

Criando meu primeiro cluster de Cassandra

DBA BRASIL - Data & Cloud 2023

Adriano Bonacin

Apresentação

- DBA há 10+ anos
- Oracle OCP DBA + PL/SQL
- Apache Cassandra Dev + Admin + K8ssandra
- AWS Dev/Sol Arch/SysOps Associate + Bigdata/Database Specialty

Diamante



AGGRANDIZE

COMMVault



TD SYNnex

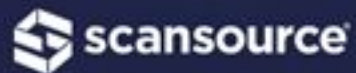
Platina



GOLDENGATEBR
Consultoria & Treinamentos

DISCOVER

Ouro



scansource

VERTICA
by opentext

Prata

TRACES



CAFÉ
COM
CLOUD

ROX

We take care
of your data

Apoio

FIAP

GRUPO
POSEIDON
DIGITAL

Carreira

- Pagseguro – DBA
- Semantix – Especialista BigData
- Agility – Consultor BigData
- Pagseguro – DBA
- Pythian – Cassandra Database Consultant
- Blog – yadax.com.br

Pythian

Love your data.

Transform your business with the power of data, analytics, and cloud.

- 25+ anos – Global IT Services
- Database experts
- Data & Analytics experts
- Profissionais em mais de 25 países

- Estamos com vagas abertas (muitas) para experts
 - Trabalhe remotamente
 - Muitos benefícios para sua carreira (certificação, treinamentos, ...)

Agenda

- Overview NoSQL
- Overview Cassandra
- Instalando Cassandra
- Arquivos de configuração
- Criando nosso cluster
- Arquivos de log

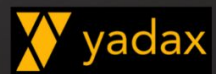
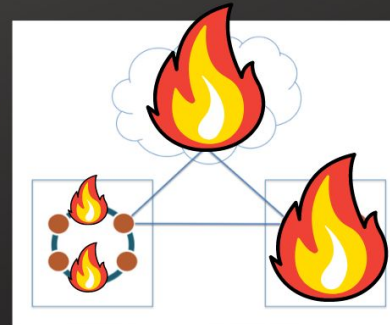
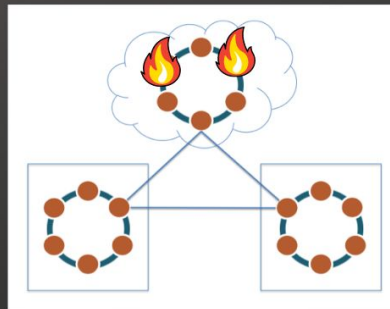
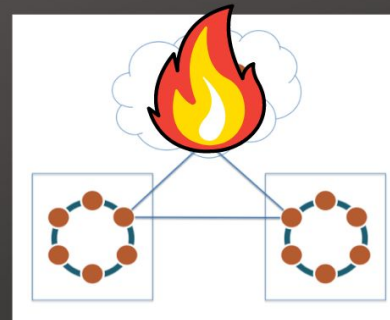
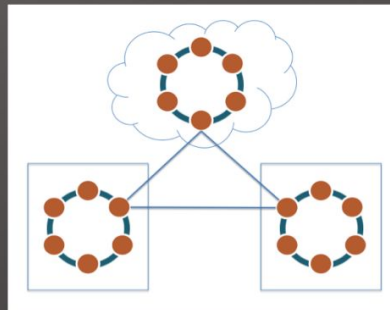
NoSQL

NoSQL

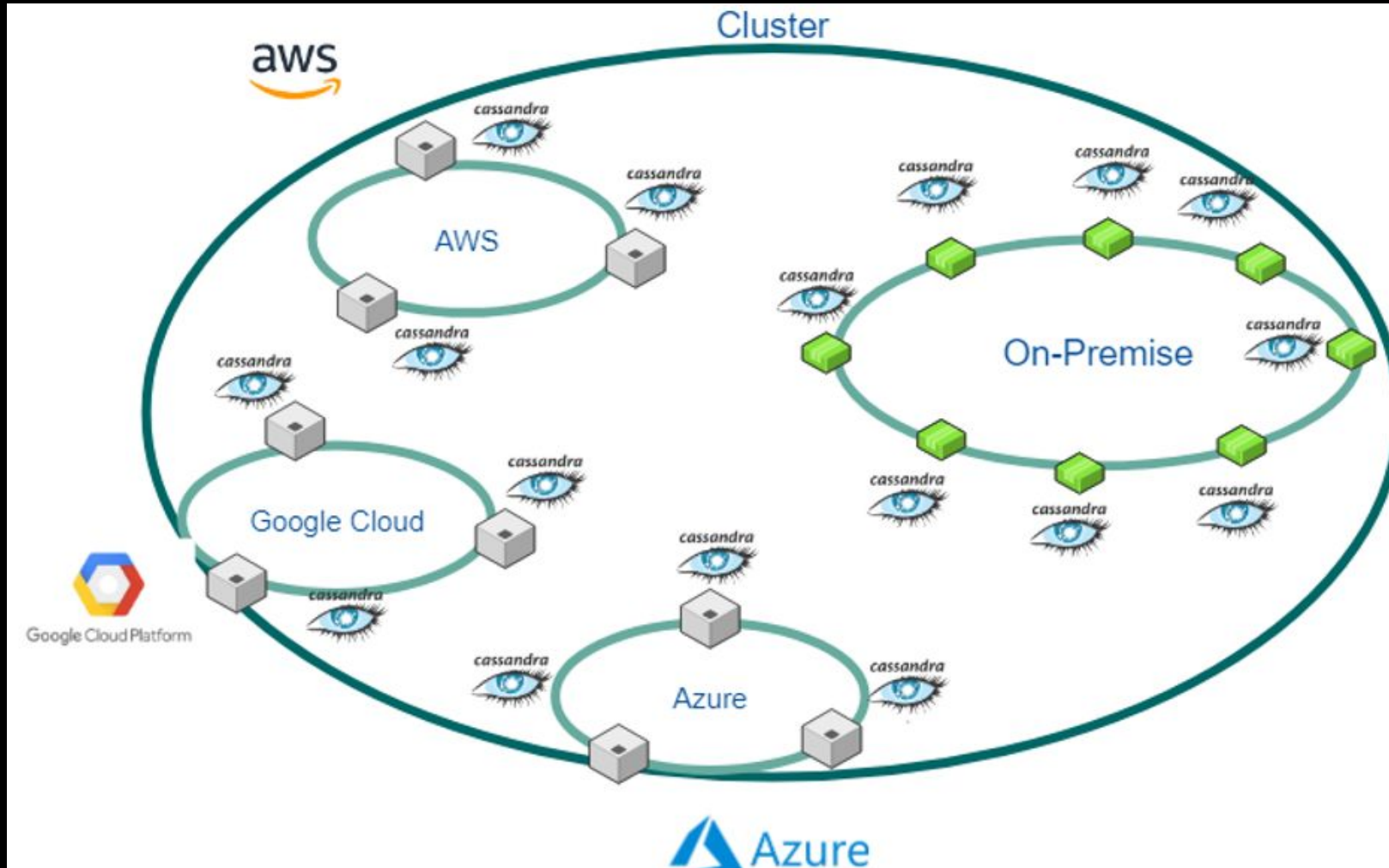
- Alta disponibilidade
- Escalabilidade
- Distribuídos
- Alto volume de dados
- Schema!?
- Normalização!?
- Relacionamento!?
- Substitui o RDBMS?



Alta disponibilidad



Distribuído



Escalabilidad

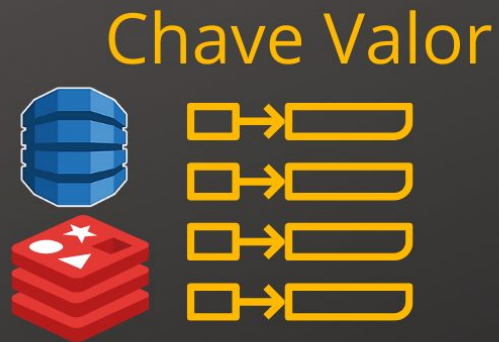
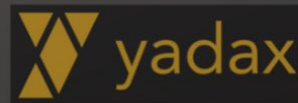
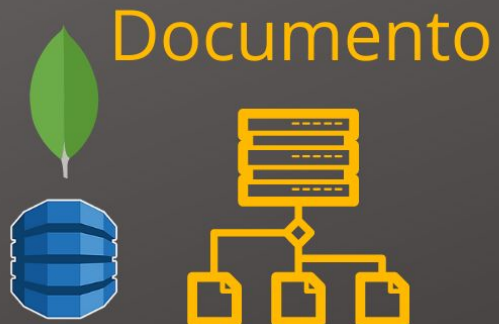


Vertical

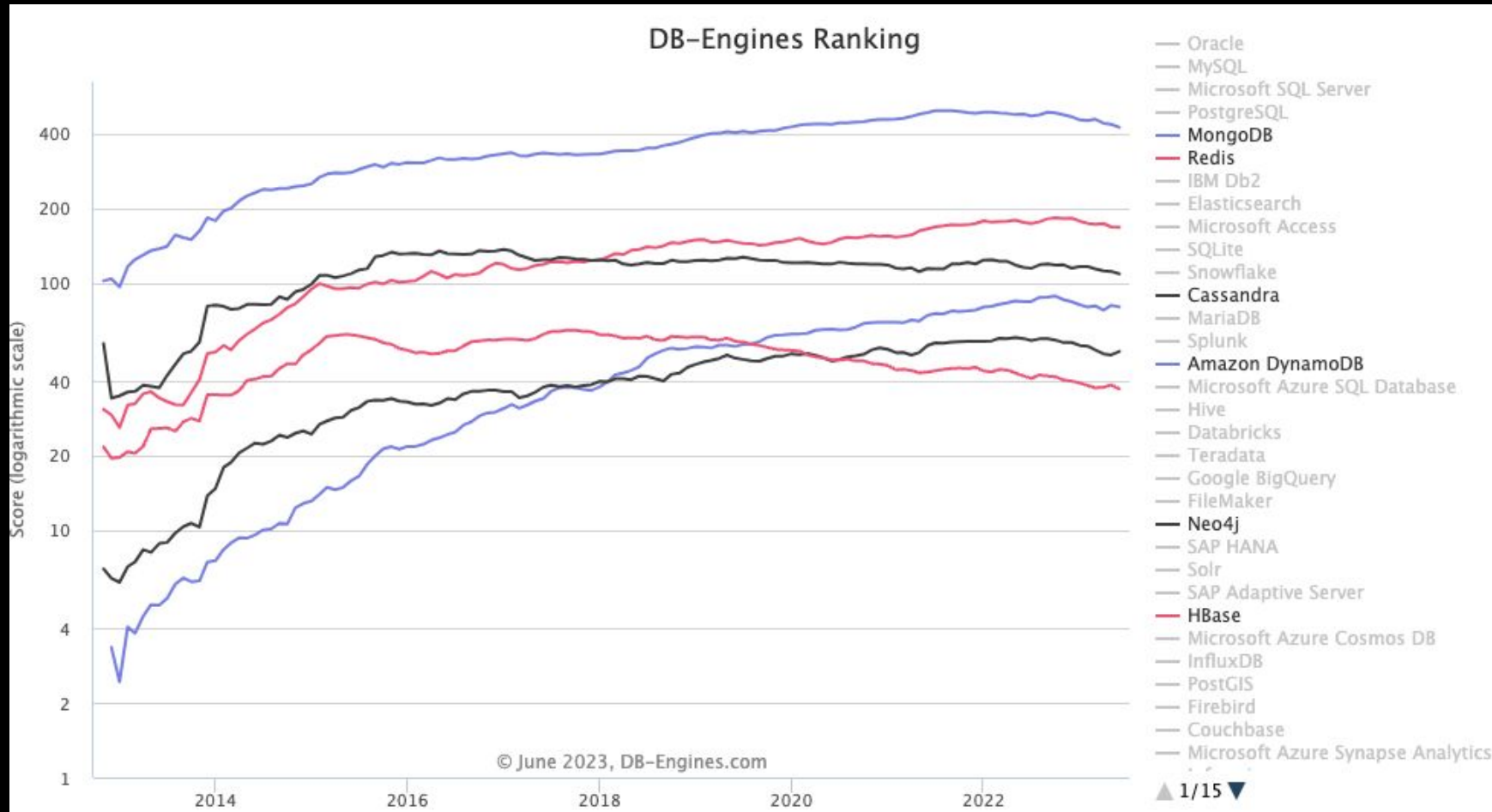


Horizontal

Tipos de NoSQL



Ranking DB-Engine



Cassandra

O que é Cassandra

- Projeto open source
- Distribuído - MultiRegion
- Híbrido: Cloud <----> OnPrem
- Tipo Wide-column, mas não colunar
- Consistência ajustável
- Multi Master
- Sem ponto único de falha
- Escalabilidade linear (out/in)
- Replicação ajustável



Gossip Protocol

O Cassandra utiliza o Gossip Protocol

A cada segundo, cada node fala com outros nodes

Trocam informações sobre status e outros membros

Não há nodes pré-determinados

Quando não usar

- Queries não usando PK (Secondary Indexes, Mview)
- Agregação
- Join
- Locks
- Updates/Deletes
- Transação
- Consistência forte

Quando usar

- Escrita maior que Leitura
- Dados raramente atualizados
- Leitura apenas pela PK
- Dados particionados
- Sem Join
- Precisa trabalhar globalmente
- TTL

Modelo wide-column*

- Baseado no BigTable do Google
- Suporta grande volume de requisições R/W
- VLDB
- Dados replicados



Wide-column stores versus columnar databases [\[edit \]](#)

Wide-column stores such as [Bigtable](#) and [Apache Cassandra](#) are not [column stores](#) in the original sense of the term, since their two-level structures do not use a columnar data layout. In genuine column stores, a columnar data layout is adopted such that each column is stored separately on disk. Wide-column stores do often support the notion of [column families](#) that are stored separately. However, each such column family typically contains multiple columns that are used together, similar to traditional relational database tables. Within a given column family, all data is stored in a row-by-row fashion, such that the columns for a given row are stored together, rather than each column being stored separately.

Wide-column stores that support column families are also known as *column family databases*.^{[\[citation needed\]](#)}

Dados Replicados

Node 1

Keyspace Loja

Tabela Venda por Clientes

ID: 10212	Venda 1	Venda 3	ID: 10332	Venda 1	Venda 3
	Venda 2			Venda 2	Venda 4

Node 2

Keyspace Loja

Tabela Venda por Clientes

ID: 10212	Venda 1	Venda 3
	Venda 2	

ID: 3291					

Node 3

Keyspace Loja

Tabela Venda por Clientes

ID: 10332	Venda 1	Venda 3
	Venda 2	Venda 4

ID: 3291					

Instalação no Rocky Linux 8.7

Update SO

```
[rocky@node1 ~]$ sudo dnf update -y
```

```
Rocky Linux 8 - AppStream      14 MB/s | 9.1 MB      00:00  
Rocky Linux 8 - BaseOS        4.4 MB/s | 2.8 MB      00:00  
Rocky Linux 8 - Extras        31 kB/s | 13 kB       00:00
```

```
Dependencies resolved.
```

```
=====
```

```
...
```

Versões do Apache Cassandra

Index of cassandra-rpm

Name	Last Modified
21x/	26-04-21 16:31:25
22x/	26-04-21 16:25:43
30x/	26-04-21 16:19:34
311x/	26-04-21 16:40:04
40x/	26-04-21 16:07:07
41x/	19-06-22 19:13:31

Vamos instalar o Apache Cassandra 4.1

Pré-Requisitos

JAVA

Python 3.6 + (Para o client, cqlsh)

Instalar o Java

```
[rocky@node1 ~]$ sudo yum install -y java-11-openjdk
```

```
Dependencies resolved.
```

```
=====
```

Package	Arch	Version	...
---------	------	---------	-----

```
=====
```

```
Installing:
```

```
java-11-openjdk      x86_64      1:11.0.19.0.7-4.e18
```

```
...
```

```
Complete!
```

```
[rocky@node1 ~]$
```

Conferindo a instalação

```
[rocky@node1 ~]$ java -version
openjdk version "11.0.19" 2023-04-18 LTS
OpenJDK Runtime Environment
(Red_Hat-11.0.19.0.7-2) (build 11.0.19+7-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-11.0.19.0.7-2)
(build 11.0.19+7-LTS, mixed mode, sharing)
```

Instalando o Python

```
[rocky@node1 ~]$ sudo yum install -y python3 python3-pip
Dependencies resolved.
```

```
=====
Package                Arch          Version      ...
=====
Installing:
python3-pip            noarch        9.0.3-22.el8.rocky.0
python36               x86_64        3.6.8-38.module+el8.5.0+671+
```

Conferindo a versão do Python

```
[rocky@node1 ~]$ python3 -V
```

```
Python 3.6.8
```

Repositório do Cassandra

Editar esse arquivo: `/etc/yum.repos.d/cassandra.repo`

21x

22x

30x

311x

40x

41x

Repositório do Cassandra

```
[rocky@node1 ~]$ sudo vi /etc/yum.repos.d/cassandra.repo
[cassandra]
name=Apache Cassandra
baseurl=https://redhat.cassandra.apache.org/41x/
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://downloads.apache.org/cassandra/KEYS
```

Instalação do Apache Cassandra

```
[rocky@node1 ~]$ sudo dnf install -y cassandra
```

```
Dependencies resolved.
```

```
=====
Package                Architecture          Version              ...
=====
```

Package	Architecture	Version	...
Installing:			
cassandra	noarch	4.1.2-1	...

```
...
```

```
Installed:
```

```
  cassandra-4.1.2-1.noarch
```

```
Complete!
```

```
[rocky@node1 ~]$
```

Conferindo o pacote instalado

```
[rocky@node1 ~]$ rpm -qa | grep cassandra  
cassandra-4.1.2-1.noarch
```


Conferindo o status do serviço do cassandra

```
[rocky@node1 ~]$ sudo service cassandra status
```

```
Unit cassandra.service could not be found.
```

```
[rocky@node1 ~]$ sudo systemctl daemon-reload
```

```
[rocky@node1 ~]$ sudo service cassandra status
```

```
● cassandra.service - LSB: distributed storage system for structured data
```

```
Loaded: loaded (/etc/rc.d/init.d/cassandra; generated)
```

```
Active: inactive (dead)
```

```
Docs: man:systemd-sysv-generator(8)
```

Portas LISTEN antes do Cassandra

```
[rocky@node1 ~]$ ss -nltp
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
LISTEN	0	128	0.0.0.0:111	0.0.0.0:*
LISTEN	0	128	0.0.0.0:22	0.0.0.0:*
LISTEN	0	128	:::111	:::*
LISTEN	0	128	:::22	:::*

Startando o Cassandra

```
[rocky@node1 ~]$ sudo service cassandra start
```

```
Starting cassandra (via systemctl):
```

```
[ OK ]
```

```
[rocky@node1 ~]$
```

Portas LISTEN após o Cassandra

```
[rocky@node1 ~]$ ss -nltp
```

State Process	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
LISTEN	0	128	0.0.0.0:111	0.0.0.0:*
LISTEN	0	128	0.0.0.0:22	0.0.0.0:*
LISTEN	0	2048	127.0.0.1:9042	0.0.0.0:*
LISTEN	0	512	127.0.0.1:7000	0.0.0.0:*
LISTEN	0	50	127.0.0.1:39321	0.0.0.0:*
LISTEN	0	50	127.0.0.1:7199	0.0.0.0:*
LISTEN	0	128	:::111	:::*
LISTEN	0	128	:::22	:::*

Conectando no Cassandra

```
[rocky@node1 ~]$ cqlsh
```

```
Connected to Test Cluster at 127.0.0.1:9042
```

```
[cqlsh 6.1.0 | Cassandra 4.1.2 | CQL spec 3.4.6  
| Native protocol v5]
```

```
Use HELP for help.
```

```
cqlsh>
```

Nodetool

Ferramenta de administração e monitoração
Fala na porta 7199 (JMX)

```
[rocky@node1 ~]$ nodetool status
```

```
Datacenter: datacenter1
```

```
=====
```

```
Status=Up/Down
```

```
|/ State=Normal/Leaving/Joining/Moving
```

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UN	127.0.0.1	104.34 KiB	16	100.0%	8ff6c6498c96ba4	rack1

Arquivos de Configuração

`/etc/cassandra/conf/cassandra.yaml` (principal)

`/etc/cassandra/conf/cassandra-env.sh`

`/etc/cassandra/conf/cassandra-rackdc.properties`

Diretório de dados

```
data_file_directories:  
  - /var/lib/cassandra
```


Limpendo tudo

```
[rocky@node1 ~]$ sudo service cassandra stop
```

```
[rocky@node1 ~]$
```

```
[rocky@node1 ~]$ rm -rf /var/lib/cassandra/*/*
```

```
[rocky@node1 ~]$
```

Em seguida vamos criar um cluster novinho.

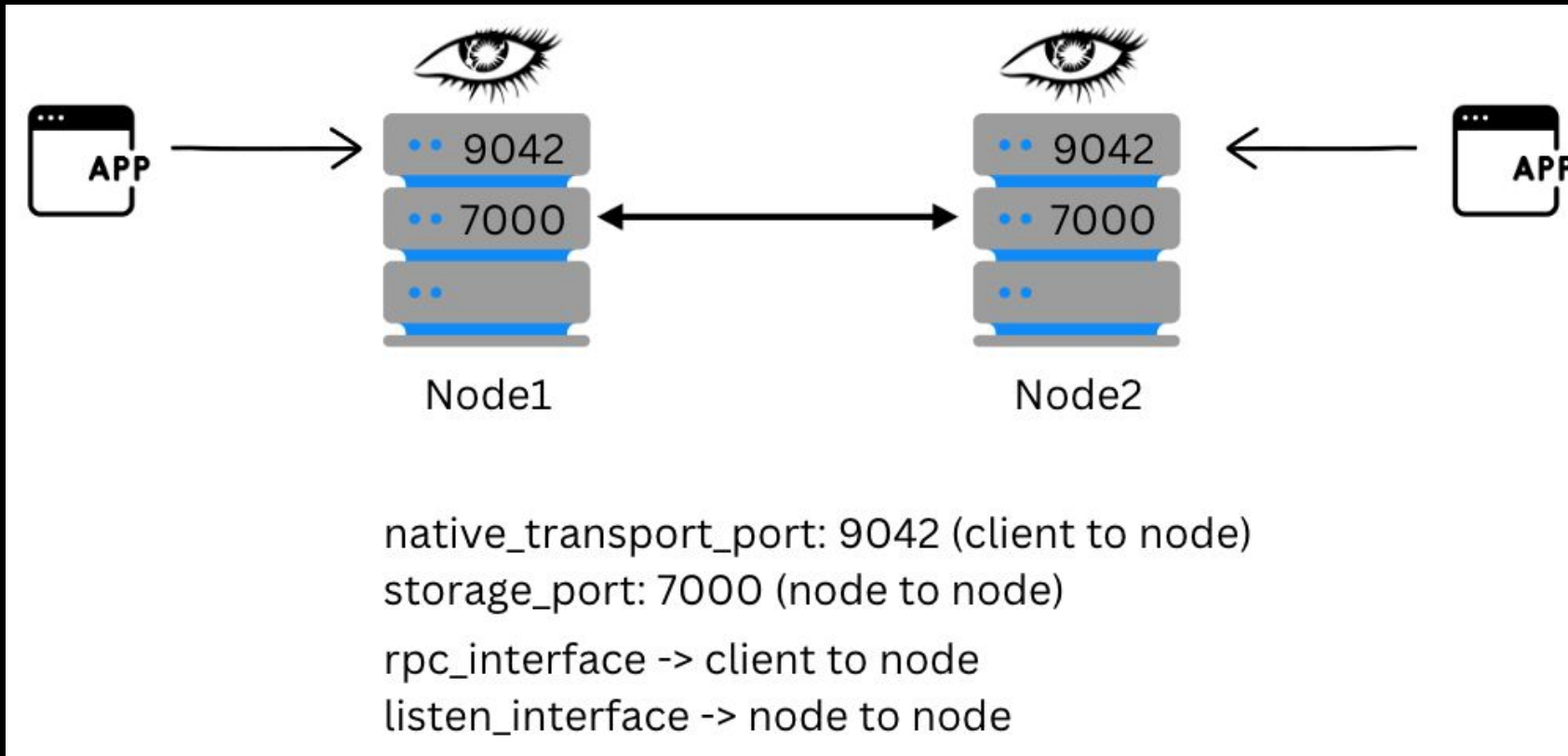
Nosso cluster

node1: 10.0.7.8

node2: 10.0.11.239

node3: 10.0.3.116

Comunicação no Cassandra



Comunicação no Cassandra

Porta 7000 precisa estar liberada entre os nodes

Porta 9042 precisa estar liberada entre node e app.

listen_address ou listen_interface

Controla a comunicação entre os nodes.

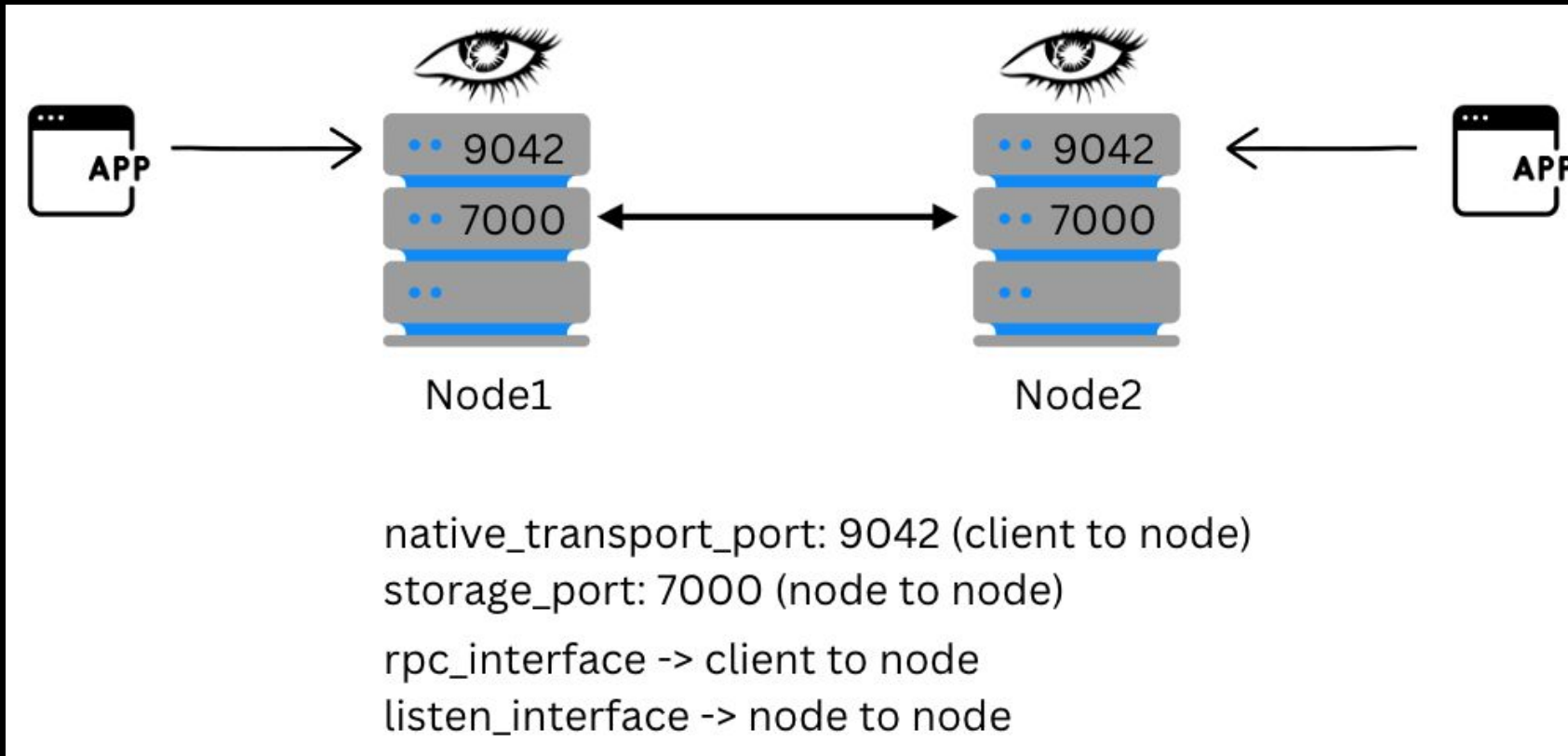
```
# listen_address: localhost  
listen_interface: eth0
```

rpc_address ou rpc_interface

Controla a comunicação entre nodes e app.

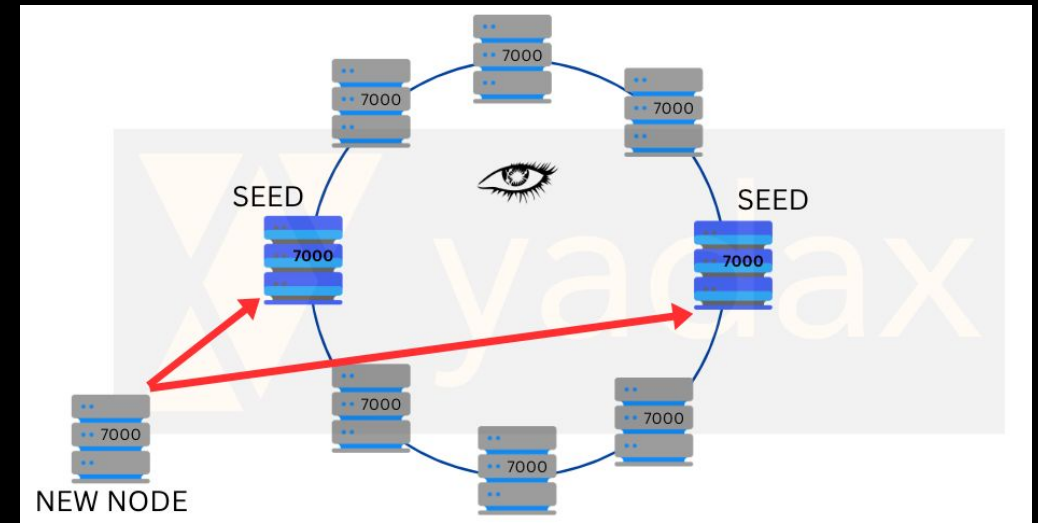
```
# rpc_address: localhost  
rpc_interface: eth0
```

Comunicação no Cassandra



Seeds

- Nodes especiais, mas nem tanto
- Controlam a entrada e saída de nodes
- Hardcode no cassandra.yaml
- Use dois ou três por DC



seed_provider:

- **class_name:** org.apache.cassandra.locator.SimpleSeedProvider

parameters:

- **seeds:** "10.0.7.8,10.0.11.239"

endpoint snitch

- O mais usado também é baseado no Protocolo Gossip
- Cada node sabe onde (DC + RACK) ele está
- Esta informação fica no arquivo:
`/etc/cassandra/conf/cassandra-rackdc.properties`
- Há algumas opções que usam a API da AWS para determinar a localização (Região: DC, AZ: RACK)

endpoint_snitch

```
endpoint_snitch: GossipingPropertyFileSnitch
```

```
[rocky@node1 ~]$ cat /etc/cassandra/conf/cassandra-rackdc.properties  
dc=dc1  
rack=rack1
```

```
[rocky@node2 ~]$ cat /etc/cassandra/conf/cassandra-rackdc.properties  
dc=dc1  
rack=rack2
```

```
[rocky@node3 ~]$ cat /etc/cassandra/conf/cassandra-rackdc.properties  
dc=dc2  
rack=rack1
```

cluster_name

Para os nodes entrarem em um cluster existente, ele precisa comunicar com um SEED

O cluster name do novo node precisa ser igual ao do cluster

```
cluster_name: 'Yadax'
```

Conferindo

```
[rocky@node1 ~]$ cat /etc/cassandra/conf/cassandra.yaml | egrep -v  
'^$|^#' | egrep  
'^cluster_name|^listen_interface|^rpc_interface|^storage_port|^ssl_st  
orage_port|^native_transport_port|^rpc_address|^listen_address|^seeds|^  
^endpoint_snitch'
```

```
cluster_name: 'Yadax'  
    # seeds is actually a comma-delimited list of addresses.  
    - seeds: "10.0.7.8,10.0.11.239"  
storage_port: 7000  
ssl_storage_port: 7001  
listen_interface: eth0  
native_transport_port: 9042  
rpc_interface: eth0  
endpoint_snitch: GossipingPropertyFileSnitch
```

Agora é só startar o cassandra

```
[rocky@node1 ~]$ sudo service cassandra start
```

```
Starting cassandra (via systemctl): [ OK ]
```

```
[rocky@node1 ~]$
```

Portas LISTEN

```
[rocky@node1 ~]$ ss -nltp
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
LISTEN	0	512	10.0.7.8:7000	0.0.0.0:*
LISTEN	0	50	127.0.0.1:7199	0.0.0.0:*
LISTEN	0	50	127.0.0.1:38255	0.0.0.0:*
LISTEN	0	128	0.0.0.0:111	0.0.0.0:*
LISTEN	0	2048	10.0.7.8:9042	0.0.0.0:*
LISTEN	0	128	0.0.0.0:22	0.0.0.0:*
LISTEN	0	128	:::111	:::*
LISTEN	0	128	:::22	:::*

```
[rocky@node1 ~]$
```

O que muda?

```
[rocky@node1 ~]$ cqlsh
```

```
Connection error: ('Unable to connect to any  
servers', {'127.0.0.1:9042':
```

```
ConnectionRefusedError(111, "Tried connecting to  
[('127.0.0.1', 9042)]. Last error: Connection  
refused"))
```

```
[rocky@node1 ~]$
```

Portas LISTEN

```
[rocky@node1 ~]$ ss -nltp
```

State	Local Address:Port	Peer Address:Port
LISTEN	10.0.7.8:7000	0.0.0.0:*
LISTEN	10.0.7.8:9042	0.0.0.0:*

O que muda?

```
[rocky@node1 ~]$ cqlsh 10.0.7.8
```

```
Connected to Yadax at 10.0.7.8:9042
```

```
[cqlsh 6.1.0 | Cassandra 4.1.2 | CQL spec 3.4.6 |  
Native protocol v5]
```

```
Use HELP for help.
```

```
cqlsh>
```

Nodetool status

```
[rocky@node1 ~]$ nodetool status
```

```
Datacenter: dc1
```

```
=====
```

```
Status=Up/Down
```

```
|/ State=Normal/Leaving/Joining/Moving
```

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UN	10.0.7.8	104.31 KiB	16	100.0%	ad9c89b7	rack1

Startar o cassandra no node2

```
[rocky@node2 ~]$ sudo service cassandra start
```

```
Starting cassandra (via systemctl):
```

```
[ OK ]
```

```
[rocky@node2 ~]$
```

Bootstrap

O novo node precisa receber os dados que passa a ser responsável

Nodetool status

```
[rocky@node1 ~]$ nodetool status
```

```
Datacenter: dc1
```

```
=====
```

```
Status=Up/Down
```

```
|/ State=Normal/Leaving/Joining/Moving
```

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UN	10.0.11.239	70.21 KiB	16	100.0%	10aed30d9a3	rack2
UN	10.0.7.8	125.39 KiB	16	100.0%	cab5eaa29b7	rack1

```
[rocky@node1 ~]$
```

Startar o cassandra no node3

```
[rocky@node3 ~]$ sudo service cassandra start
```

```
Starting cassandra (via systemctl):
```

```
[ OK ]
```

```
[rocky@node3 ~]$
```

Nodetool status

```
[rocky@node1 ~]$ nodetool status
```

```
Datacenter: dc1
```

```
=====
```

```
Status=Up/Down
```

```
|/ State=Normal/Leaving/Joining/Moving
```

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UN	10.0.11.239	70.21 KiB	16	100.0%	10aed30d9a3	rack2
UN	10.0.7.8	125.39 KiB	16	100.0%	cab5eaa29b7	rack1

```
Datacenter: dc2
```

```
=====
```

```
Status=Up/Down
```

```
|/ State=Normal/Leaving/Joining/Moving
```

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UJ	10.0.3.116	30.42 KiB	16	?	dfceb2bdf8	rack1

Nodetool status

```
[rocky@node1 ~]$ nodetool status
```

```
Datacenter: dc1
```

```
=====
```

```
Status=Up/Down
```

```
|/ State=Normal/Leaving/Joining/Moving
```

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UN	10.0.11.239	70.21 KiB	16	56.9%	10aed899	rack2
UN	10.0.7.8	125.39 KiB	16	67.2%	cab5eaa2	rack1

```
Datacenter: dc2
```

```
=====
```

```
Status=Up/Down
```

```
|/ State=Normal/Leaving/Joining/Moving
```

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UN	10.0.3.116	70.23 KiB	16	76.0%	dfceb2bd	rack1

Cleanup

Os nodes antigos que deixaram de ser responsáveis por parte dos dados, precisam ter esses dados limpos manualmente

```
[rocky@node1 ~]$ nodetool cleanup  
[rocky@node1 ~]$
```

Dúvidas!?

Diamante



Platina



DISCOVER

Ouro



VERTICA
by opentext™

Prata

TRACES



Apoio

FIAP



Obrigado