



SLOUPCOVÉ DATABÁZE

Lukáš Matějů

6.5.2024 | DPB

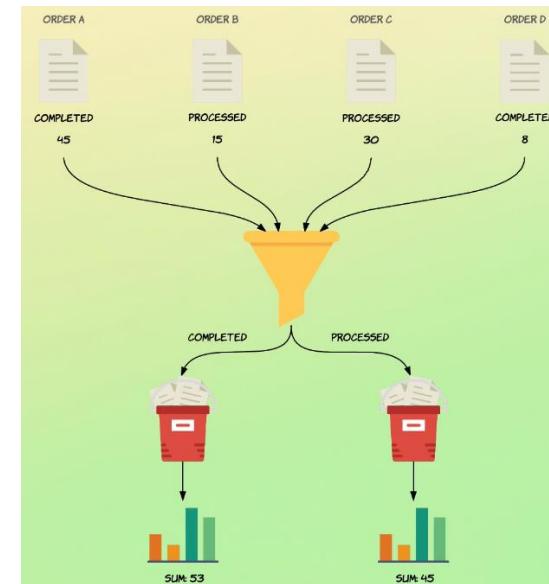


ČÁST I.: OPAKOVÁNÍ



OPAKOVÁNÍ

- agregace a Elasticsearch
- způsob jak seskupit data a jak z nich získat statistiky a závěry
- definované nad dokumenty
- dva základní typy
 - agregace metrik (metric aggregation)
 - počítají statistiky přes data
 - min, max, avg, sum, ...
 - bucket agregace
 - vytváří skupiny (buckets) dokumentů
 - každý bucket má kritéria, která rozhodují, jestli do nich dokument spadá
 - umožňuje vnořené agregace
 - agregace termínů (term), filter agregace
 - agregace rozsahů (range), ...



<https://www.udemy.com/course/elasticsearch-complete-guide/>

OPAKOVÁNÍ

- agregace a Elasticsearch

```
18 GET /order/_search ▶ 🔍  
19 {  
20   "size": 0,  
21   "aggs": {  
22     "amount_stats": {  
23       "stats": {  
24         "field": "total_amount"  
25       }  
26     }  
27   }  
28 }
```

```
18 "aggregations": {  
19   "amount_stats": {  
20     "count": 1000,  
21     "min": 10.27,  
22     "max": 281.77,  
23     "avg": 109.20961,  
24     "sum": 109209.61  
25   }  
26 }  
27 }  
28 }
```

```
1 GET /order/_search ▶ 🔍  
2 {  
3   "size": 0,  
4   "aggs": {  
5     "status_terms": {  
6       "terms": {  
7         "field": "status",  
8         "missing": "N/A",  
9         "order": {  
10           "_key": "asc"  
11         }  
12       }  
13     }  
14   }  
15 }
```

```
1 GET /order/_search ▶ 🔍  
2 {  
3   "size": 0,  
4   "aggs": {  
5     "status_terms": {  
6       "terms": {  
7         "field": "status"  
8       },  
9       "aggs": {  
10         "status_stats": {  
11           "stats": {  
12             "field": "total_amount"  
13           }  
14         }  
15       }  
16     }  
17   }  
18 }  
19 }
```

```
1 GET /order/_search ▶ 🔍  
2 {  
3   "size": 0,  
4   "aggs": {  
5     "total_salesmen": {  
6       "cardinality": {  
7         "field": "salesman.id"  
8       }  
9     }  
10   }  
11 }
```

```
1 GET /order/_search ▶ 🔍  
2 {  
3   "size": 0,  
4   "aggs": {  
5     "purchased_ranges": {  
6       "date_range": {  
7         "field": "purchased_at",  
8         "ranges": [  
9           {  
10             "from": "2016-01-01",  
11             "to": "2016-01-01||+6M"  
12           },  
13           {  
14             "from": "2016-01-01||+6M",  
15             "to": "2016-01-01||+1y"  
16           }  
17         ]  
18       }  
19     }  
20   }  
21 }
```

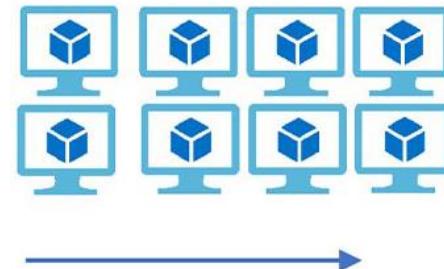


OPAKOVÁNÍ

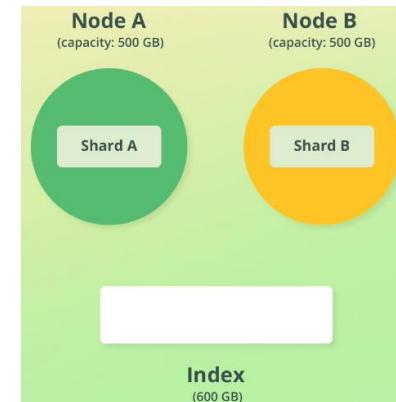
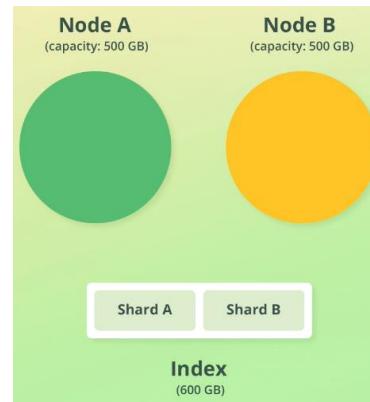
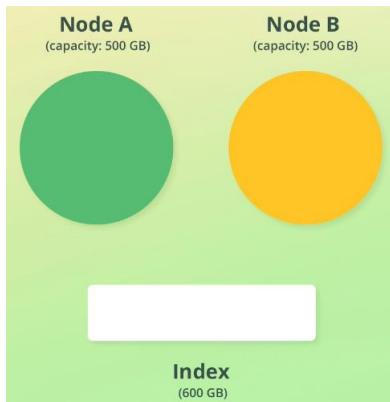
- **sharding**
 - rozdělení indexů na menší části
 - každá část je nazývána shard
 - prováděno na úrovni indexů
 - ne uzlu nebo clusteru
 - horizontální škálování
 - škálování datového úložiště

Horizontal Scaling

(Add more instances)



<https://www.webairy.com/horizontal-and-vertical-scaling/>



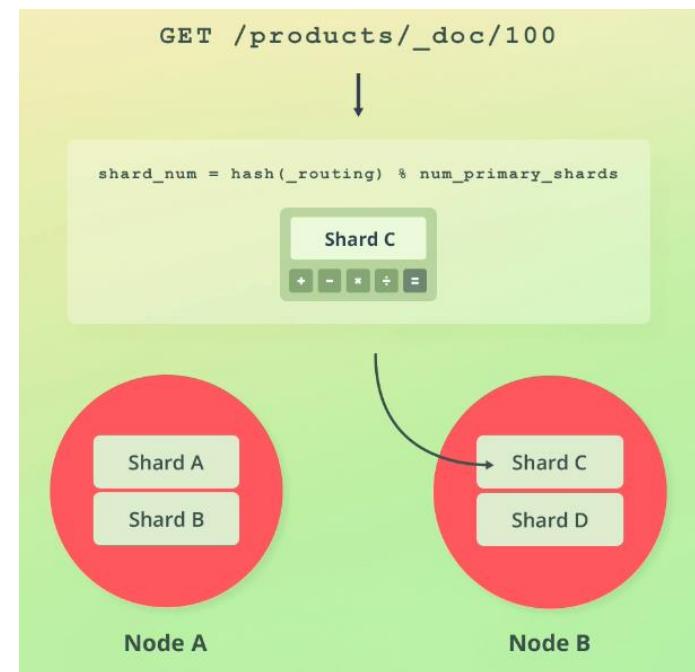
<https://www.udemy.com/course/elasticsearch-complete-guide/>



OPAKOVÁNÍ

- **routing**
 - proces přiřazující dokumentu shard
 - určuje, kde bude dokument uložen
 - na jakém shardu
 - udává, jak jsou dokumenty nalezeny po zaindexování
 - pro operace čtení, aktualizace nebo smazání
- Elasticsearch využívá jednoduchý vzorec

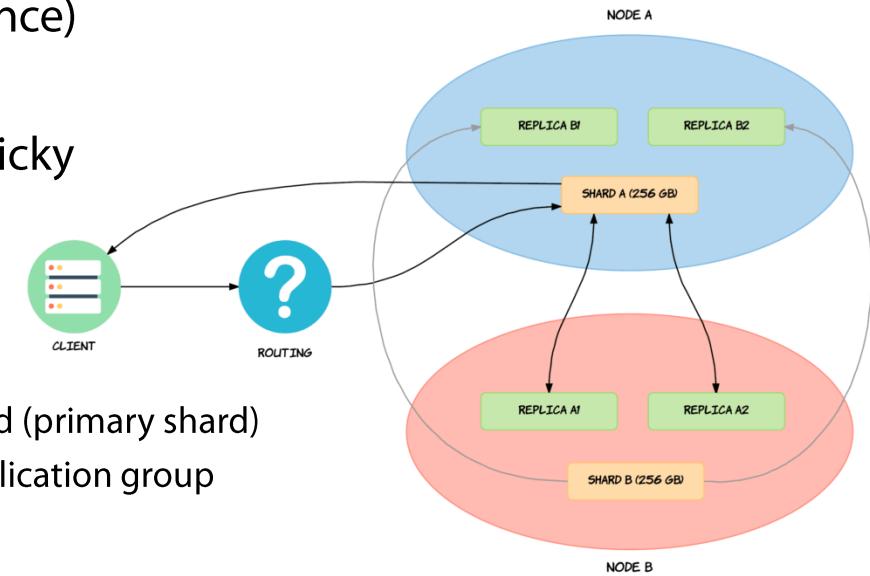
```
shard_num = hash(_routing) % num_primary_shards
```
- zajišťuje rovnoměrné rozmístění dokumentů mezi shardy



<https://www.udemy.com/course/elasticsearch-complete-guide/>

OPAKOVÁNÍ

- replikace
 - automatická distribuce změn v originále do jeho kopií
 - ochrana proti chybám (fault tolerance)
 - selhání hardware
 - nativní podpora, zapnuta automaticky
 - definována na úrovni indexů
 - vytváří úplné kopie shardů
 - repliky, repliky shardů (replica shards)
 - původní shard se nazývá primární shard (primary shard)
 - primary shard a replica shards tvoří replication group
 - mohou plně obsluhovat dotazy
 - pro synchronizaci využíván model primary-backup
 - může také zvyšovat propustnost dotazování

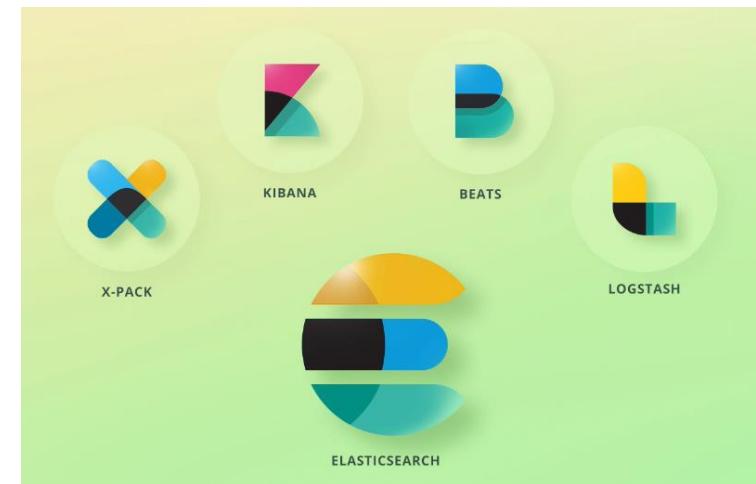


<https://codingexplained.com/coding/elasticsearch/understanding-replication-in-elasticsearch>



OPAKOVÁNÍ

- Elastic Stack
 - technologie vyvinuté a spravované Elastic BV
 - silná synergie mezi technologiemi
 - často používané společně s Elasticsearch
 - Kibana
 - platforma pro analýzu a vizualizaci dat
 - webový interface pro data v Elasticsearch
 - Logstash
 - roura (pipeline) na zpracování dat
 - X-Pack
 - balík funkcí do Elasticsearch a Kibana
 - Beats
 - nenároční agenti posílající data do Logstash / Elasticsearch instalovaní na servery



<https://www.udemy.com/course/elasticsearch-complete-guide/>



ČÁST II.: SLOUPCOVÉ DATABÁZE

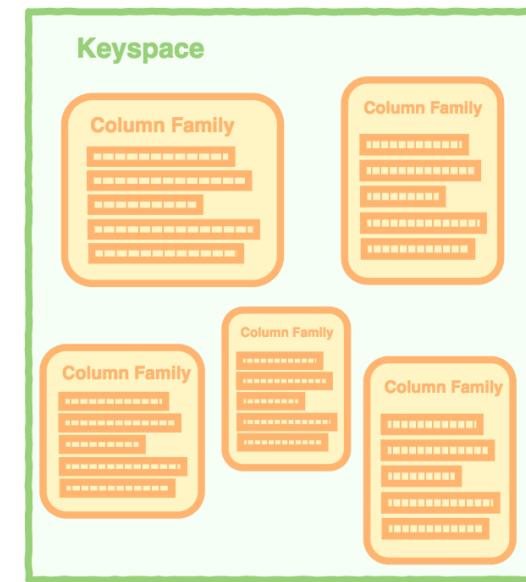
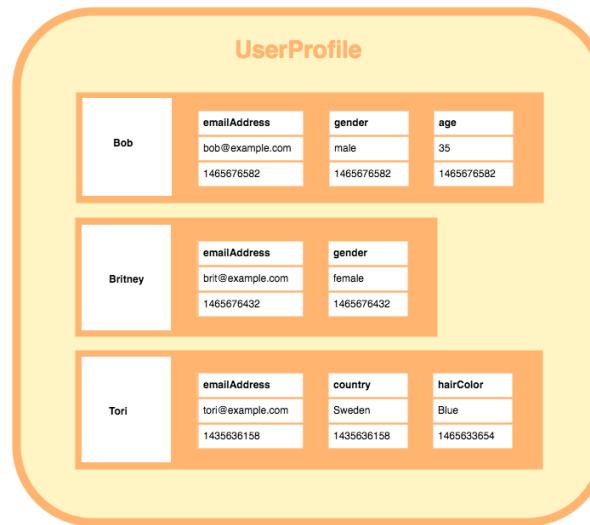


SLOUPCOVÉ DATABÁZE

- distribuované NoSQL databáze
- označovány také jako
 - sloupcově orientované databáze
- sloupcově orientovaný model pro ukládání dat
 - data ukládána do záznamů schopných pojmut v elké množství dynamických sloupců
 - klíče záznamů ani jména sloupců nejsou fixní
 - často považovány za dvoudimenzionální key-value úložiště
- optimalizované pro rychlý přístup k velkým objemům dat
 - analytické aplikace
 - big data

SLOUPCOVÉ DATABÁZE

- sloupcově orientovaný model
 - základem je **keyspace**
 - volně odpovídá schématu v relačních databázích
 - dává představu o struktuře databáze
 - **keyspace** obsahuje **column families**
 - volně odpovídají relacím v relačních databázích

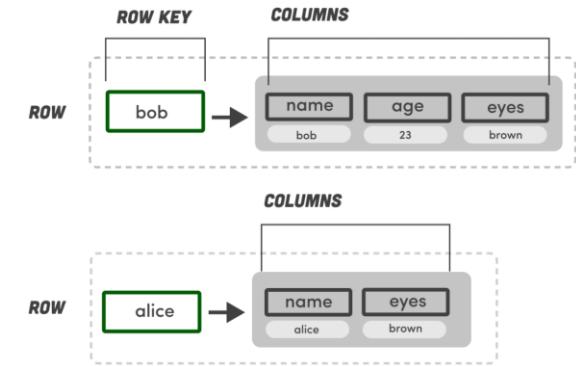


<https://database.guide/what-is-a-column-store-database/>

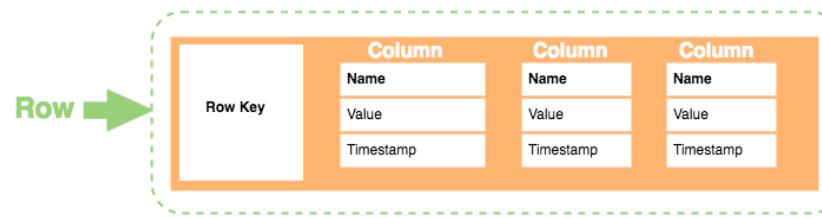


SLOUPCOVÉ DATABÁZE

- sloupcově orientovaný model
 - column family obsahuje seřazené řádky (rows)
 - jednoznačně identifikovány klíčem (row key)
 - každý řádek obsahuje sloupce (columns)
 - sloupce patří jen svému řádku
 - na rozdíl od relačních databází
 - počet sloupců se mezi řádky může lišit
 - stejně tak se mohou lišit názvy sloupců i jejich datové typy
 - každý sloupec obsahuje páry jméno – hodnota a časovou značku



<https://fireship.io/lessons/top-seven-database-paradigms/>



<https://database.guide/what-is-a-column-store-database/>



SLOUPCOVÉ DATABÁZE

- výhody sloupcových databází
 - horizontální škálování a replikace
 - volné schéma
 - rychlé načítání dat, dotazování a agregace, komprese dat
- hlavní použití
 - množství zápisů je výrazně vyšší než čtení, malé množství aktualizací
 - ukládání dat ze senzorů, časových řad
 - ukládání historických záznamů
 - čtení podle primárního klíče
 - sledování statusů, balíčků
- významnější zástupci
 - Cassandra, HBase, Azure Cosmos DB



VÝZNAMNĚJŠÍ ZÁSTUPCI

- Cassandra (od 2008)

- původně vyvinuta pro Facebook
- od Apache
- open-source NoSQL databáze
- v současnosti nejpopulárnější sloupcová databáze
- hlavní vlastnosti
 - distribuovaná databáze
 - horizontální škálování
 - replikace
 - odolnost vůči chybám
 - eventuálně konzistentní
 - laditelné
 - podpora Map-Reduce
 - dotazovací jazyk CQL
 - vysoký výkon



PROVEN

Cassandra is in use at Activision, Apple, BazaarVoice, Best Buy, CERN, Constant Contact, Comcast, eBay, Fidelity, Github, Hulu, ING, Instagram, Intuit, Macy's™, Macquarie Bank, Microsoft, McDonalds, Netflix, New York Times, Outbrain, Pearson Education, Sky, Spotify, Uber, Walmart, and thousands of other companies that have large, active data sets. In fact, Cassandra is used by 40% of the Fortune 100.

VÝZNAMNĚJŠÍ ZÁSTUPCI

- HBase (od 2008)
 - od Apache
 - postavená na konceptu Google BigTable
 - první sloupcově orientovaná databáze z roku 2005
 - dodává BigTable funkcionality pro Apache Hadoop
 - NoSQL distribuovaná sloupcová databáze
 - vlastnosti
 - velmi vysoká dostupnost
 - vysoká škálovatelnost
 - vysoký výkon



VÝZNAMNĚJŠÍ ZÁSTUPCI

- Azure Cosmos DB (od 2017)
 - globálně distribuovaná cloudová služba Microsoftu
 - více modelová databáze
 - API – SQL, MongoDB, Cassandra, Azure Table Storage, Gremlin a etcd
 - elastická škálovatelnost propustnosti
 - možnost volby konzistence
 - až pět úrovní
 - automaticky indexovaná všechna data
 - terminologie – řádek -> item



MARS



CHIPOTLE

asos

Symantec

ONLINE KURZY

- Cassandra

<https://www.udemy.com/course/apache-cassandra/>

<https://www.udemy.com/course/learn-apache-cassandra-from-scratch/>

- HBase

<https://www.udemy.com/course/apache-hbase-course/>

- Azure Cosmos DB

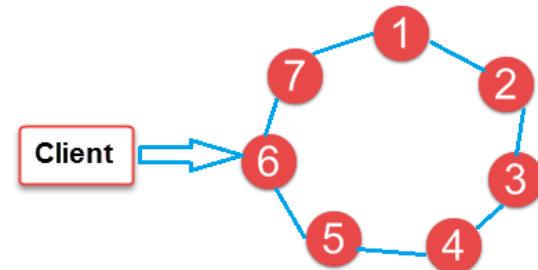
<https://www.udemy.com/course/azure-cosmosdb/>

ČÁST III.: CASSANDRA



CASSANDRA

- sloupcová NoSQL databáze (od 2008)
 - od Apache
 - open source
 - napsaná v Javě
 - založená na Amazon Dynamo a Google BigTable
 - distribuovaná
 - horizontální (lineární) škálování, replikace
 - technologie pro big data
 - vysoká dostupnost
 - velmi často na velkém množství uzlů přes více datových center
 - laditelná konzistence
 - odolnost vůči chybám
 - každý uzel má stejnou funkci (vs. master – slave)



<https://www.guru99.com/cassandra-tutorial.html>

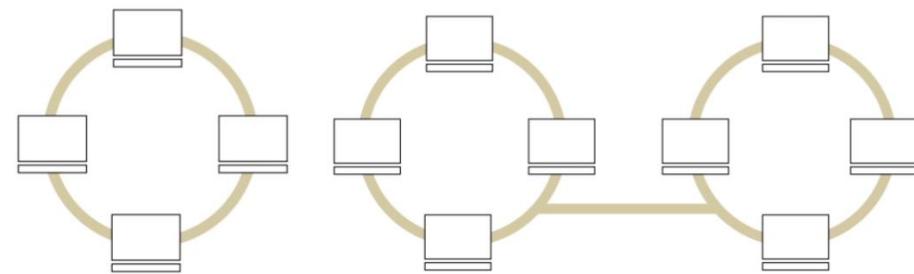
CASSANDRA

- využívána ve více jak 1500 organizacích pro různé účely
- možnosti aplikace
 - katalogy produktů / playlisty
 - Netflix, Comcast, Hulu
 - doporučovací systémy
 - Outbrain, eBay
 - detekce podvodů
 - Barracuda Networks, Instagram
 - platformy pro posílání zpráv
 - senzorová data / IoT
 - herní průmysl
 - služby založené na poloze

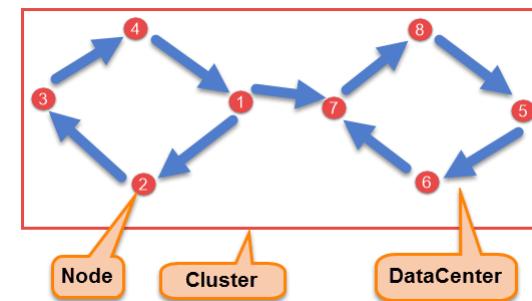


ARCHITEKTURA CASSANDRY

- základem je **uzel** (node)
 - ukládá data
 - každý uzel plní stejnou roli
 - neexistují slaves ani master
 - uzly jsou replikované
 - neexistuje jeden bod selhání
- příbuzné uzly tvoří **datová centra**
 - fyzické i virtuální
 - replikace na úrovni datových center
 - prstencová struktura
- datová centra tvoří **cluster**
 - dostupnost dat i při výpadku centra



<https://www.udemy.com/course/apache-cassandra>



<https://www.guru99.com/cassandra-architecture.html>

ARCHITEKTURA CASSANDRY

- jak o sobě ale uzly vědí?

- neexistuje master a všechny uzly také zastávají stejnou roli...

➤ **snitch**

- poskytuje uzlům informaci o topologii sítě a směruje requesty
- definuje skupiny uzelů do datových center a racků
 - replikační strategie využívá skupiny pro umístění replik
- konfigurována při vytvoření clusteru
- konfigurační soubor uložený na všech uzlech
- SimpleSnitch (základní)
 - použití jen pro clustery s jedním datovým centrem
- PropertyFileSnitch
- GossipingPropertyFileSnitch
 - doporučená pro produkční nasazení
 - používá soubor přiřazený ke každému uzlu popisující jeho lokaci, uzly si tuto informaci vyměňují
- všechny typy v základu používají dynamickou snitch vrstvu (DynamicEndpointSnitch)
 - monitoruje výkon a volí nejlepší repliku pro čtení

130.77.100.147 =DC1:RAC1
130.77.100.148 =DC1:RAC1
130.77.100.165 =DC1:RAC1
130.77.200.109 =DC1:RAC2
130.77.200.110 =DC1:RAC2
130.77.200.111 =DC1:RAC2

155.23.100.128 =DC2:RAC1
155.23.100.129 =DC2:RAC1
155.23.200.107 =DC2:RAC2
155.23.200.108 =DC2:RAC2

<https://www.udemy.com/course/apache-cassandra>



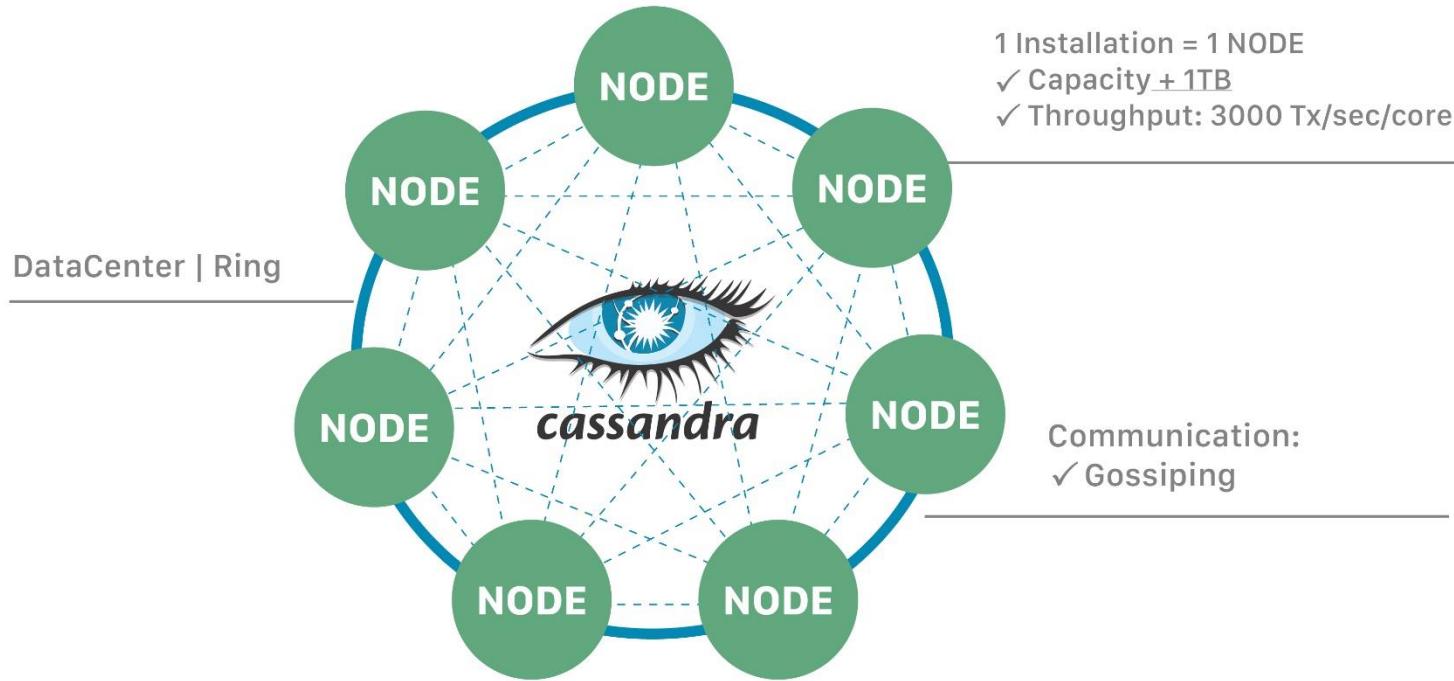
ARCHITEKTURA CASSANDRY

- jak spolu uzly komunikují?
 - gossip
 - peer-to-peer komunikační protokol pro zjišťování a sdílení informací o ostatních uzlech
 - umístění a stav
 - každou vteřinu každý uzel komunikuje až se třemi dalšími uzly
 - uzly si vyměňují informace o sobě a o všech ostatních uzlech, o kterých mají informace
 - každá gossip zpráva také obsahuje verzi pro snadnou aktualizaci
 - starší zprávy jsou přepsány novějšími
 - jedná se o interní komunikační metodu
 - pro externí komunikaci je používáno CQL
 - aplikace – Cassandra



ARCHITEKTURA CASSANDRY

ApacheCassandra™ = NoSQL Distributed Database



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



ARCHITEKTURA CASSANDRY

- distribuce dat
 - prováděna pomocí konzistentního hashování
 - cílem je rovnoměrné rozložení dat na uzlech v clusteru
 - uzly jsou členěny na menší jednotky - tokeny
 - záznamy (řádky) tabulky jsou distribuovány mezi uzly clusteru
 - snaha o rovnoměrné rozložení zátěže při načítání dat tabulky
 - pro distribuci řádků je použit **partitioner**
 - každý řádek je definovaný primárním klíčem, ze kterého je odvozen partition klíč
 - využívá algoritmus přidělující každému řádku odpovídající token (a uzel)
 - v základu Murmur3 partitioner
 - Murmur3 vezme partition klíč (celý primární klíč, případně jeho první část) a vygeneruje unikátní číslo v rozmezí -2^{63} a 2^{63}



ARCHITEKTURA CASSANDRY

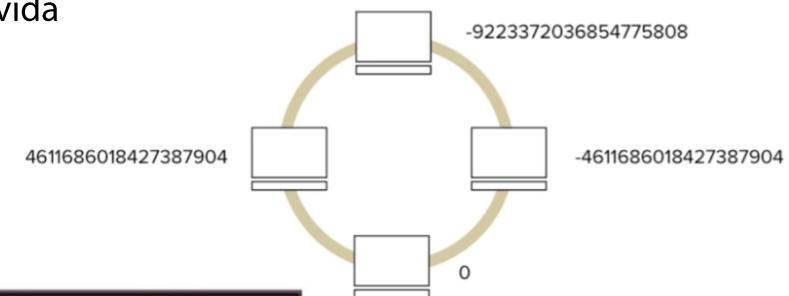
- distribuce dat

home_id	datetime	event	code_used
H01033638	2014-05-21 07:55:58	alarm set	2121
H01545551	2014-05-21 08:30:14	alarm set	8889
H00999943	2014-05-21 09:05:54	alarm set	1245

H01033638 -> **-7456322128261119046**
H01545551 -> **-2487391024765843411**
H00999943 -> **6394005945182357732**

<https://www.udemy.com/course/apache-cassandra>

- každý uzel má přiřazenou koncovou hodnotu
 - definuje rozsah tokenů, za které uzel zodpovídá
 - token range
 - nastavena manuálně
 - pro vygenerování tokenů rozsahů možnost použít libovolnou Murmur3 kalkulačku



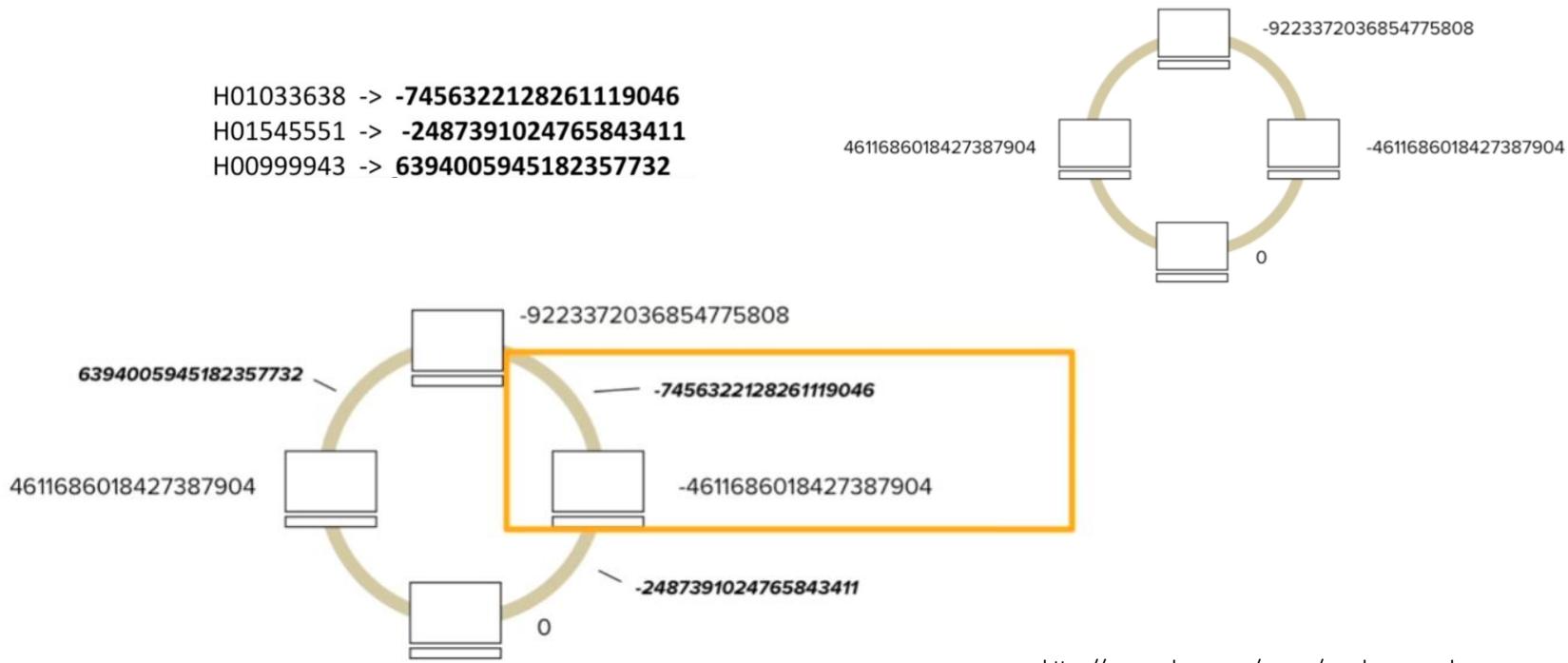
```
vm1@ubuntu:~$ python -c 'print [str(((2**64 / 4) * i) - 2**63) for i in range(4)]'  
[-9223372036854775808, '-4611686018427387904', '0', '4611686018427387904']
```



ARCHITEKTURA CASSANDRY

- distribuce dat
 - data se stejným partition klíčem jsou uložena ve stejné partition
 - na stejném tokenu

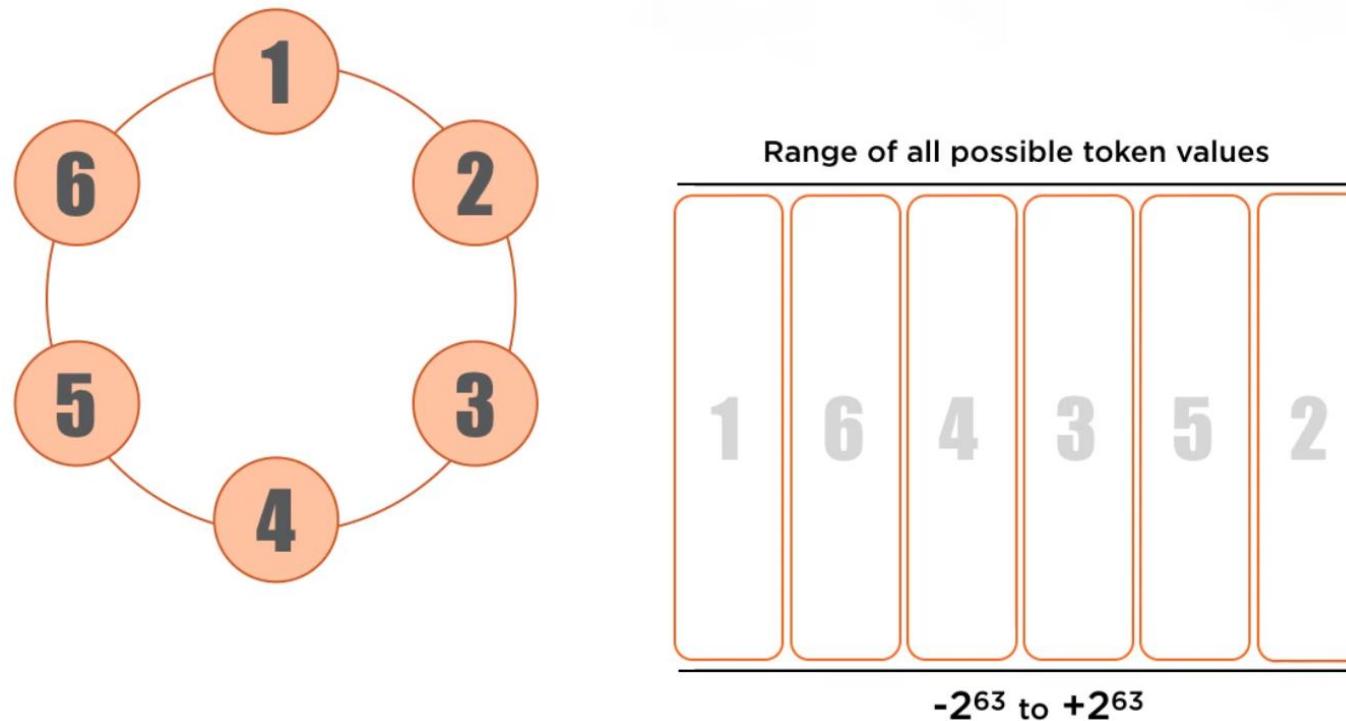
H01033638 -> **-7456322128261119046**
H01545551 -> **-2487391024765843411**
H00999943 -> **6394005945182357732**



<https://www.udemy.com/course/apache-cassandra>

ARCHITEKTURA CASSANDRY

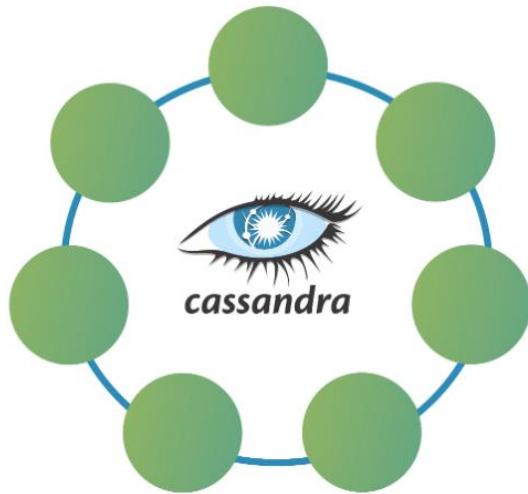
- distribuce dat



<https://app.pluralsight.com/library/courses/cassandra-developers>

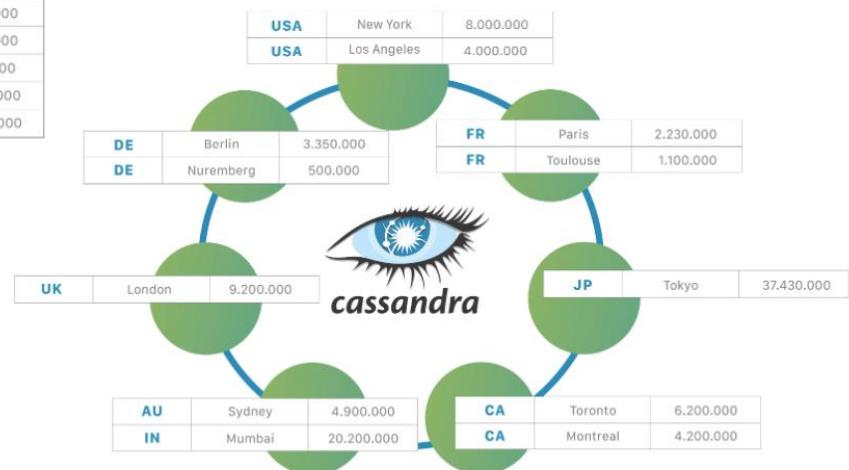
ARCHITEKTURA CASSANDRY

- distribuce dat



COUNTRY	CITY	POPULATION
USA	New York	8.000.000
USA	Los Angeles	4.000.000
FR	Paris	2.230.000
DE	Berlin	3.350.000
UK	London	9.200.000
AU	Sydney	4.900.000
DE	Nuremberg	500.000
CA	Toronto	6.200.000
CA	Montreal	4.200.000
FR	Toulouse	1.100.000
JP	Tokyo	37.430.000
IN	Mumbai	20.200.000

Partition Key

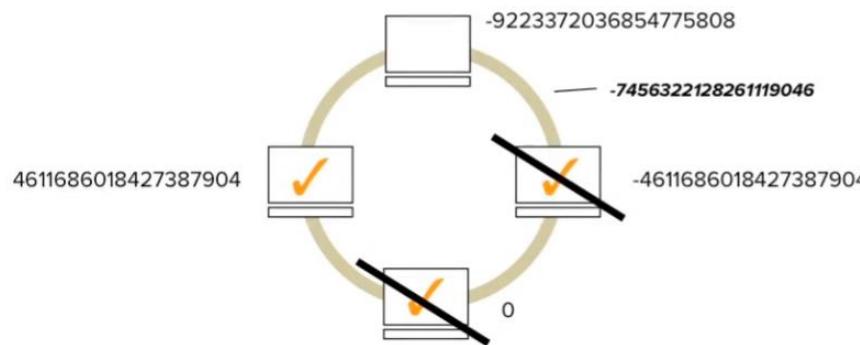


https://cassandra.apache.org/_/cassandra-basics.html



ARCHITEKTURA CASSANDRY

- replikace
 - při definování databáze je potřeba specifikovat **replikační faktor**
 - udává počet instancí dat v dané databázi
 - 1 bez replikace
 - 3 pro zajištění neexistence jednoho bodu selhání
 - výpadek uzlu neznamená ztrátu dat
 - o umístění další repliky rozhoduje **replikační strategie**



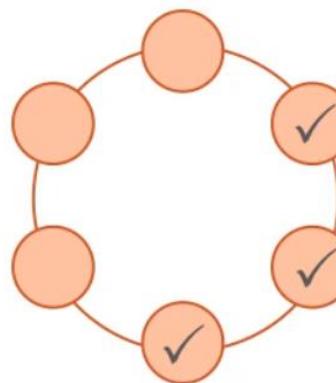
<https://www.udemy.com/course/apache-cassandra>



ARCHITEKTURA CASSANDRY

- replikace
 - o umístění další repliky rozhoduje **replikační strategie**
 - SimpleStrategy
 - většinou pro vývoj, případně pro cluster v jediném datacentru
 - synergie se SimpleSnitch

```
create keyspace pluralsight with replication =  
    {'class': 'SimpleStrategy', 'replication_factor': 3};
```



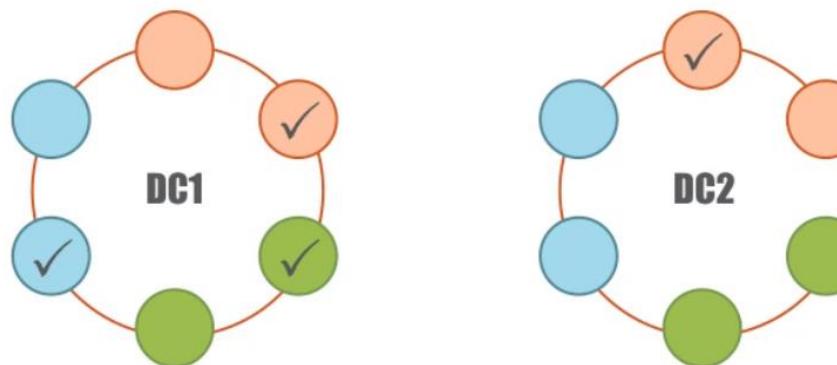
<https://app.pluralsight.com/library/courses/cassandra-developers>



ARCHITEKTURA CASSANDRY

- replikace
 - o umístění další repliky rozhoduje **replikační strategie**
 - NetworkTopologyStrategy
 - i pro clustery ve více datových centrech
 - synergie s GossipingPropertyFileSnitch

```
create keyspace pluralsight with replication =  
    {'class': 'NetworkTopologyStrategy', 'DC1': 3, 'DC2': 1};
```



<https://app.pluralsight.com/library/courses/cassandra-developers>



ARCHITEKTURA CASSANDRY

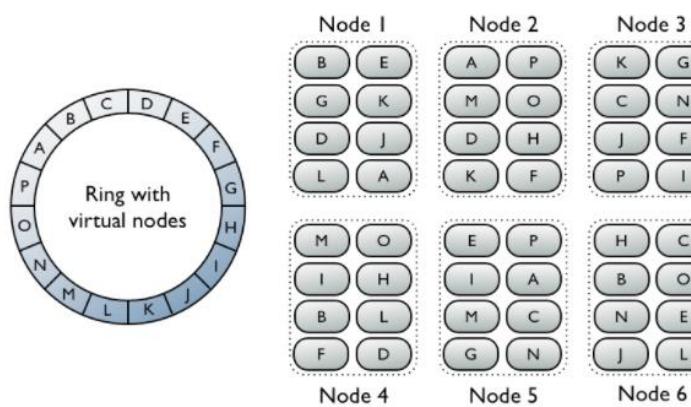
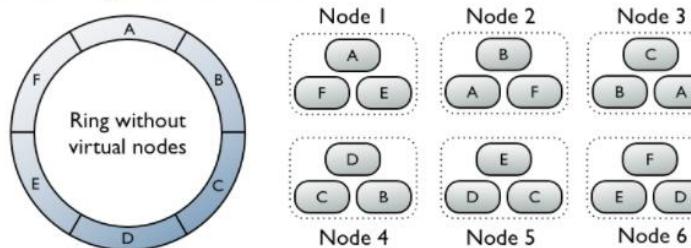
- virtuální uzly
 - již v základním nastavení
 - alternativní způsob přiřazení rozsahů uzelům
 - uzel je místo jednoho rozsahu zodpovědný za mnoho menších rozsahů
 - v základu 256
 - umožňují nastavit více rozsahů lepším strojům a méně horším
 - vytvořeny pro usnadnění přidávání nových uzelů a zároveň udržení rovnovážného rozložení dat v clusteru
 - nový uzel po přidání obdrží mnoho menších rozsahů z ostatních uzelů



ARCHITEKTURA CASSANDRY

- virtuální uzly

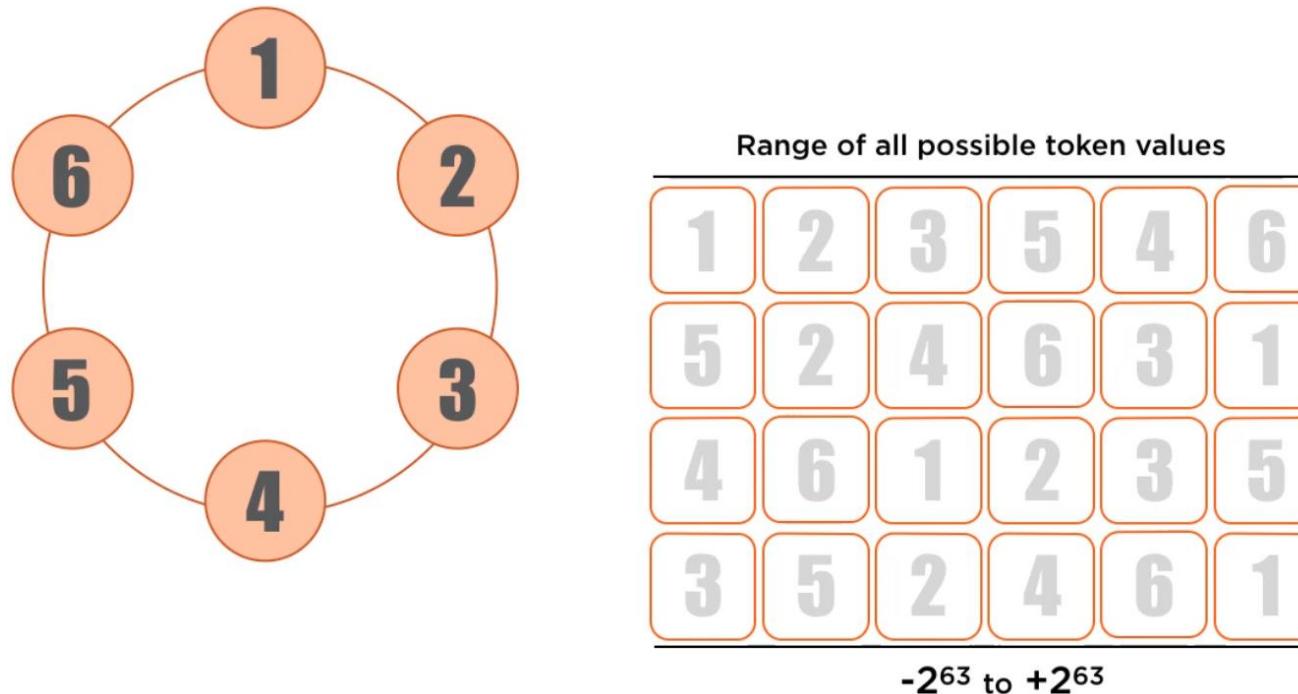
Virtual vs single-token architecture



<https://docs.datastax.com/en/cassandra-oss/3.x/cassandra/architecture/archDataDistributeDistribute.html>

ARCHITEKTURA CASSANDRY

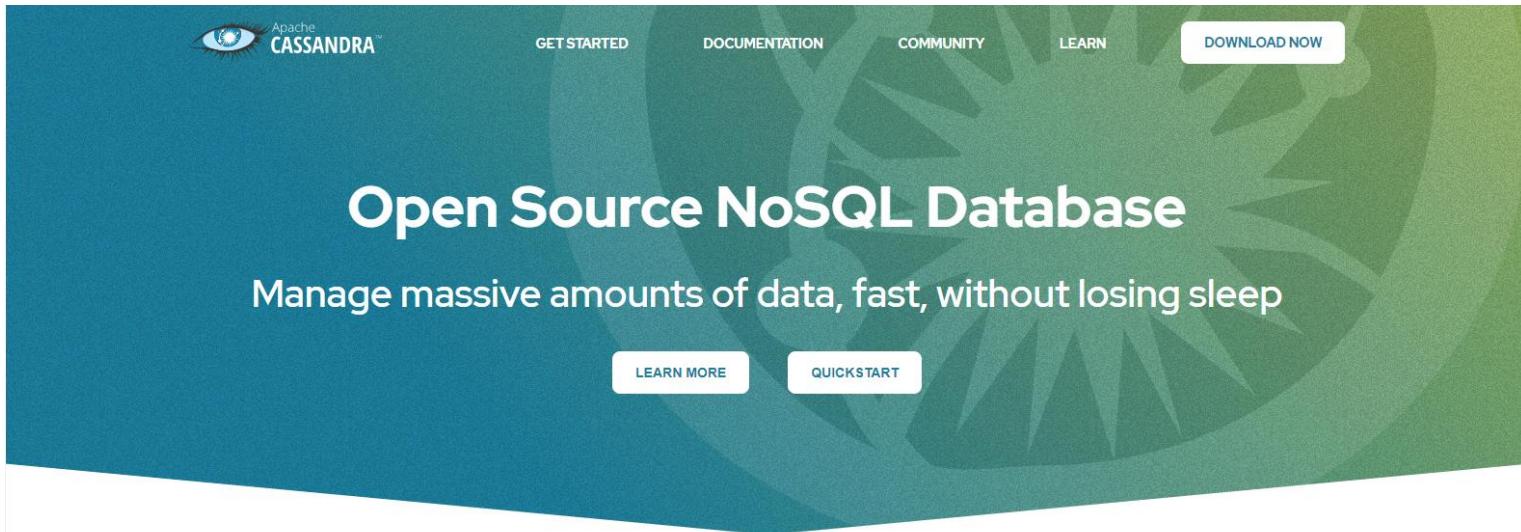
- virtuální uzly



<https://app.pluralsight.com/library/courses/cassandra-developers>

ČÁST IV.: CASSANDRA PRAKTICKY

CASSANDRA



The banner features a green gradient background with a stylized sunburst or leaf pattern. At the top, there's a navigation bar with links: GET STARTED, DOCUMENTATION, COMMUNITY, LEARN, and DOWNLOAD NOW. The Apache Cassandra logo (an eye icon) is on the left. The main text area contains the heading "Open Source NoSQL Database" and the subtext "Manage massive amounts of data, fast, without losing sleep". Below this are two buttons: "LEARN MORE" and "QUICKSTART".

Join the global Cassandra community on April 28 as we celebrate the upcoming launch of the 4.0 release!

[LEARN MORE](#)

What is Apache Cassandra?

<https://cassandra.apache.org/>



INSTALACE

- aktuální verze
 - 5.0 z 12/2023
- instalace ([Linux](#), Windows)
 - Linux preferovaný pro produkci
 - předpoklady
 - Java 8 / 11
 - nutno nastavit systémovou proměnou JAVA_HOME
JAVA_HOME C:\Progra~1\Java\jdk1.8.0_271
 - Python 3.6+
 - rozbalení staženého tar.gz archivu
 - případně instalace z repozitáře
 - jen Linux
 - hlavní adresář bin
 - alternativně [Docker](#)

Releases

Latest Version

Apache Cassandra 5.0
Latest release on 2023-12-05
Maintained until 5.3.0 release (~July 2026)

5.0-BETA1

(pgp, sha256, sha512)
(source: pgp, sha256, sha512)

```
C:\Users\Rimmer>java -version
java version "1.8.0_271"
Java(TM) SE Runtime Environment (build 1.8.0_271-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.271-b09, mixed mode)
```

bin	Složka souborů	11.1.2021 11:34
conf	Složka souborů	7.1.2021 12:44
data	Složka souborů	11.1.2021 11:37
doc	Složka souborů	7.1.2021 12:44
interface	Složka souborů	7.1.2021 12:44
javadoc	Složka souborů	7.1.2021 12:44
lib	Složka souborů	11.1.2021 11:37
logs	Složka souborů	11.1.2021 11:37
pylib	Složka souborů	7.1.2021 12:44
tools	Složka souborů	7.1.2021 12:44
var	Složka souborů	11.1.2021 13:17
ToDelete	Soubor TODELETE	11.1.2021 13:22 0 kB
CASSANDRA-14092.txt	Soubor TXT	29.10.2020 12:36 5 kB
CHANGES.txt	Soubor TXT	29.10.2020 12:36 370 kB
LICENSE.txt	Soubor TXT	29.10.2020 12:36 12 kB
NEWS.txt	Soubor TXT	29.10.2020 12:36 114 kB
NOTICE.txt	Soubor TXT	29.10.2020 12:36 3 kB
pid.txt	Soubor TXT	11.1.2021 13:22 1 kB



INSTALACE

- instalace ([Linux](#), Windows)
 - konfigurační soubory
 - adresář conf
 - hlavní konfigurační soubor `cassandra.yaml`
 - jméno clusteru
 - počet virtuálních uzlů
 - partitioner
 - umístění dat
 - snitch
 - a spousta dalšího

```
# Cassandra storage config YAML

# NOTE:
#   See http://wiki.apache.org/cassandra/StorageConfiguration for
#   full explanations of configuration directives
# /NOTE

# The name of the cluster. This is mainly used to prevent machines in
# one logical cluster from joining another.
cluster_name: 'MůjCluster'

# This defines the number of tokens randomly assigned to this node on the ring
# The more tokens, relative to other nodes, the larger the proportion of data
# that this node will store. You probably want all nodes to have the same number
# of tokens assuming they have equal hardware capability.
#
# If you leave this unspecified, Cassandra will use the default of 1 token for legacy compatibility,
# and will use the initial_token as described below.
#
# Specifying initial_token will override this setting on the node's initial start,
# on subsequent starts, this setting will apply even if initial token is set.
#
# If you already have a cluster with 1 token per node, and wish to migrate to
# multiple tokens per node, see http://wiki.apache.org/cassandra/Operations
num_tokens: 256
```



INSTALACE

- instalace ([Linux](#), Windows)
 - spuštění serveru přes .\bin\cassandra.bat
 - spustí v pozadí
 - přes .\bin\cassandra.bat -f
 - spustí v popředí

```
(base) PS D:\Programy\apache-cassandra-3.11.9\bin> .\cassandra.bat -f
```

```
(base) PS D:\Programy\apache-cassandra-3.11.9\bin> .\cassandra.bat --f
Detected powershell execution permissions.  Running with enhanced startup scripts.

Security warning
Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your
computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning
message. Do you want to run D:\Programy\apache-cassandra-3.11.9\bin\cassandra.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"): R

Security warning
Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your
computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning
message. Do you want to run D:\Programy\apache-cassandra-3.11.9\bin\source-conf.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"): R

Security warning
Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your
computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning
message. Do you want to run D:\Programy\apache-cassandra-3.11.9\conf\cassandra-env.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"): R

[INFO  [main] 2021-01-11 11:37:33,107 StorageService.java:1492 - JOINING: Finish joining ring
[INFO  [main] 2021-01-11 11:37:34,514 NativeTransportService.java:73 - Netty using Java NIO event loop
[INFO  [main] 2021-01-11 11:37:34,831 Server.java:158 - Using Netty Version: [netty-buffer=netty-buffer-4.0.44.Final.452812a, netty-codec=netty-codec-4.0.44.Final.452812a, netty-codec-haproxy=netty-codec-haproxy-4.0.44.Final.452812a, netty-common=netty-common-4.0.44.Final.452812a, netty-handler=netty-handler-4.0.44.Final.452812a, netty-tcnative=netty-tcnative-1.1.3.3.Fork26.142ecbb, netty-transport=netty-transport-4.0.44.Final.452812a, netty-transport-native-epoll=netty-transport-native-epoll-4.0.44.Final.452812a, netty-transport-epoll=netty-epoll-4.0.44.Final.452812a, netty-transport-rxtx=netty-transport-rxtx-4.0.44.Final.452812a, netty-transport-sctp=netty-transport-sctp-4.0.44.Final.452812a, netty-transport-udt=netty-transport-udt-4.0.44.Final.452812a]
[INFO  [main] 2021-01-11 11:37:34,832 Server.java:159 - Starting listening for CQL clients on localhost/127.0.0.1:9042 (unencrypted)...
[INFO  [main] 2021-01-11 11:37:35,073 CassandraDaemon.java:564 - Not starting RPC server as requested. Use JMX (StorageService->startRPCServer()) or nodetool (enablethrift) to start it
[INFO  [main] 2021-01-11 11:37:35,074 CassandraDaemon.java:650 - Startup complete
[INFO  [OptionalTasks:1] 2021-01-11 11:37:43,980 CassandraRoleManager.java:372 - Created default superuser role 'cassandra'
```



INSTALACE

- instalace ([Linux](#), Windows)

- .\bin\nodetool

- status serveru

- nodetool status
 - UN – up a normal
 - IP adresa
 - množství dat
 - počet rozsahů
 - podíl rozsahů
 - host ID
 - rack

```
(base) PS D:\Programy\apache-cassandra-3.11.9\bin> nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load      Tokens     Owns (effective)  Host ID
UN 127.0.0.1    127,71 KiB   256        100,0%          d200cb1d-5008-4f34-b5cf-f92ed4b90ad5  rack1
```

- info o uzlu

- nodetool -h 127.0.0.1 info

```
(base) PS D:\Programy\apache-cassandra-3.11.9\bin> nodetool -h 127.0.0.1 info
ID: d200cb1d-5008-4f34-b5cf-f92ed4b90ad5
Gossip active: true
Thrift active: false
Native Transport active: true
Load: 127,71 KiB
Generation No: 1610367772
Uptime (seconds): 1524
Heap Memory (MB): 547,69 / 8112,00
Off Heap Memory (MB): 0,00
Data Center: datacenter1
Rack: rack1
Exceptions: 0
Key Cache: entries 16, size 1,23 KiB, capacity 100 MiB, 70 hits, 86 requests, 0,814 recent hit rate, 14400 save period in seconds
Row Cache: entries 0, size 0 bytes, capacity 0 bytes, 0 hits, 0 requests, NaN recent hit rate, 0 save period in seconds
Counter Cache: entries 0, size 0 bytes, capacity 50 MiB, 0 hits, 0 requests, NaN recent hit rate, 7200 save period in seconds
Chunk Cache: entries 13, size 832 KiB, capacity 480 MiB, 26 misses, 121 requests, 0,785 recent hit rate, NaN microseconds miss latency
Percent Repaired: 100,0%
Token: (invoke with -T--tokens to see all 256 tokens)
```



KOMUNIKACE

- pro vývojové aktivity
 - Cassandra Query Language (cqlsh)
 - CQL (shell)
 - dotazovací jazyk podobný SQL
 - vytvořený pro usnadnění používání Cassandry
 - dnes hlavní způsob komunikace s Cassandrou
 - drivery pro různé programovací jazyky
 - Datastax Studio
- pro administrační aktivity
 - monitorování clusteru, management úlohy
 - nástroje postavené na Java Management Extensions (JMX)
 - např. nodetool, JConsole





CQL

- Cassandra Query Language
 - dotazovací jazyk podobný SQL
 - [dokumentace](#)
 - není case sensitive
 - nemá veškeré možnosti jako SQL
 - z důvodu distribuované povahy Cassandry
 - např. chybí joins
 - byly by neefektivní

```
SELECT home_id, datetime, event, code_used FROM activity;
```

vybere sloupce home_id, ..., code_used z tabulky activity



CQL

- Cassandra Query Language
 - cqlsh
 - CQL shell commands
 - .\bin\cqlsh
 - umožňuje používání CQL i CQLSH dotazů z příkazové řádky

```
(base) PS D:\Programy\apache-cassandra-3.11.9\bin> .\cqlsh
WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.

Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.9 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> _
```

```
cqlsh> HELP;
Documented shell commands:
=====
CAPTURE  CLS          COPY  DESCRIBE  EXPAND  LOGIN  SERIAL  SOURCE  UNICODE
CLEAR    CONSISTENCY  DESC   EXIT      HELP    PAGING  SHOW   TRACING

CQL help topics:
=====
AGGREGATES      CREATE_KEYSPACE     DROP_TRIGGER    TEXT
ALTER_KEYSPACE   CREATE_MATERIALIZED_VIEW  DROP_TYPE    TIME
ALTER_MATERIALIZED_VIEW  CREATE_ROLE    DROP_USER    TIMESTAMP
ALTER_TABLE      CREATE_TABLE        FUNCTIONS    TRUNCATE
ALTER_TYPE       CREATE_TRIGGER     GRANT        TYPES
ALTER_USER       CREATE_TYPE        INSERT        UPDATE
APPLY           CREATE_USER        INSERT_JSON   USE
ASCII            DATE             INT         UUID
BATCH            DELETE           JSON
BEGIN           DROP_AGGREGATE    KEYWORDS
BLOB             DROP_COLUMNFAMILY LIST_PERMISSIONS
BOOLEAN          DROP_FUNCTION    LIST_ROLES
COUNTER          DROP_INDEX       LIST_USERS
CREATE_AGGREGATE  DROP_KEYSPACE   PERMISSIONS
CREATE_COLUMNFAMILY  DROP_MATERIALIZED_VIEW REVOKE
CREATE_FUNCTION   DROP_ROLE        SELECT
CREATE_INDEX      DROP_TABLE       SELECT_JSON
```

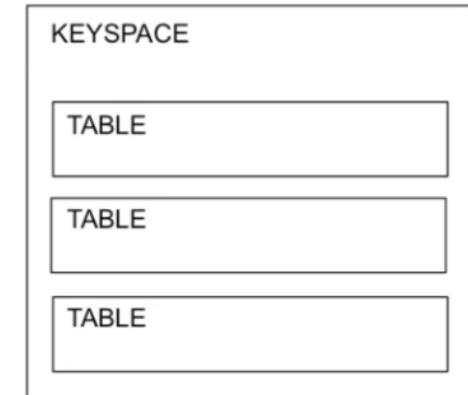


PRAKTICKÝ

- v Cassandře je databáze definována jako **keyspace**
 - volně odpovídá schématu v relačních databázích
 - dává představu o struktuře databáze
 - při instalaci vytvoří Cassandra systémové keyspaces

```
cqlsh> DESCRIBE KEYSPACES;
system_traces  system_schema  system_auth  system  system_distributed
```

```
cqlsh> DESCRIBE KEYSPACE system;
CREATE KEYSPACE system WITH replication = {'class': 'LocalStrategy'} AND durable_writes = true;
CREATE TABLE system.available_ranges (
    keyspace_name text PRIMARY KEY,
    ranges set<blob>
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = 'available keyspace/ranges during bootstrap/replace that are ready to be served'
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32',
'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.0
    AND default_time_to_live = 0
    AND gc_grace_seconds = 0
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 3600000
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
CREATE TABLE system.batches (
    id timeuuid PRIMARY KEY,
    mutations list<blob>,
```

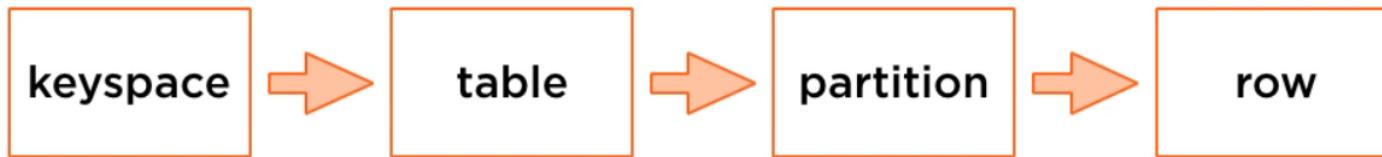


<https://www.udemy.com/course/apache-cassandra>



PRAKTICKÝ

- v Cassandře je databáze definována jako **keyspace**
- keyspace obsahuje **tabulky** (tables)
- všechna data mají přiřazený **partition** klíč
 - určuje jejich umístění v clusteru
 - všechna data v partition uložena spolu
- data v partition reprezentována jako jeden nebo více **řádků**



<https://app.pluralsight.com/library/courses/cassandra-developers>

PRAKTICKÝ

- vytvoření keyspace
 - CREATE KEYSPACE <identifikátor> WITH <vlastnosti>;

```
CREATE KEYSPACE vehicle_tracker
  WITH REPLICATION = { 'class' : 'SimpleStrategy',
    'replication_factor' : 1};
```

- vytvoří keyspace vehicle_tracker
- identifikátor je název keyspace
 - vehicle_tracker
 - case sensitive
 - bez použití uvozovek automaticky převeden na malá písmena
- vlastnosti jsou možné dvě
 - replication
 - durable_writes



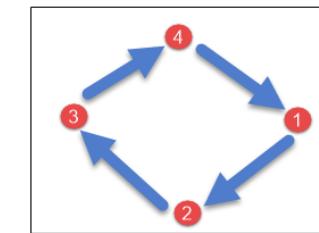
PRAKTICKÝ

- vytvoření keyspace
 - CREATE KEYSPACE <identifikátor> WITH <vlastnosti>;

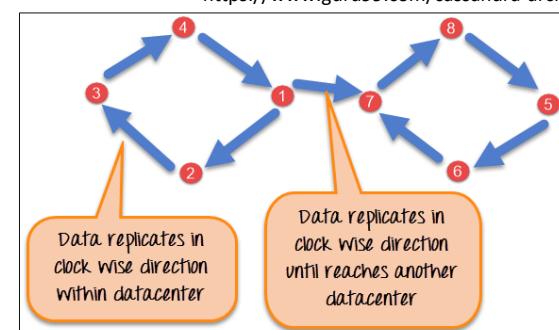
```
CREATE KEYSPACE vehicle_tracker
WITH REPLICATION = { 'class' : 'SimpleStrategy',
'replication_factor' : 1};
```

<https://www.udemy.com/course/apache-cassandra>

- vlastnost replication
 - specifikuje strategii pro umístování replik a jejich množství
 - class: SimpleStrategy nebo NetworkTopologyStrategy
 - SimpleStrategy – umístění keyspace jen v jednom datovém centru
 - NetworkTopologyStrategy – umožnuje umístění ve více centrech
 - replication_factor: počet replik
 - udává počet replik
 - u NetworkTopologyStrategy se definuje počet replik pro všechna datová centra
 - v reálné aplikaci se využívá většinou NetworkTopologyStrategy
 - replikace na úrovni keyspace



<https://www.guru99.com/cassandra-architecture.html>

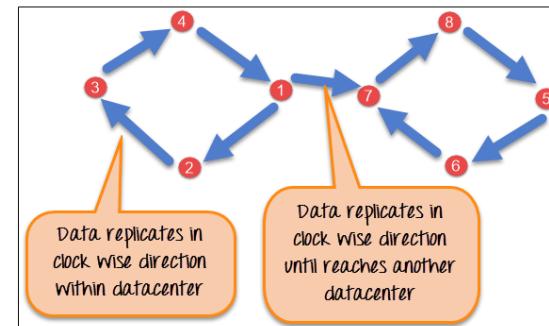


PRAKTICKÝ

- vytvoření keyspace

```
CREATE KEYSPACE vehicle_tracker
WITH REPLICATION = { 'class' : 'NetworkTopologyStrategy',
'dc1' : 3, 'dc2' : 2};
```

- datové centrum 1 – 3 repliky
- datové centrum 2 – 2 repliky



<https://www.guru99.com/cassandra-architecture.html>

```
cqlsh> CREATE KEYSPACE vehicle_tracker WITH REPLICATION = { 'class': 'SimpleStrategy', 'replication_factor': 1 };
cqlsh> DESCRIBE KEYSPACES

system_schema  system          system_distributed
system_auth    vehicle_tracker  system_traces

cqlsh> DESCRIBE KEYSPACE vehicle_tracker;

CREATE KEYSPACE vehicle_tracker WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'} AND durable_writes = true;
```

<https://www.udemy.com/course/apache-cassandra>



PRAKTICKÝ

- aktualizace vlastností keyspace
 - ALTER KEYSPACE <identifikátor> WITH <vlastnosti>;
 - vlastnosti stejné jako při vytváření

```
cqlsh> ALTER KEYSPACE vehicle_tracker WITH REPLICATION = { 'class': 'SimpleStrategy', 'replication_factor': 2 };
cqlsh> DESCRIBE KEYSPACE vehicle_tracker;

CREATE KEYSPACE vehicle_tracker WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '2'} AND durable_writes = true;
```

- smazání keyspace
 - DROP KEYSPACE <identifikátor>;

```
cqlsh> DROP KEYSPACE vehicle_tracker;
cqlsh> DESCRIBE KEYSPACES;

system_schema  system_auth  system  system_distributed  system_traces
```



PRAKTICKÝ

- přepnutí do keyspace
 - USE KEYSPACE;
 - veškeré další příkazy budou prováděny nad vybranou keyspace

```
cqlsh> USE vehicle_tracker;  
cqlsh:vehicle_tracker>
```

- ke keyspace lze také přistoupit přes tečkovou notaci
 - ALTER TABLE, CREATE TABLE, DELETE, INSERT, SELECT, TRUNCATE, UPDATE

```
cqlsh> INSERT INTO cycling.race_winners ( race_name, race_position, cyclist_name ) VALUES (  
    'National Championships South Africa WJ-ITT (CN)',  
    1,  
    {firstname:'Frances',lastname:'DU TOUT'}  
);
```

https://docs.datastax.com/en/cql-oss/3.3/cql/cql_using/useKSQualifier.html



PRAKTICKÝ

- vytvoření tabulky
 - CREATE TABLE <jméno tabulky> (<definice sloupců>) (WITH <možnosti> AND <možnosti>);

```
CREATE TABLE activity(
    home_id text,
    datetime timestamp,
    event text,
    code_used text,
    PRIMARY KEY (home_id, datetime)
) WITH CLUSTERING ORDER BY (datetime DESC);
```

<https://www.udemy.com/course/apache-cassandra>

- odstranění tabulky
 - DROP TABLE <jméno tabulky>;

```
DROP TABLE activity;
```





PRAKTICKÝ

- vytvoření tabulky
 - definice sloupců
 - jméno a datový typ
 - podpora různých datových typů

Numeric

`bigint, decimal, double, float, int, varint`

String

`ascii, text, varchar`

Date

`timestamp, timeuuid`

Other

`boolean, uuid, inet, blob`

home_id	datetime	event	code_used
H01474777	2014-05-21 07:32:16	alarm set	5599
H01474777	2014-05-21 18:30:33	alarm turned off	5599
H01474777	2014-05-22 07:44:13	alarm set	5599
H01474777	2014-05-22 11:23:59	alarm breached	
H01474777	2014-05-22 11:25:00	police called	
H01474777	2014-05-22 11:44:07	alarm reset by office	

```
CREATE TABLE activity (
    home_id text,
    datetime timestamp,
    event text,
    code_used text,
    ...
```

<https://app.pluralsight.com/library/courses/cassandra-developers>

<https://www.udemy.com/course/apache-cassandra>



PRAKTICKÝ

- vytvoření tabulky
 - primární klíč
 - PRIMARY KEY
 - jednoznačný identifikátor záznamu v CQL tabulce
 - definovaný nad jedním (jednoduchý, simple) nebo více sloupců (složený, compound)

home_id	datetime	event	code_used
H01474777	2014-05-21 07:32:16	alarm set	5599
H01474777	2014-05-21 18:30:33	alarm turned off	5599
H01474777	2014-05-22 07:44:13	alarm set	5599
H01474777	2014-05-22 11:23:59	alarm breached	
H01474777	2014-05-22 11:25:00	police called	
H01474777	2014-05-22 11:44:07	alarm reset by office	

- složený klíč nad home_id a datetime

PRIMARY KEY (home_id, datetime)

```
CREATE TABLE activity (
    home_id text,
    datetime timestamp,
    event text,
    code_used text,
    PRIMARY KEY (home_id, datetime)
    ...
```

<https://www.udemy.com/course/apache-cassandra>



PRAKTICKÝ

- vytvoření tabulky
 - primární klíč
 - v případě definice jen nad jedním sloupcem, dva způsoby zápisu

```
CREATE TABLE home (
    home_id text,
    address text,
    city text,
    state text,
    zip text,
    owner text,
    phone text,
    alt_phone text,
    email text,
    phone_password text,
    main_code text,
    guest_code text,
    PRIMARY KEY (home_id)
);
```

```
CREATE TABLE home (
    home_id text PRIMARY KEY,
    address text,
    city text,
    state text,
    zip text,
    owner text,
    phone text,
    alt_phone text,
    email text,
    phone_password text,
    main_code text,
    guest_code text
);
```

<https://www.udemy.com/course/apache-cassandra>

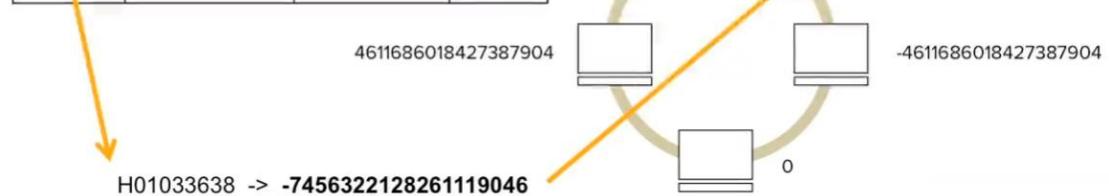


PRAKTICKÝ

- vytvoření tabulky
 - partition klíč
 - zodpovídá za distribuci dat na uzly
 - a přístup k nim
 - partitioner hodnotu klíče hashuje a na základě výsledku přiřadí partition (a rozsah a uzel)
 - odvozený z primárního klíče
 - z jednoduchého primárního klíče je partition klíč celý primární klíč
 - ze složeného primárního klíče je partition klíč první část primárního klíče
 - zbylé sloupce se nazývají clustering columns a slouží k setřídění partition
 - v praxi všechny CQL řádky se stejným partition klíčem jsou uloženy ve stejné partition

```
CREATE TABLE activity (
    home_id text,
    datetime timestamp,
    event text,
    code_used text,
    PRIMARY KEY (home_id, datetime)
    ...
```

home_id	datetime	event	code_used
H01033638	2014-05-21 07:55:58	alarm set	2121
H0154551	2014-05-21 08:30:14	alarm set	8889
H0099943	2014-05-21 09:05:54	alarm set	1245



<https://www.udemy.com/course/apache-cassandra>

PRAKTICKÝ

- vytvoření tabulky
 - partition klíč

home_id	datetime	event	code_used
H01474777	2014-05-21 07:32:16	alarm set	5599
H01474777	2014-05-21 18:30:33	alarm tur	RowKey: H01474777 => (name=2014-05-21 07\:32\:16-0700:, value=, timestamp=1398811533) => (name=2014-05-21 07\:32\:16-0700:code_used, value=35353939) => (name=2014-05-21 07\:32\:16-0700:event, value=616c61726d20)
H01474777	2014-05-22 07:44:13	alarm se	=> (name=2014-05-21 18\:30\:33-0700:, value=, timestamp=1398811533) => (name=2014-05-21 18\:30\:33-0700:code_used, value=35353939) => (name=2014-05-21 18\:30\:33-0700:event, value=616c61726d20)
H01474777	2014-05-22 11:23:59	alarm br	=> (name=2014-05-21 18\:30\:33-0700:, value=, timestamp=1398811533) => (name=2014-05-21 18\:30\:33-0700:code_used, value=35353939) => (name=2014-05-21 18\:30\:33-0700:event, value=616c61726d20)
H01474777	2014-05-22 11:25:00	police ca	=> (name=2014-05-21 18\:30\:33-0700:, value=, timestamp=1398811533) => (name=2014-05-21 18\:30\:33-0700:code_used, value=35353939) => (name=2014-05-21 18\:30\:33-0700:event, value=616c61726d20)
H01474777	2014-05-22 11:44:07	alarm re	=> (name=2014-05-22 07\:44\:13-0700:, value=, timestamp=1398811533) => (name=2014-05-22 07\:44\:13-0700:code_used, value=35353939) => (name=2014-05-22 07\:44\:13-0700:event, value=616c61726d20)

<https://www.udemy.com/course/apache-cassandra>



PRAKTICKÝ

- vytvoření tabulky
 - composite partition klíč
 - využití více sloupců jako partition klíč
 - více záznamů v jedné partition
 - zpravidla se používá v případech, kdy data uložená v partitions jsou příliš velká
 - rozdělení dat na části

```
cqlsh> CREATE TABLE cycling.rank_by_year_and_name (
    race_year int,
    race_name text,
    cyclist_name text,
    rank int,
    PRIMARY KEY ((race_year, race_name), rank)
);
```

https://docs.datastax.com/en/cql-oss/3.3/cql/cql_using/useKSQualifier.html



PRAKTICKÝ

- vytvoření tabulky
 - tabulka může být definovaná, aby data ukládala vzestupně (v základu) nebo sestupně
 -) WITH CLUSTERING ORDER BY (datetime DESC);
 - datetime se v tomto případě nazývá clustering column a tabulka je podle něj řazena
 - nastavení sestupného řazení zpomaluje zápis (záznamy jsou ukládány na začátek místo na konec partition), ale zrychluje čtení, pokud aplikace vyžaduje sestupné řazení
 - řazení nelze později změnit
 - tabulka by musela být smazána a znova vytvořena s požadovaným řazením

```
cqlsh:vehicle_tracker> CREATE TABLE activity (
    ... home_id text,
    ... datetime timestamp,
    ... event text,
    ... code_udes text,
    ... PRIMARY KEY (home_id, datetime)
    ... ) WITH CLUSTERING ORDER BY (datetime DESC);
```



PRAKTICKÝ

- vytvoření tabulky

```
cqlsh:vehicle_tracker> CREATE TABLE activity (
    ... home_id text,
    ... datetime timestamp,
    ... event text,
    ... code_udes text,
    ... PRIMARY KEY (home_id, datetime)
    ... ) WITH CLUSTERING ORDER BY (datetime DESC);
```

```
cqlsh:vehicle_tracker> DESCRIBE TABLE activity
CREATE TABLE vehicle_tracker.activity (
    home_id text,
    datetime timestamp,
    code_udes text,
    event text,
    PRIMARY KEY (home_id, datetime)
) WITH CLUSTERING ORDER BY (datetime DESC)
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy',
        'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

```
cqlsh:vehicle_tracker> CREATE TABLE home (
    ... home_id text,
    ... address text,
    ... city text,
    ... state text,
    ... zip text,
    ... contact_name text,
    ... phone text,
    ... alt_phone text,
    ... phone_password text,
    ... email text,
    ... main_code text,
    ... guest_code text,
    ... PRIMARY KEY(home_id)
    ... );
```

```
cqlsh:vehicle_tracker> DESCRIBE TABLES
home  activity
```



PRAKTICKÝ

- vytvoření tabulky
 - pokročilé datové typy
 - kolekce
 - jednoduchý způsob, jak shlukovat a ukládat data ve sloupci
 - set
 - obsahuje skupinu elementů s unikátní hodnotou

```
cqlsh> CREATE TABLE cycling.cyclist_career_teams ( id UUID PRIMARY KEY, lastname text, teams set<text> );
```

- list
 - podporuje i duplicitní hodnoty
 - elementy ukládány v pořadí a lze k nim tak přistupovat

<https://app.pluralsight.com/library/courses/cassandra-developers>

```
cqlsh> CREATE TABLE cycling.upcoming_calendar ( year int, month int, events list<text>, PRIMARY KEY ( year, month ) );
```

- map
 - přiřazuje jednu hodnotu ke druhé ve vztahu klíč – hodnota
 - pro každý klíč jen jedna hodnota
 - duplikáty nejsou povoleny
 - klíč i hodnota mají definovaný datový typ

```
cqlsh> CREATE TABLE cycling.cyclist_teams ( id UUID PRIMARY KEY, lastname text, firstname text, teams map<int,text> );
```

https://docs.datastax.com/en/cql-oss/3.3/cql/cql_using/useAdvancedDataTypesTOC.html



PRAKTICKÝ

- vytvoření tabulky
 - pokročilé datové typy
 - kolekce

lastname	teams
ARMITSTEAD	{'AA Drink - Leontien.nl', 'Boels-Dolmans Cycling Team', 'Team Garmin - Cervelo'}
VOS	{'Nederland bloeit', 'Rabobank Women Team', 'Rabobank-Liv Giant', 'Rabobank-Liv Woman Cycling Team'}
BRAND	{'AA Drink - Leontien.nl', 'Leontien.nl', 'Rabobank-Liv Giant', 'Rabobank-Liv Woman Cycling Team'}
VAN DER BREGGEN	{'Rabobank-Liv Woman Cycling Team', 'Sengers Ladies Cycling Team', 'Team Flexpoint'}

year	month	events
2015	6	['Criterium du Dauphine', 'Tour de Suisse']
2015	7	['Tour de France']

lastname	firstname	teams
ARMITSTEAD	Elizabeth	{2011: 'Team Garmin - Cervelo', 2012: 'AA Drink - Leontien.nl', 2013: 'Boels-Dolmans Cycling Team', 2014: 'Boels-Dolmans Cycling Team', 2015: 'Boels-Dolmans Cycling Team'}
VOS	Marianne	{2011: 'Nederland bloeit', 2012: 'Rabobank Women Team', 2013: 'Rabobank-Liv Giant', 2014: 'Rabobank-Liv Woman Cycling Team', 2015: 'Rabobank-Liv Woman Cycling Team'}
VAN DER BREGGEN	Anna	{2009: 'Team Flexpoint', 2012: 'Sengers Ladies Cycling Team', 2013: 'Sengers Ladies Cycling Team', 2014: 'Rabobank-Liv Woman Cycling Team', 2015: 'Rabobank-Liv Woman Cycling Team'}

https://docs.datastax.com/en/cql-oss/3.3/cql/cql_using/useAdvancedDataTypesTOC.html



PRAKTICKÝ

- vytvoření tabulky
 - pokročilé datové typy
 - tuple
 - datový typ umožňující uložit dvě nebo více hodnot (v pořadí) ve sloupci

```
CREATE TABLE cycling.nation_rank ( nation text PRIMARY KEY, info tuple<int,text,int> );
```

```
CREATE TABLE cycling.popular (rank int PRIMARY KEY, cinfo tuple<text,text,int> );
```

https://docs.datastax.com/en/cql-oss/3.3/cql/cql_using/useCreateTableTuple.html

- možnost vnořovat pokročilé datové typy
 - klíčové slovo frozen
 - vnořené typy serializovány jako jediná (blob) hodnota
 - čteny a ukládány jako celek
- možnost definovat i vlastní datové typy

```
CREATE TABLE users (
    id varchar,
    first_name varchar,
    last_name varchar,
    password varchar,
    reset_token varchar,
    last_login map<varchar,
        frozen<tuple<timestamp,inet>>>,
    PRIMARY KEY (id)
```

`last_login` <https://app.pluralsight.com/library/courses/cassandra-developers>

```
{'3383cc0867cd2': ('2019-10-30 09:02:24.000000+0000', '98.203.153.64')}
```



PRAKTICKÝ

- aktualizace tabulky
 - ALTER TABLE <jméno tabulky> <instrukce>;
 - podpora dvou základních operací
 - přidání sloupce
 - odstranění sloupce

```
cqlsh:vehicle_tracker> ALTER TABLE activity
... ADD duration int;
cqlsh:vehicle_tracker> DESCRIBE TABLE activity

CREATE TABLE vehicle_tracker.activity (
    home_id text,
    datetime timestamp,
    code_udes text,
    duration int,
    event text,
    PRIMARY KEY (home_id, datetime)
) WITH CLUSTERING ORDER BY (datetime DESC)
```

```
cqlsh:vehicle_tracker> ALTER TABLE activity
... DROP duration;
cqlsh:vehicle_tracker> DESCRIBE TABLE activity

CREATE TABLE vehicle_tracker.activity (
    home_id text,
    datetime timestamp,
    code_udes text,
    event text,
    PRIMARY KEY (home_id, datetime)
) WITH CLUSTERING ORDER BY (datetime DESC)
```

- vyčištění tabulky
 - TRUNCATE <jméno tabulky>;
 - vymaže všechny záznamy z tabulky (nejprve vytvoří zálohu)



PRAKTICKÝ

- vkládání dat
 - několik základních způsobů vkládání dat do tabulek
 - **INSERT INTO**
 - CQL příkaz pro zápis jednoho záznamu
 - pomocí cqlsh nebo přes klientský driver podporující CQL (Datastax Java Driver, ...)
 - **COPY**
 - import dat z CSV souborů
 - také možnost pro export
 - **sstableloader**
 - stream SSTable souborů na cluster



PRAKTICKÝ

- vkládání dat
 - INSERT INTO <jméno tabulky> (<sloupec1>, <sloupec2>, ...) VALUES (<hodnota1>, <hodnota2>, ...) USING <možnosti>;
 - vhodné pro zápis jednoho záznamu z cqlsh nebo z klientské aplikace

```
INSERT INTO activity (home_id, datetime, event,
code_used) VALUES ('H01474777', '2014-05-21 07:32:16',
'alarm set', '5599');
```

<https://www.udemy.com/course/apache-cassandra>

```
cqlsh:vehicle_tracker> INSERT INTO activity (home_id, datetime, event, code_used)
VALUES ('H01474777', '2014-05-21 07:32:16', 'alarm set', '5599');
cqlsh:vehicle_tracker> SELECT * FROM activity;

 home_id | datetime           | code_used | event
-----+-----+-----+-----+
 H01474777 | 2014-05-21 05:32:16.000000+0000 |      5599 | alarm set

(1 rows)
```

- nemusí být uvedeny hodnoty pro všechny sloupce

```
cqlsh:vehicle_tracker> INSERT INTO activity (home_id, datetime, event)
VALUES ('H01474777', '2014-05-21 08:32:16', 'alarm');
cqlsh:vehicle_tracker> SELECT * FROM activity;

 home_id | datetime           | code_used | event
-----+-----+-----+-----+
 H01474777 | 2014-05-21 06:32:16.000000+0000 |      null |     alarm
 H01474777 | 2014-05-21 05:32:16.000000+0000 |      5599 | alarm set
```



PRAKTICKÝ

- odbočka ke čtení dat
 - SELECT <sloupce> FROM <jméno tabulky>;
 - pro čtení dat z tabulky
 - velmi podobné SQL
 - přečtení všech sloupu
 - přečtení vybraných sloupců

```
SELECT * FROM activity;
```

```
SELECT home_id, datetime, event FROM activity;
```

<https://www.udemy.com/course/apache-cassandra>

```
cqlsh:vehicle_tracker> SELECT * FROM activity;  
  
home_id | datetime | code_used | event  
-----+-----+-----+-----  
H01474777 | 2014-05-21 06:32:16.000000+0000 | null | alarm  
H01474777 | 2014-05-21 05:32:16.000000+0000 | 5599 | alarm set  
(2 rows)
```

```
cqlsh:vehicle_tracker> SELECT home_id, datetime, event FROM activity;  
  
home_id | datetime | event  
-----+-----+-----  
H01474777 | 2014-05-21 06:32:16.000000+0000 | alarm  
H01474777 | 2014-05-21 05:32:16.000000+0000 | alarm set  
(2 rows)
```



A PŘÍŠTĚ?

- sloupcové databáze
 - Cassandra...





Děkuji za pozornost.
Otázky?