Example 1:
Creating a Cassandra Cluster

What will we learn in this example?

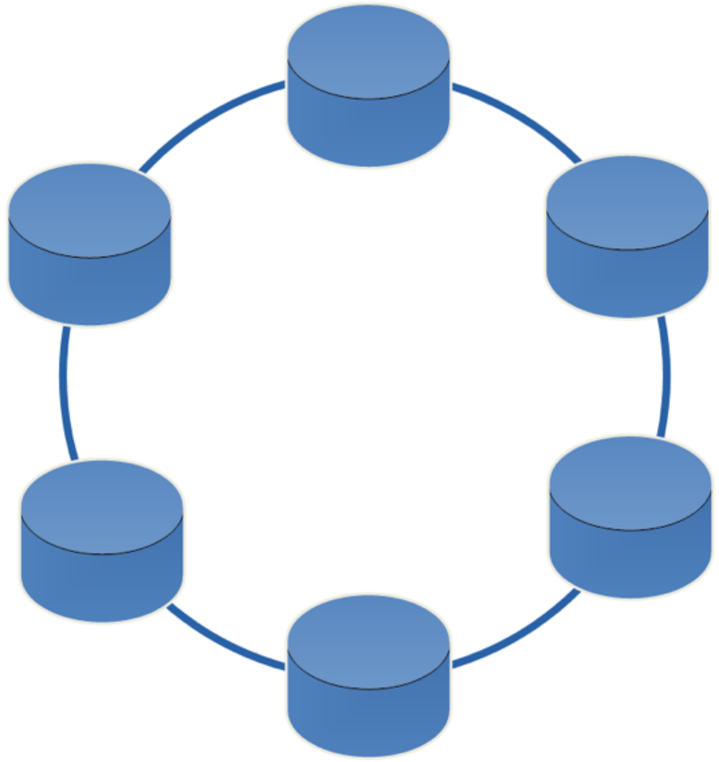
Bird's eye view of a cluster

Create a cassandra cluster on your local machine

Bird's eye view of a cluster

Cassandra is a distributed, decentralized database

one ring is called data centre

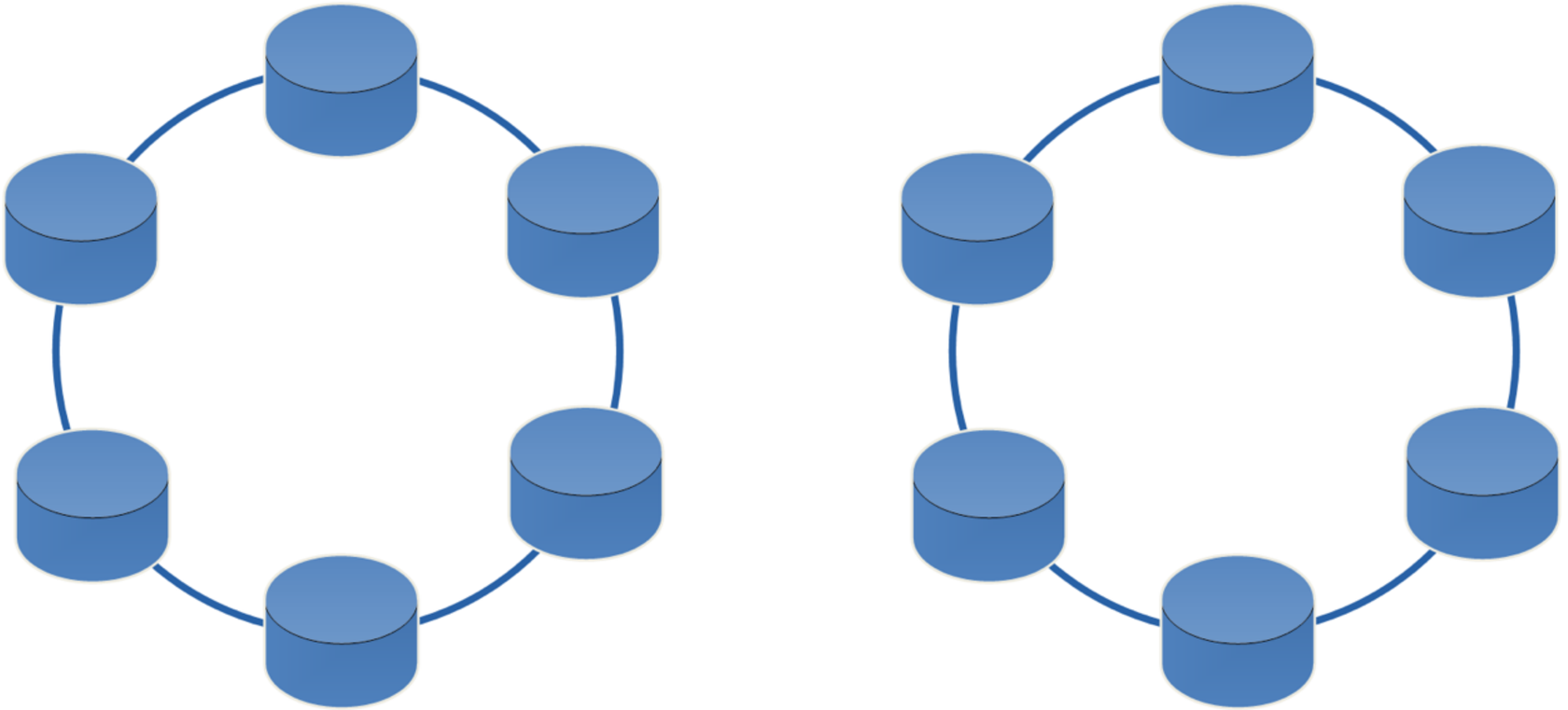


This cluster has 1 datacenter

Cassandra Cluster

Bird's eye view of a cluster

Multiple Datacenters

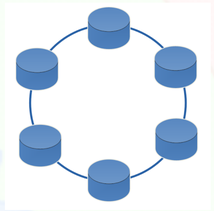


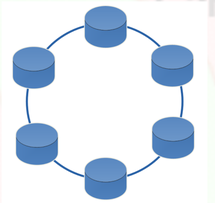
Datacenter1

Datacenter2

Bird's eye view of a cluster

datacenters can be at different geographical locations





But why would we do that?

Users connect to datacenters based on geographic location

This way we can ensure faster performance for end users.

What will we learn in this example?

~~Bird's eye view of a cluster~~

Create a cassandra cluster on your local machine

Create a cassandra cluster on your local machine

We will use the Cassandra Cluster Manager (ccm)

note that ccm cannot be used on production

Its made for 2 purposes

1. To learn cassandra
2. To test an application using cassandra on your local environment

Create a cassandra cluster on your local machine

`ccm` has a list of commands to manage the cluster

You can run `ccm help` to see the full list of commands

Create a cassandra cluster on your local machine

On its start, cassandra opens 3 ports on a node

9160	thrift based clients connect to cassandra using this port number
9042	binary protocol based clients connect to cassandra using this port number eg CQL
7000	Nodes communicate with each other using this port number

Create a cassandra cluster on your local machine

We create a cluster on 1 machine (local)

Create a network interface on loopback addresses (`127.0.0.*`)

In windows and unix, the network interface aliasing is done internally by ccm

For macOS, we need to it manually

Create a cassandra cluster on your local machine

If you have n nodes in your cluster
then run the following command for all n of them

```
sudo ifconfig lo0 alias 127.0.0.2
sudo ifconfig lo0 alias 127.0.0.3
.
.
sudo ifconfig lo0 alias 127.0.0.n
```

Run these commands BEFORE you run
the command to create your cluster

Create a cassandra cluster on your local machine

Let's create a cluster for "easybuy"

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

Command to create the cluster

Create a cassandra cluster on your local machine

Let's create a cluster for "easybuy"

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

Name of the cluster
to be created

Create a cassandra cluster on your local machine

Let's create a cluster for "easybuy"

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

Options for create command

You can see the full list of options for create command by running

```
ccm create help
```

Create a cassandra cluster on your local machine

Lets go through the options

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

version of
Cassandra installed

To know the version of Cassandra run the following command on the terminal

```
cassandra -v
```

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

this option
populates the cluster
with the number of
nodes passed

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

this option populates the cluster with the number of nodes passed

We can create the cluster even without this option

In that case we will have an empty cluster without any nodes

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

this option populates the cluster with the number of nodes passed

We can create the cluster even without this option

Run the following command to populate cluster with 5 nodes

```
ccm populate -n 5
```

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

this option starts the cassandra cluster

applications can now connect to the cluster

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

Again, the cluster can be created without this option

If you are not using this option then you start the cluster with the following command

```
ccm start
```

run the command after you have populated the cluster

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

we set the authenticator to PasswordAuthenticator

What is an authenticator?

Authenticator is the class that authenticates clients/users connecting to cassandra

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

default authenticator is
AllowAllAuthenticator

No user/password required

Anyone can connect to the cluster

This is not very secure - pretty obvious

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

We want that only the application and our team should be able to connect to the cluster

We can ensure this by using the PasswordAuthenticator

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

To connect to cassandra,
PasswordAuthenticator forces the user to
provide a valid username/password

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

default username/password for connecting to the cluster is "cassandra"

We can configure our own username/ password for our cluster

Create a cassandra cluster on your local machine

```
ccm create easybuy -v 3.7 -n 5 -s --pwd-auth
```

If the cluster is created successfully then the output is :

```
Current cluster is now: easybuy
```

What will we learn in this example?

~~Bird eye view of a cluster~~

~~Create a cassandra cluster on your local machine~~

Example 2:
Exploring CCM

Let's see the basic commands of CCM

Basic CCM commands

```
ccm list
```

This lists all the existing clusters

output

```
*easybuy
```

* denotes that this is the current cluster

Basic CCM commands

```
ccm status
```

This displays status of the nodes in the current cluster

output

```
Cluster: 'easybuy'
-------------------
node1: UP
node3: UP
node2: UP
node5: UP
node4: UP
```

UP - node is ready to take requests

DOWN - node is not ready to take requests

Basic CCM commands

```
ccm stop
```

Stops all the nodes in the cluster

After we have stopped the cluster, the output of the `ccm status` command will be:

```
Cluster: 'easybuy'
------------------
node1: DOWN
node3: DOWN
node2: DOWN
node5: DOWN
node4: DOWN
```

Basic CCM commands

```
ccm node1 show
```

Displays information of node named "node1"

output

```
node1: UP
       cluster=easybuy
       auto_bootstrap=False
       thrift=('127.0.0.1', 9160)
       binary=('127.0.0.1', 9042)
       storage=('127.0.0.1', 7000)
       jmx_port=7100
       remote_debug_port=0
       byteman_port=0
       initial_token=-9223372036854775808
       pid=3028
```

Basic CCM commands

Lets understand the fields of the output

```
ccm node1 show
```

```
node1: UP
        cluster=easybuy
        auto_bootstrap=False
        thrift=('127.0.0.1', 9160)
        binary=('127.0.0.1', 9042)
        storage=('127.0.0.1', 7000)
        jmx_port=7100
        remote_debug_port=0
        byteman_port=0
        initial_token=-9223372036854775808
        pid=3028
```

Name of the node and the status

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
       cluster=easybuy
       auto_bootstrap=False
       thrift=('127.0.0.1', 9160)
       binary=('127.0.0.1', 9042)
       storage=('127.0.0.1', 7000)
       jmx_port=7100
       remote_debug_port=0
       byteman_port=0
       initial_token=-9223372036854775808
       pid=3028
```

node1 belongs to this cluster

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
        cluster=easybuy
        auto_bootstrap=False
        thrift=('127.0.0.1', 9160)
        binary=('127.0.0.1', 9042)
        storage=('127.0.0.1', 7000)
        jmx_port=7100
        remote_debug_port=0
        byteman_port=0
        initial_token=-9223372036854775808
        pid=3028
```

new node will automatically migrate data from the cluster if the flag is true

default value for the flag is false

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
        cluster=easybuy
        auto_bootstrap=False
        thrift=('127.0.0.1', 9160)
        binary=('127.0.0.1', 9042)
        storage=('127.0.0.1', 7000)
        jmx_port=7100
        remote_debug_port=0
        byteman_port=0
        initial_token=-9223372036854775808
        pid=3028
```

For a new cluster without any data, set the value False

If you are adding a node to a cluster with data

set the flag to true

ONLY after the new node is properly configured

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
        cluster=easybuy
        auto_bootstrap=False
        thrift=('127.0.0.1', 9160)
        binary=('127.0.0.1', 9042)
        storage=('127.0.0.1', 7000)
        jmx_port=7100
        remote_debug_port=0
        byteman_port=0
        initial_token=-9223372036854775808
        pid=3028
```

As we had seen before this port is used for connecting via thrift

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
       cluster=easybuy
       auto_bootstrap=False
       thrift=('127.0.0.1', 9160)
       binary=('127.0.0.1', 9042)
       storage=('127.0.0.1', 7000)
       jmx_port=7100
       remote_debug_port=0
       byteman_port=0
       initial_token=-9223372036854775808
       pid=3028
```

this port is used for connecting from cqlsh or any client using binary protocol

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
       cluster=easybuy
       auto_bootstrap=False
       thrift=('127.0.0.1', 9160)
       binary=('127.0.0.1', 9042)
       storage=('127.0.0.1', 7000)
       jmx_port=7100
       remote_debug_port=0
       byteman_port=0
       initial_token=-9223372036854775808
       pid=3028
```

this port is used for inter-node communication

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
       cluster=easybuy
       auto_bootstrap=False
       thrift=('127.0.0.1', 9160)
       binary=('127.0.0.1', 9042)
       storage=('127.0.0.1', 7000)
       jmx_port=7100
       remote_debug_port=0
       byteman_port=0
       initial_token=-9223372036854775808
       pid=3028
```

Port to monitor the cluster using java based monitoring tools e.g. jconsole

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
       cluster=easybuy
       auto_bootstrap=False
       thrift=('127.0.0.1', 9160)
       binary=('127.0.0.1', 9042)
       storage=('127.0.0.1', 7000)
       jmx_port=7100
       remote_debug_port=0
       byteman_port=0
       initial_token=-9223372036854775808
       pid=3028
```

Port for remote debugging from the application. By default it is not set

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
       cluster=easybuy
       auto_bootstrap=False
       thrift=('127.0.0.1', 9160)
       binary=('127.0.0.1', 9042)
       storage=('127.0.0.1', 7000)
       jmx_port=7100
       remote_debug_port=0
       byteman_port=0
       initial_token=-9223372036854775808
       pid=3028
```

Byteman tests/ BMUnits use this port for connecting to cassandra cluster

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
       cluster=easybuy
       auto_bootstrap=False
       thrift=('127.0.0.1', 9160)
       binary=('127.0.0.1', 9042)
       storage=('127.0.0.1', 7000)
       jmx_port=7100
       remote_debug_port=0
       byteman_port=0
       initial_token=-9223372036854775808
       pid=3028
```

hash-code of the first primary key value stored on this node.

we will understand tokens in detail later on when we talk about the Cassandra architecture

Basic CCM commands

```
ccm node1 show
```

```
node1: UP
       cluster=easybuy
       auto_bootstrap=False
       thrift=('127.0.0.1', 9160)
       binary=('127.0.0.1', 9042)
       storage=('127.0.0.1', 7000)
       jmx_port=7100
       remote_debug_port=0
       byteman_port=0
       initial_token=-9223372036854775808
       pid=3028
```

processId of cassandra on the node

Basic CCM commands

```
ccm remove
```

This destroys the current cluster and all data in the cluster is deleted permanently

Use this command cautiously

We saw the basic commands of CCM to:

- check the status of the cluster
- see the configuration on a node
- start the cluster
- stop the cluster

These should be enough to get us started with using Cassandra