simulacron training

Mauricio generoso

Agenda

1. What is Simulacron
2. Cassandra cluster quick overview
3. Simulacron examples
4. Demo
5. Exercises

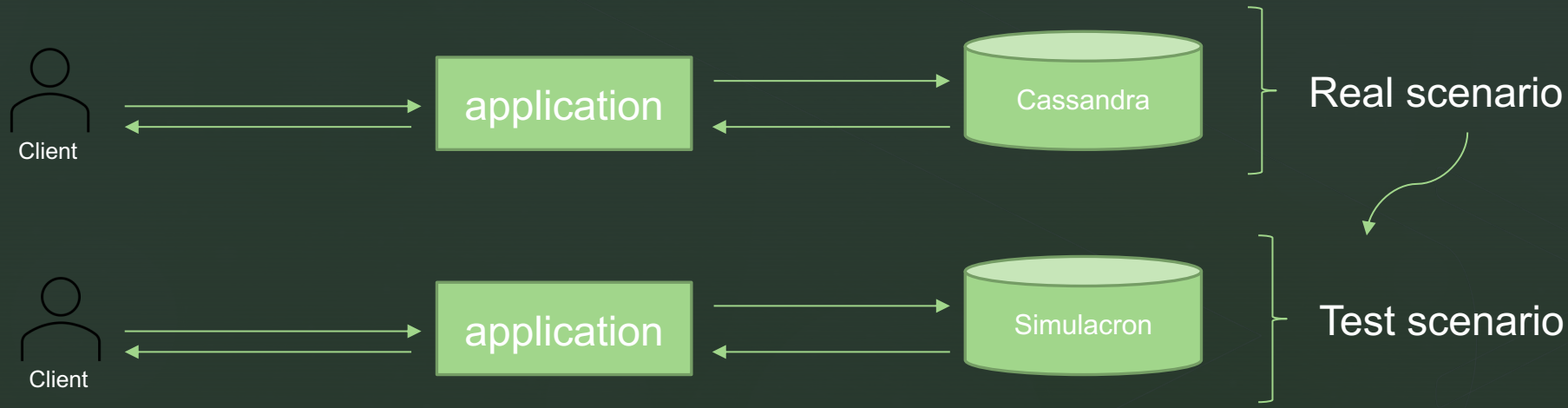
What is simulacron

- Simulacron is a native protocol server simulator that helps facilitate the testing of scenarios that are difficult to reproduce in driver clients and applications
- We can create with Simulacron a Cassandra cluster and prime the database responses or failures likes read timeout, server unavailable, and more
- Like downstream mocks, we can simulate catastrophes scenarios and ensure our application is able to be stable

Simulacron happy path scenarios

- How can we insert data into the database without a real database?
- How we verify the database content and retrieve it from database?

- Simulacron will help with that



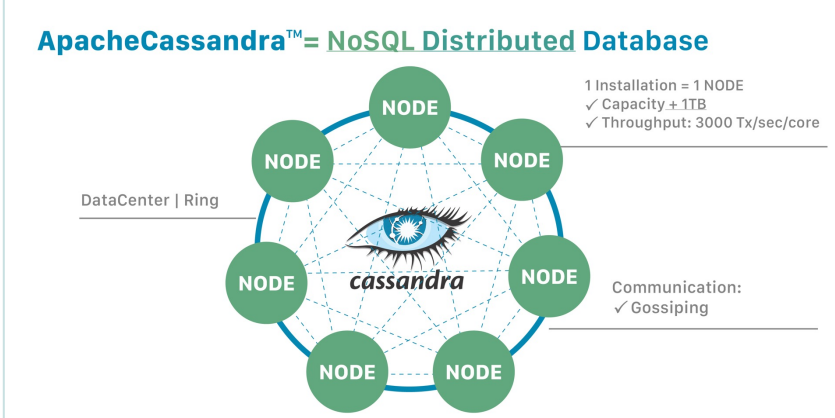
Simulacron error scenarios

- How do we test read timeout?
- How to we test connection failures?

- Simulacron will help with that

Cassandra cluster quick overview

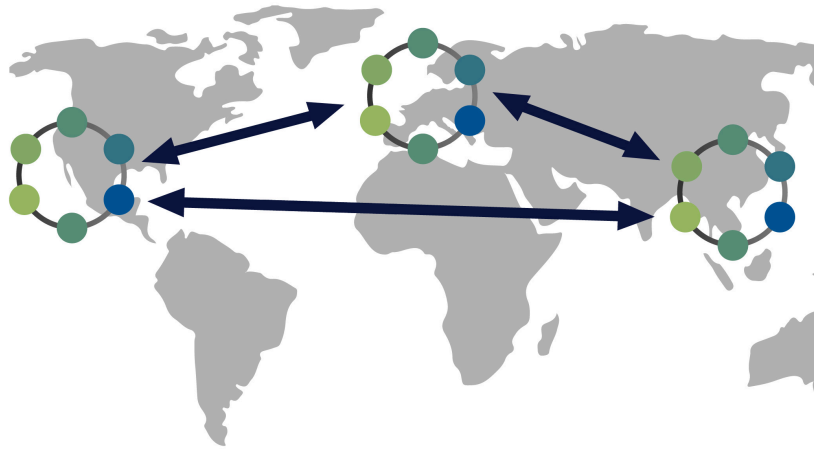
- Cassandra is a NoSQL Distributed database
- Cassandra is bases in a ring of nodes
- Provides data replication, high availability and fault tolerance



Source: https://cassandra.apache.org/_/cassandra-basics.html

Region duplication

- Cassandra distribute data across different region
- In case one region gets a catastrophe, data will be recovered from other regions



Source: https://cassandra.apache.org/_/cassandra-basics.html

Simulacron examples

Add dependency

We need to add the dependency:

```
<dependency>
      <groupId>com.datastax.oss.simulacron</groupId>
      <artifactId>simulacron-driver-3x</artifactId>
      <version>0.8.9</version>
</dependency>
```

Creating a cluster

```
// 1 - The main entry point to create a Cluster
var server = Server.builder().build();
```

```
// 2 - The cluster specification contains the datacenter and nodes
var clusterSpec = ClusterSpec.builder().build();
```

```
// 3 - The host and port to a single node
var addressResolver = new Inet4Resolver(defaultStartingIp,
CASSANDRA_PORT);
```

```
// 4 - Create the datacenter details
var dataCenter = clusterSpec.addDataCenter()
        .withName("datacenter1").withCassandraVersion("3.8").build();
```

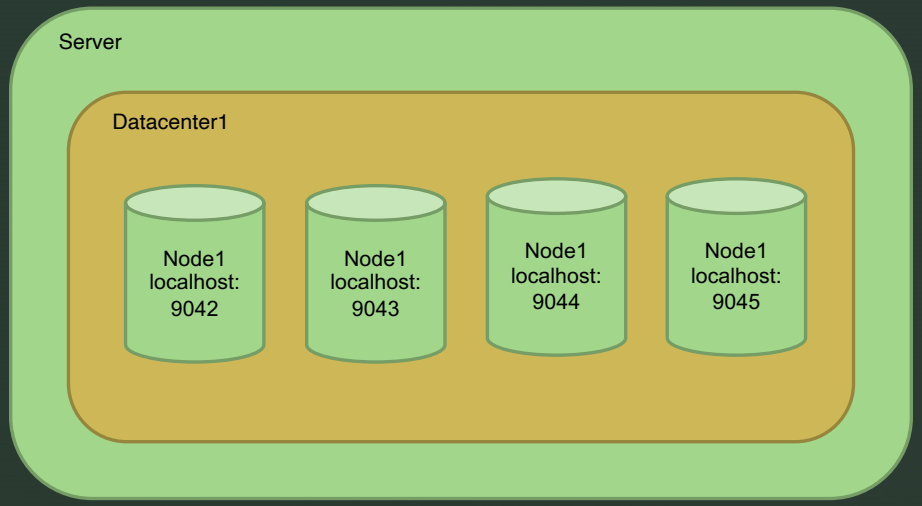
```
// 5 - Add nodes to the the datacenter
dataCenter.addNode().withAddress(addressResolver.get()).withName("Defa
ult").build();
```

```
// 6 - Register the datacenter on the server and start it
var boundCluster = server.register(clusterSpec);
boundCluster.start();
```

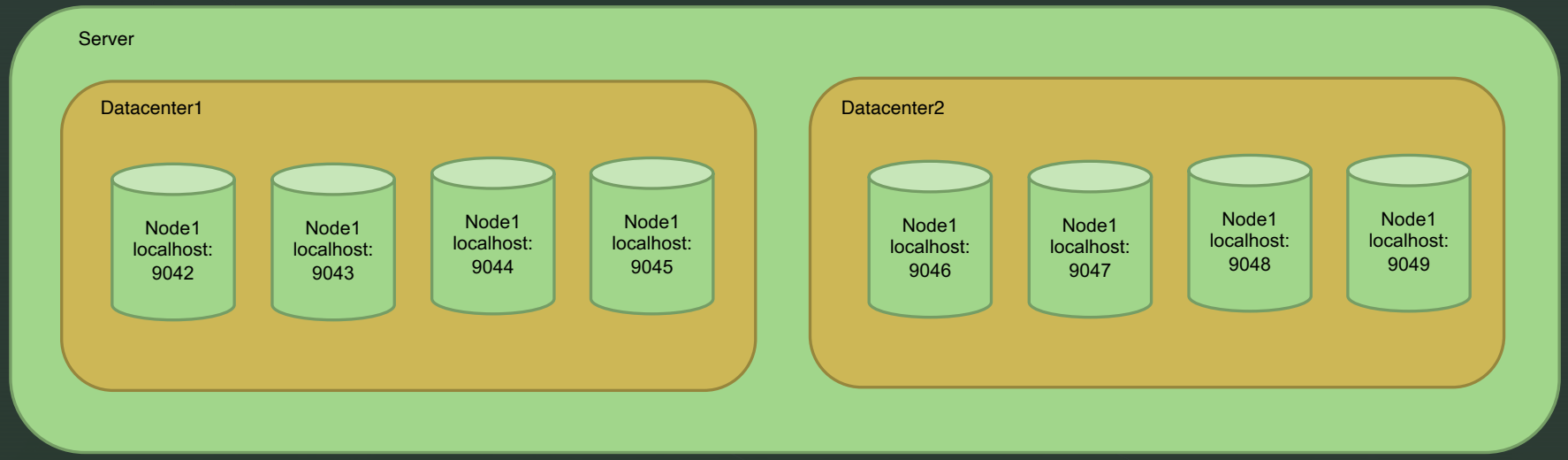
Creating a cluster

```
var server = Server.builder().build();

var clusterSpec = ClusterSpec.builder().build();
var addressResolver = new Inet4Resolver(defaultStartingIp,
CASSANDRA_PORT);
var dataCenter = clusterSpec.addDataCenter()
        .withName("datacenter1").withCassandraVersion("3.8").build();

dataCenter.addNode().withAddress(addressResolver.get()).withName("Default").
build();
var boundCluster = server.register(clusterSpec);

log.info("Starting Simulacron node");
boundCluster.start();
```

Cluster representation



Cluster multi datacenter representation



Priming requests

- *BoundCluster.prime()* is used to prime queries
- PrimeDsl is used to create the query structure and mocked behavior
  - when(): when we receive a specific query
  - then(): what simulacron does

- Common method to prime queries:
- *readTimeout(),* serverError(), syntaxError(), truncateError(), etc..
- RowBuilder to build success results

Demo

Exercises

Final thoughts

- Worry about error scenarios
- Worry about resilience scenarios
- Cover your application with tests