$ make
```

- Для контроля

```
$ make test
```

- Запуск

```
$ src/redis-server
```

- Доступ

```
$ src/redis-cli
```

Redis protocol

- **RESP** - REdis Serialization Protocol
- redis.io/topics/protocol (<https://redis.io/topics/protocol>)
- Простой в реализации
- Быстрый парсинг
- Human readable

Основные команды

- redis.io/commands (<https://redis.io/commands>)
- PING
- SET/GET
- DEL
- EXISTS
- EXPIRES
- KEYS
- TTL
- TYPE
- INCR/DECR/INCRBY

Демо: запуск команд через консоль

Клиент Redis'а для Golang

- Их довольно много redis.io/clients#go (<https://redis.io/clients#go>)
- Рассмотрим github.com/mediocregopher/radix.v2 (<https://github.com/mediocregopher/radix.v2>)

```
go get github.com/mediocregopher/radix.v2/...
```

- Создаем подключение

```
client, err := redis.Dial("tcp", "localhost:6379")
if err != nil {
    // Handle error
}
```

- И просто выполняем команды

```
resp := client.Cmd("MGET", "foo", "bar", "baz")
if resp.Err != nil {
    // Handle error
}
```

Демо: разные типы возвращаемых значений

Сборка Compose: Golang приложение + Redis

- Создаем файл **docker-compose.yml**:

```
version: '2'
services:
  app:
    build: .
    ports:
      - "8080:8080"
    depends_on:
      - redis
  redis:
    image: redis
...
$ docker-compose up
$ docker-compose up SERVICE
...
environment:
  - RACK_ENV=development
expose:
  - "3000"
```

Задание

- Подготовка к тестированию 16 декабря
- 2-3 простых вопроса по каждому занятию
- По Golang может быть чуть больше вопросов

Thank you

9 декабря 2016

Владимир Ананьев

АО "Программный Регион"

vladimir.ananyev@regium.com (mailto:vladimir.ananyev@regium.com)

