

Martin Black

Regional Sales Director @ Hazelcast

martin@hazelcast.com



Emrah Kocaman

Software Engineer @ Hazelcast



emrahkocaman

emrah@hazelcast.com



<http://www.zenika.com/formation-hazelcast-essentials.html>

30Th November - Free Training





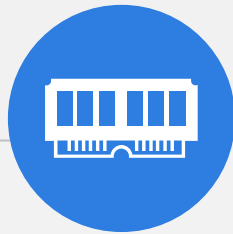
Hazelcast For Beginners

Agenda

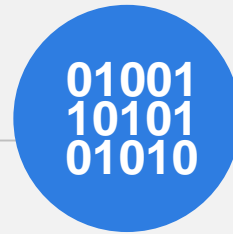
- ▶ What is Hazelcast?
- ▶ Hazelcast Demo
- ▶ Hazelcast Data Distribution
- ▶ Hazelcast Features
- ▶ Q/A Session

what is  hazelcast ?

“Hazelcast is *In-Memory Data Grid (IMDG)* Solution”



In Memory
Data **Storage**



In Memory
Data **Messaging**



In Memory
Data **Computing**

an open-source project

The screenshot shows the GitHub interface for the Hazelcast repository. At the top, there's a navigation bar with 'This repository' and a search bar. Below this, the repository name 'hazelcast / hazelcast' is displayed, along with statistics: 152 Unwatch, 879 Unstar, and 380 Fork. The description 'Open Source In-Memory Data Grid' is followed by the website URL 'http://www.hazelcast.com'. A summary bar indicates 11,565 commits, 6 branches, 66 releases, and 76 contributors. The main content area shows a commit by 'Serdaro' 4 hours ago, titled 'Workaround for wrong TOC references in Java Client'. Below the commit, a list of files and folders is shown, including 'checkstyle', 'findbugs', 'hazelcast-all', 'hazelcast-build-utils', 'hazelcast-client', 'hazelcast-cloud', 'hazelcast-documentation', 'hazelcast-hibernate', and 'hazelcast-ra'. The right sidebar contains links to 'Code', 'Issues' (246), 'Pull Requests' (20), 'Pulse', and 'Graphs'. At the bottom of the sidebar, there's a section for cloning the repository, showing the SSH clone URL 'git@github.com:haz' and buttons for 'Clone in Desktop' and 'Download ZIP'.

hazelcast / hazelcast

Open Source In-Memory Data Grid <http://www.hazelcast.com>

11,565 commits 6 branches 66 releases 76 contributors

branch: master hazelcast / +

Workaround for wrong TOC references in Java Client ...

Serdaro authored 4 hours ago latest commit 232348245d

checkstyle	CacheProxy refactored, cleaned up	4 days ago
findbugs	No findbugs issue left. Removing unnecessary suppress.	3 months ago
hazelcast-all	fixes #3537	4 days ago
hazelcast-build-utils	Revert version back to 3.4-SNAPSHOT	11 days ago
hazelcast-client	CacheProxy refactored, cleaned up, (review)	4 days ago
hazelcast-cloud	Revert version back to 3.4-SNAPSHOT	11 days ago
hazelcast-documentation	Workaround for wrong TOC references in Java Client	4 hours ago
hazelcast-hibernate	Merge pull request #3167 from mst-appear/feature/log4j2	10 days ago
hazelcast-ra	Revert version back to 3.4-SNAPSHOT	11 days ago

<> Code

Issues 246

Pull Requests 20

Pulse

Graphs

SSH clone URL


git@github.com:haz

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop

Download ZIP

an open-source project

- ▶ leading open-source in-memory data grid.
- ▶ Apache 2 License
- ▶ dead simple distributed programming
- ▶ lightweight w/o any dependency
- ▶ built with  in Istanbul

Hazelcast Inc.

- ▶ Hazelcast Enterprise
- ▶ Support Contracts
- ▶ Training / Consulting
- ▶ Offices in Istanbul & Ankara
(Engineering) , Palo Alto(HQ+Sales)
and London(Sales)

“Hazelcast is *a java library to create distributed applications*”

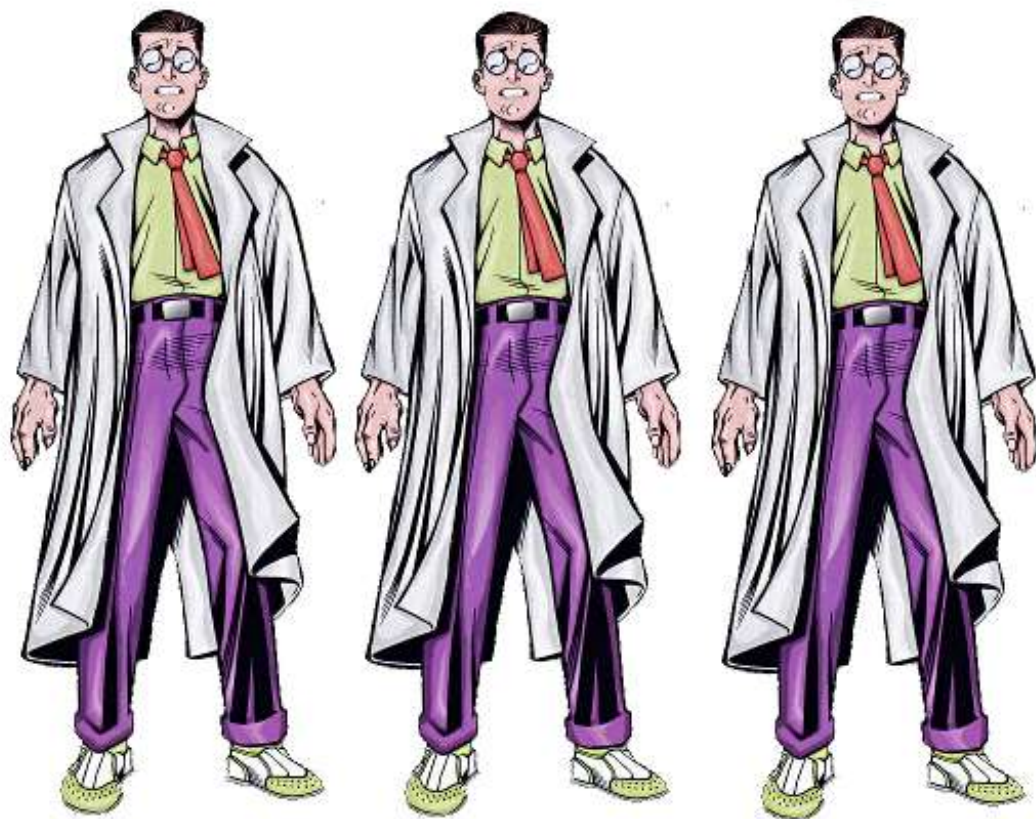
What are the main features of Distributed Applications?

What are the main features of Hazelcast?

- **Scalability**

Incredible Hulk® is a registered trademark by Marvel Characters, Inc., Cadence Industries Corporation (d.b.a. Marvel Comics Group).

Scale Out



Not Scale Up



What are the main features of Hazelcast?

- Scalability
- Speed

L1 cache reference	0.5 ns		
Branch mispredict	5 ns		
L2 cache reference	7 ns		
Mutex lock/unlock	25 ns		
Main memory reference	100 ns		
Compress 1K bytes with Zippy	3,000 ns	=	3 µs
Send 2K bytes over 1 Gbps network	20,000 ns	=	20 µs
SSD random read	150,000 ns	=	150 µs
Read 1 MB sequentially from memory	250,000 ns	=	250 µs
Round trip within same datacenter	500,000 ns	=	0.5 ms
Read 1 MB sequentially from SSD*	1,000,000 ns	=	1 ms
Disk seek	10,000,000 ns	=	10 ms
Read 1 MB sequentially from disk	20,000,000 ns	=	20 ms

Reference : <https://gist.github.com/hellerbarde/2843375>

Memory

Getting cheaper and faster



Historic RAM Prices				
Year	Manufacturer	Size (KB)	Price	Price / MB
1957	C.C.C.	0.00098	\$392	\$411,041,792
1960	E.E.Co.	0.00098	\$5	\$5,242,880
1965	IBM	0.00098	\$2.52	\$2,642,412
1970	IBM	0.00098	\$0.70	\$734,003
1975	MITS	0.25	\$103	\$421,888
1980	Interface Age	64	\$405	\$6,480
1985	Do Kay BYTE	512	\$440	\$880
1990	Unitex BYTE	8,192	\$851	\$106
1995	Pacific Coast Micro	16,384	\$494	\$30.9
2000	Crucial	65,536	\$72	\$1.12
2005	Corsair	1,048,576	\$189	\$0.185
2010	Kingston	8,388,608	\$99	\$0.0122
2013	Crucial	16,777,216	\$88	\$0.0054
2014	Patriot	32,000,000	\$294	\$0.0091

Reference : <http://www.statisticbrain.com/average-historic-price-of-ram/>

What are the main features of Distributed Applications?

- Scalability
- Speed
- Simplicity

Hazelcast is simple!



IT'S DEMO TIME

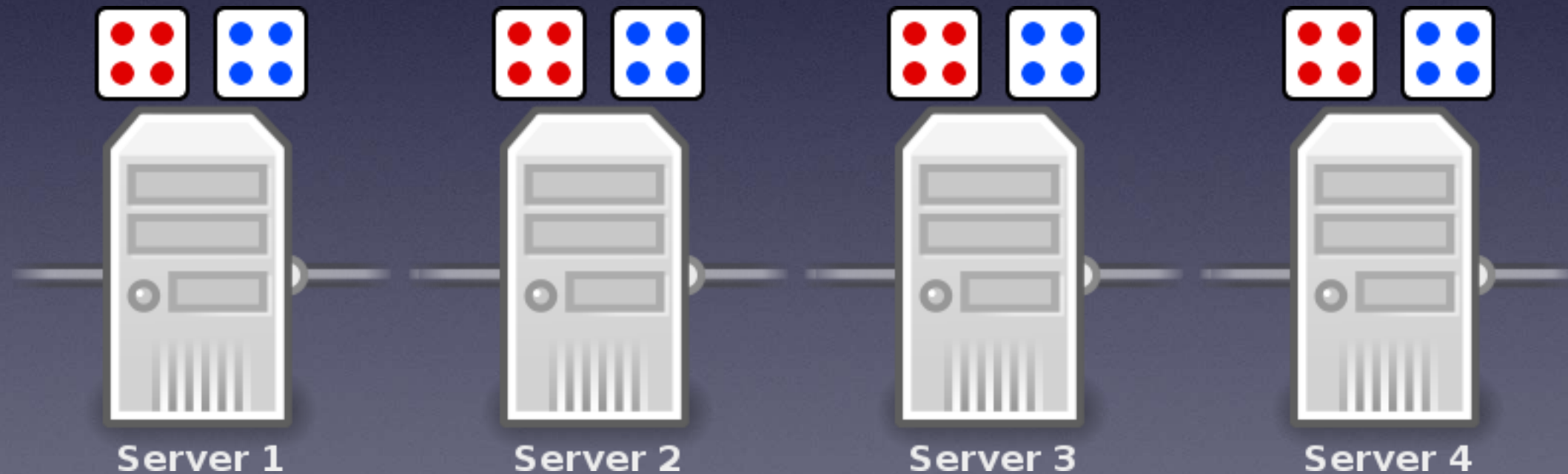


Data Backup

- ▶ `<backup-count>1</backup-count>`
- ▶ max backup count is 6
- ▶ Backups increase memory usage since they are also kept in memory.

DATA PARTITIONING

With 4 cluster nodes every server holds
 $\frac{1}{4}$ real data and $\frac{1}{4}$ of backups





Rebalance Data on New Node

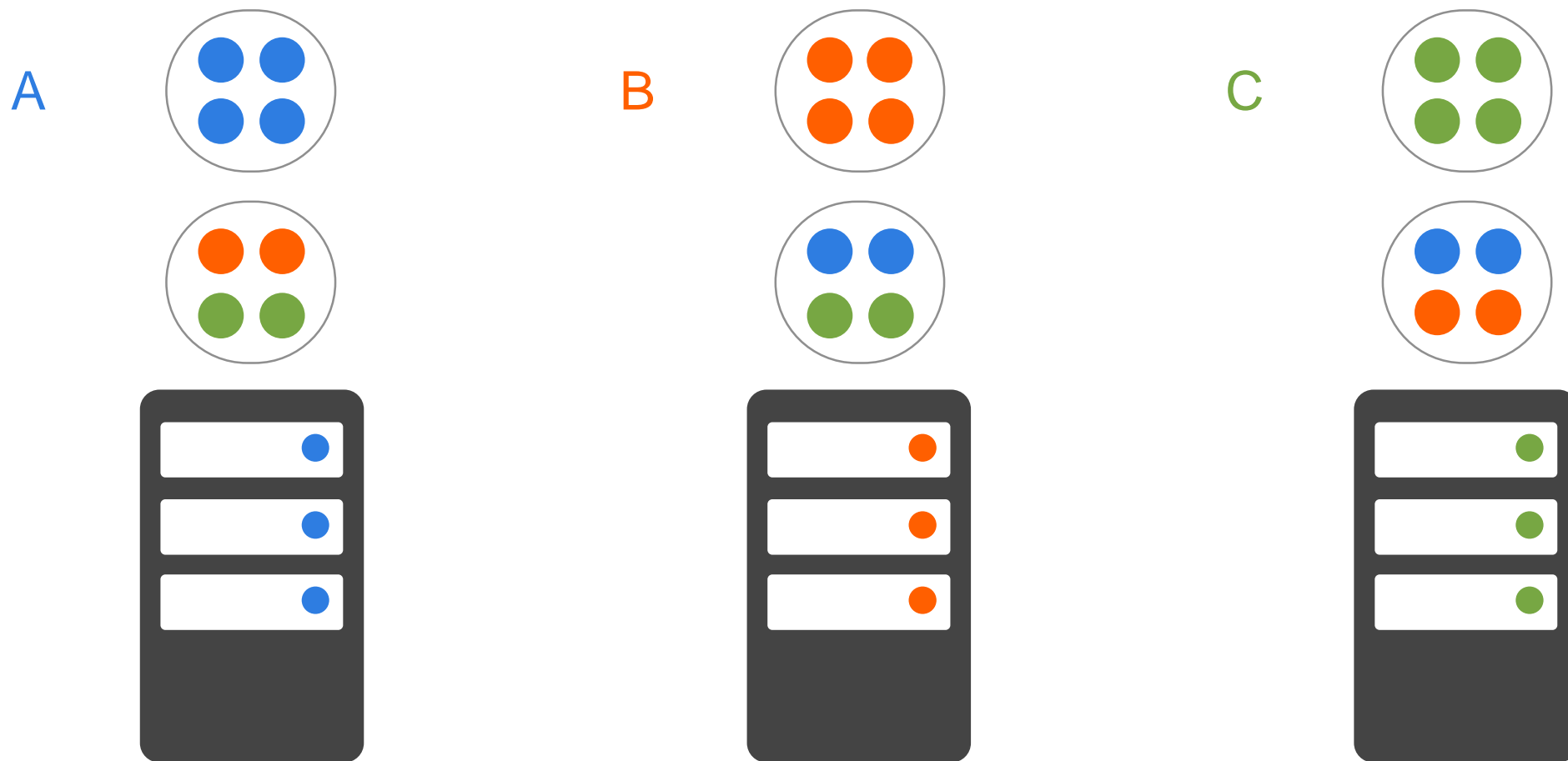
Distributed Maps

Fixed number of partitions (default 271)

Each key falls into a partition

$partitionId = hash(keyData) \% PARTITION_COUNT$

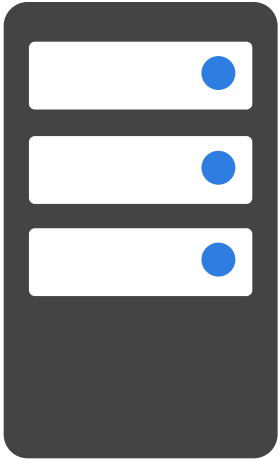
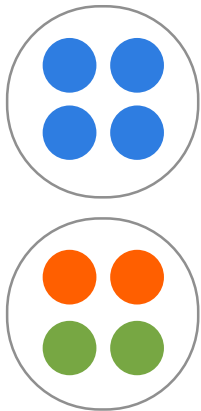
Partition ownerships are reassigned upon membership change



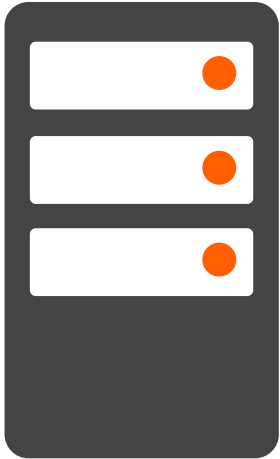
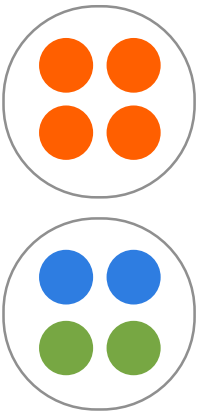


New Node Added

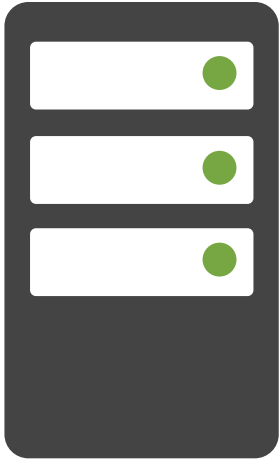
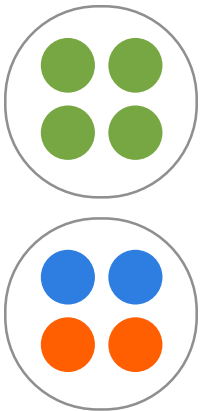
A



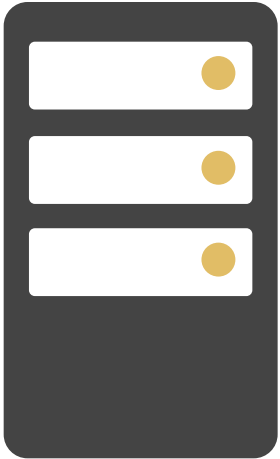
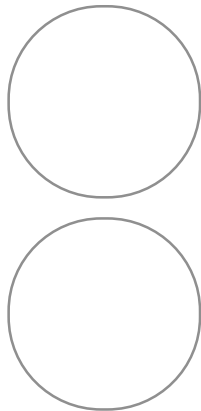
B



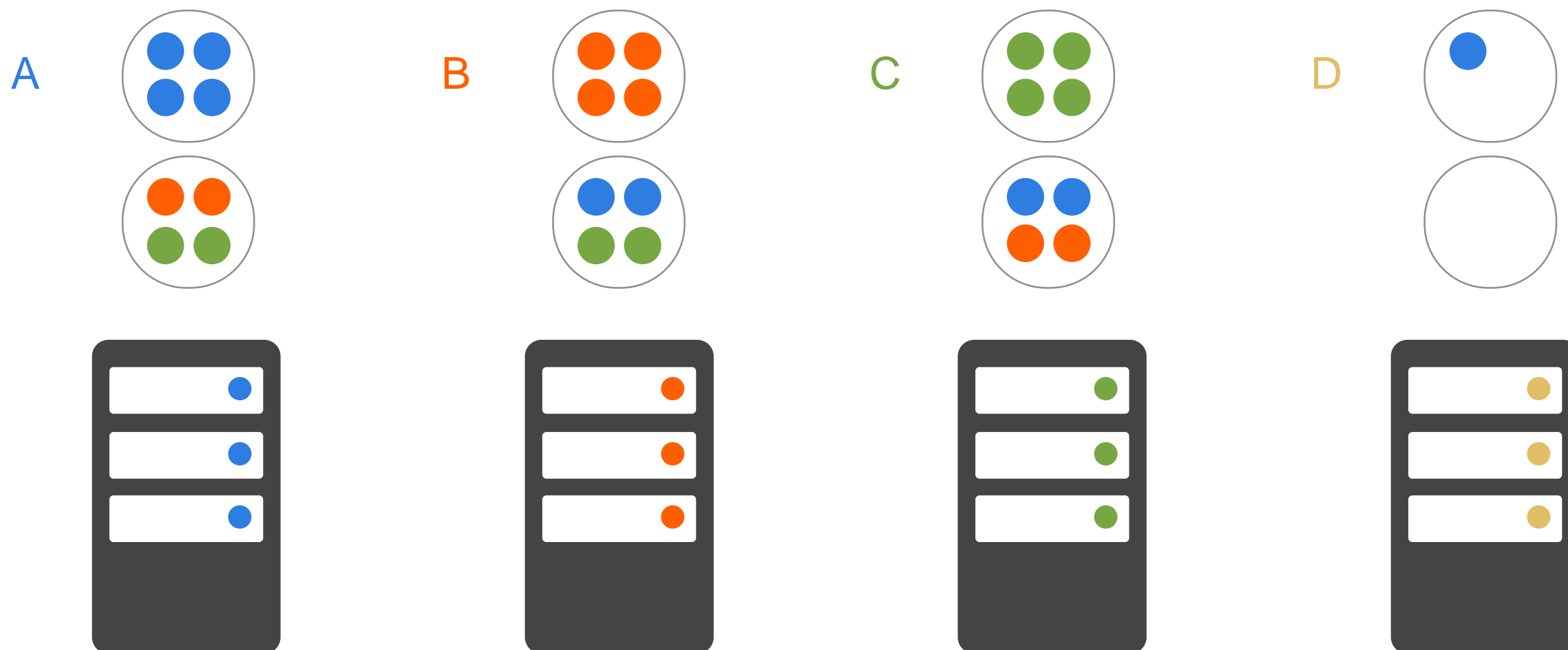
C



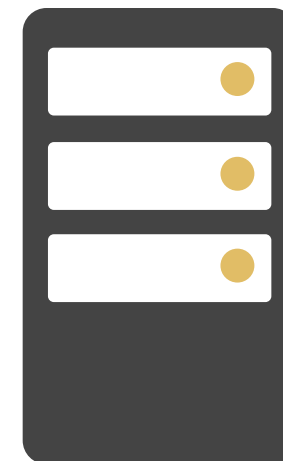
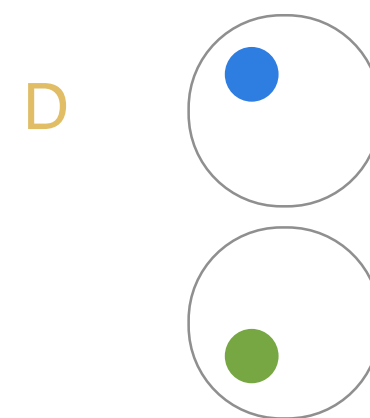
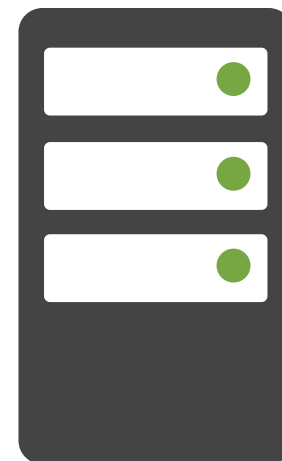
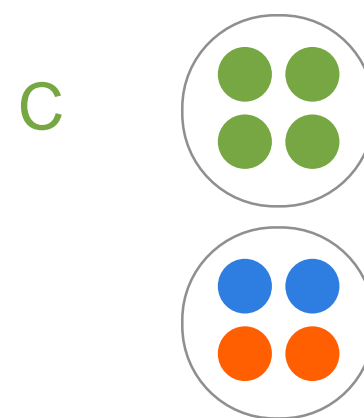
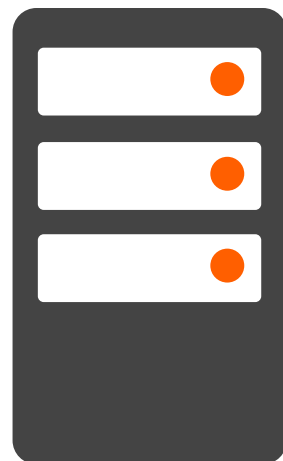
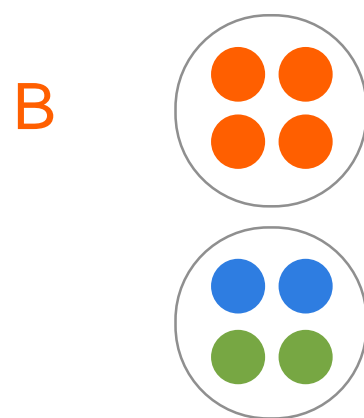
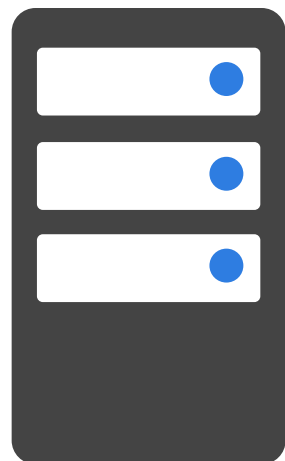
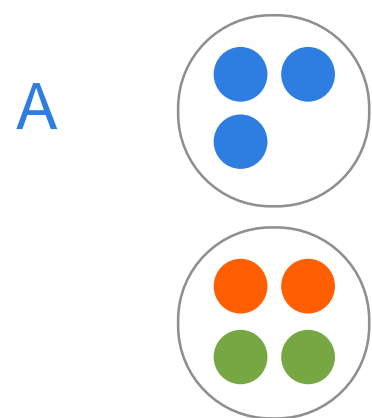
D



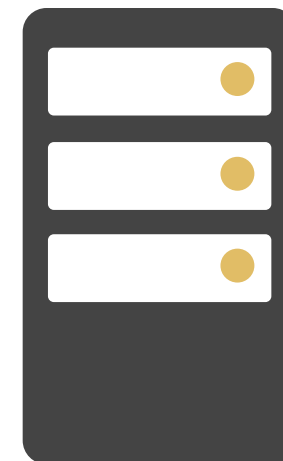
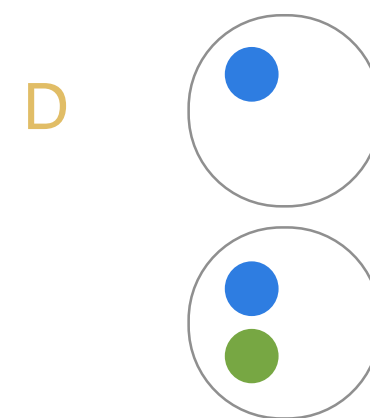
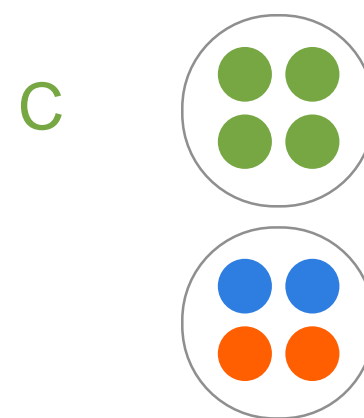
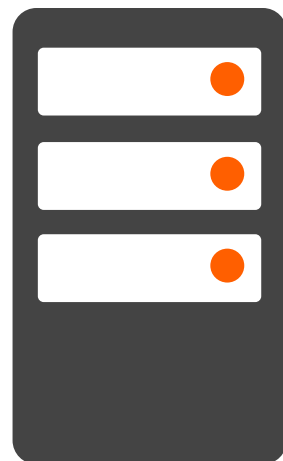
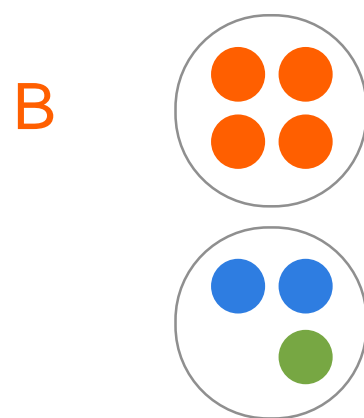
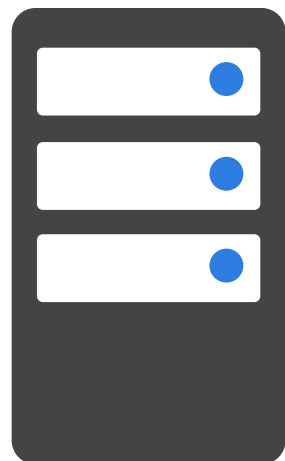
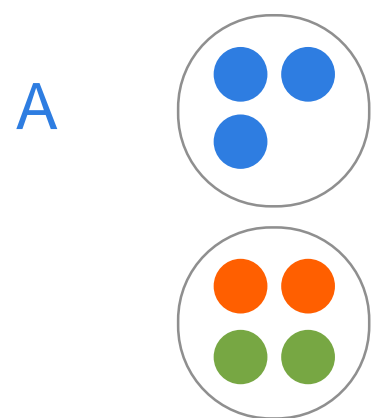
Migration



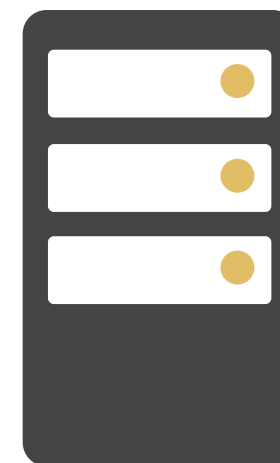
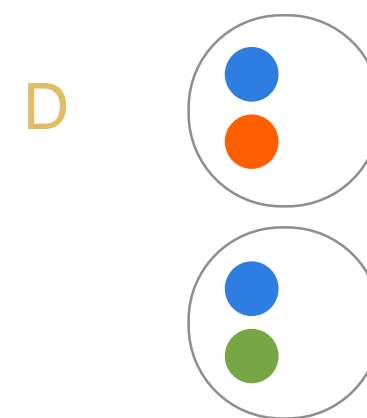
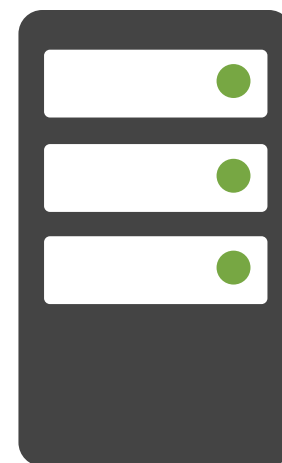
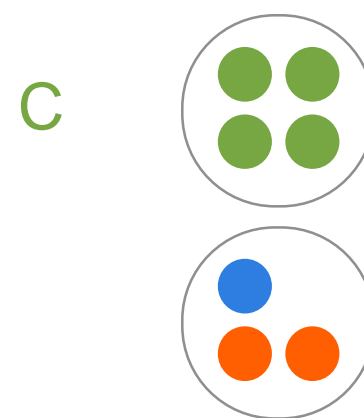
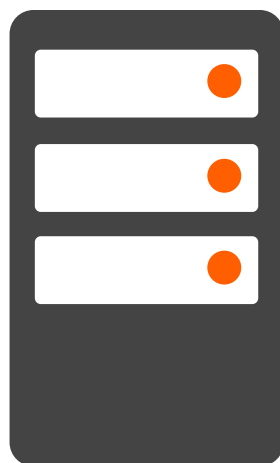
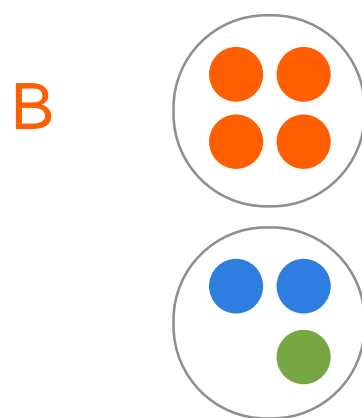
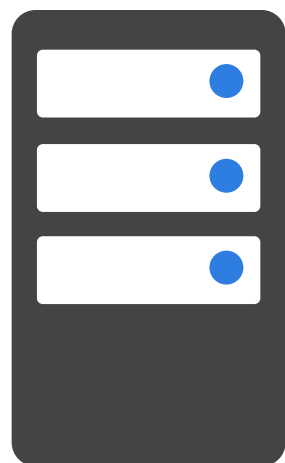
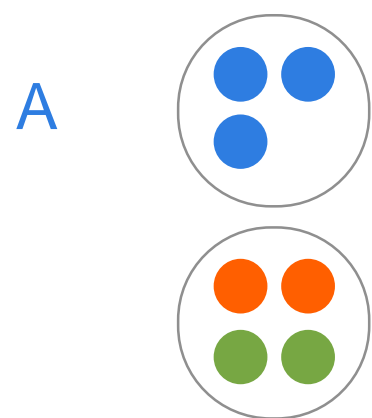
Migration



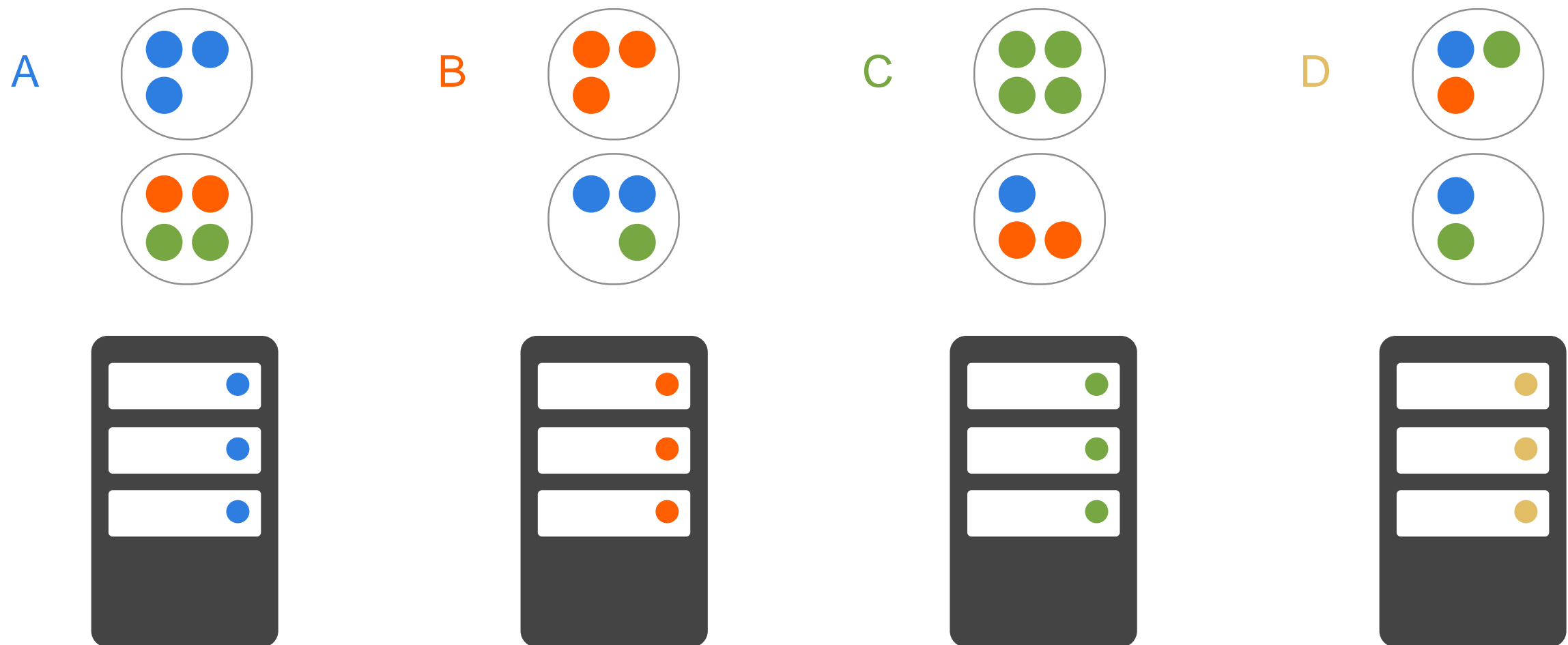
Migration



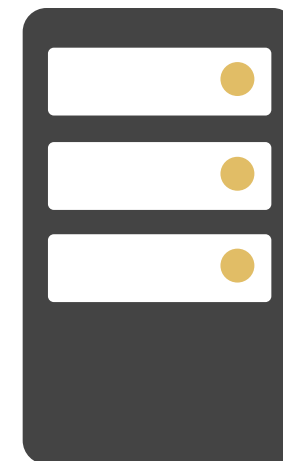
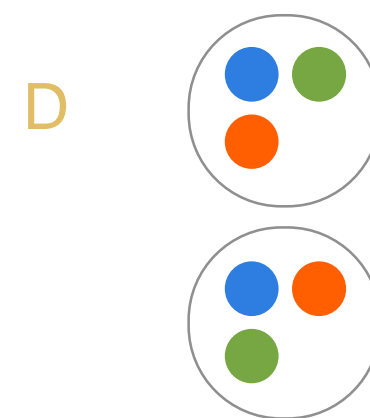
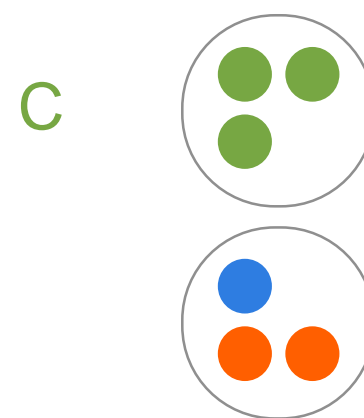
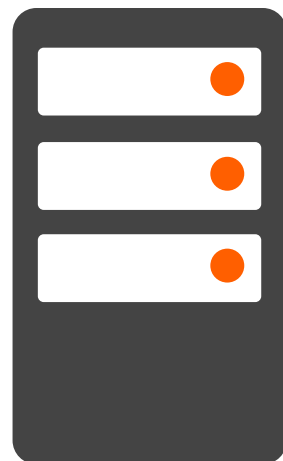
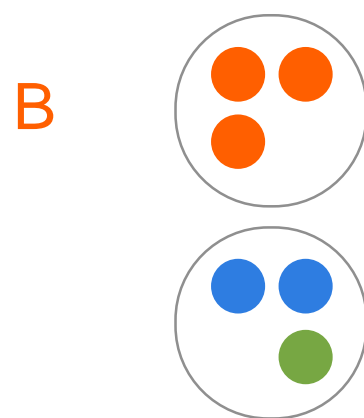
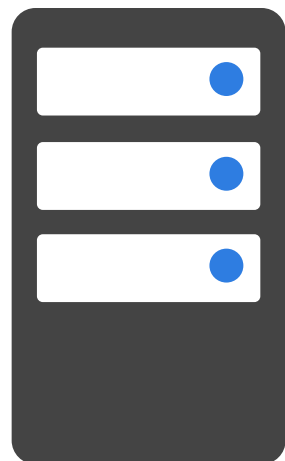
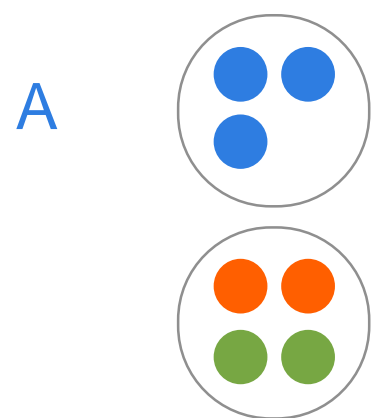
Migration



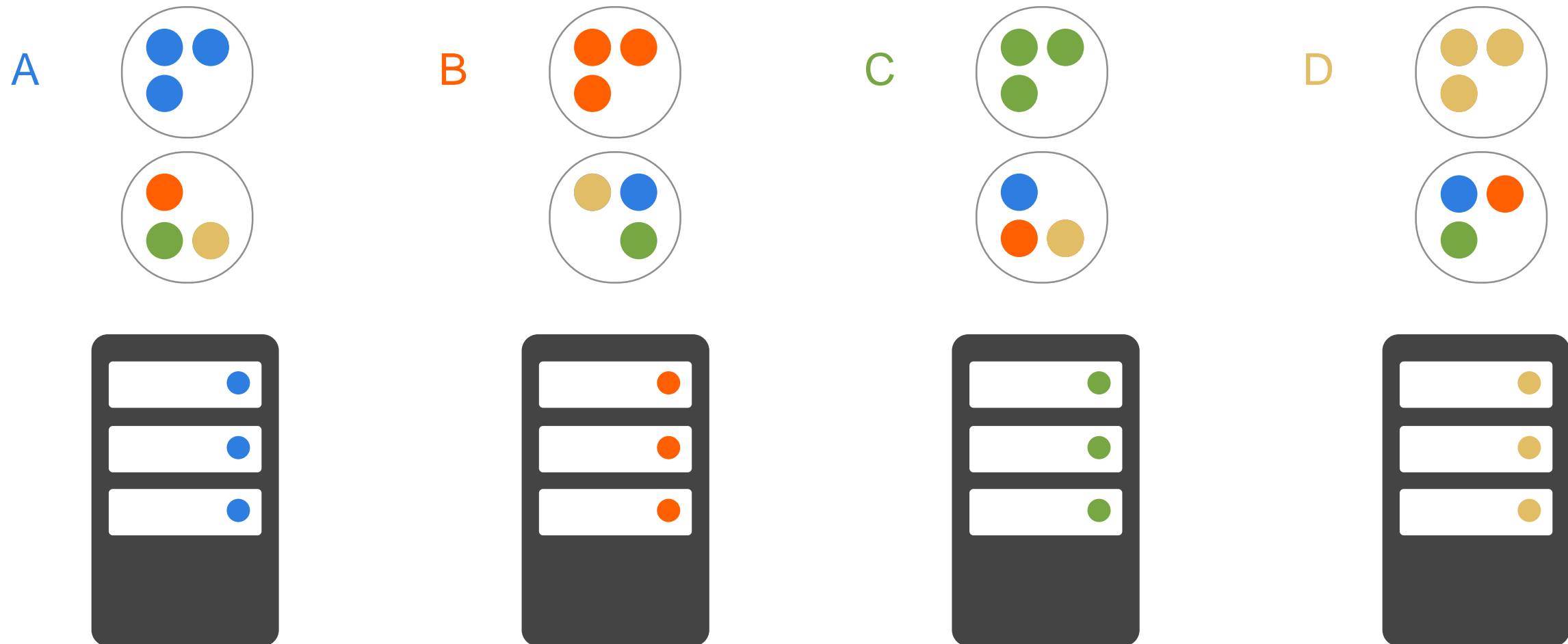
Migration



Migration



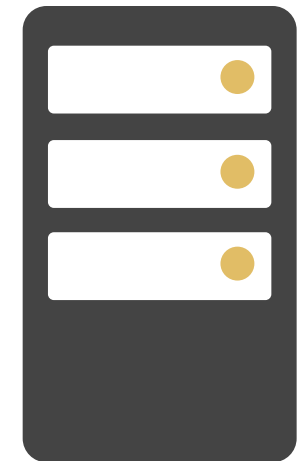
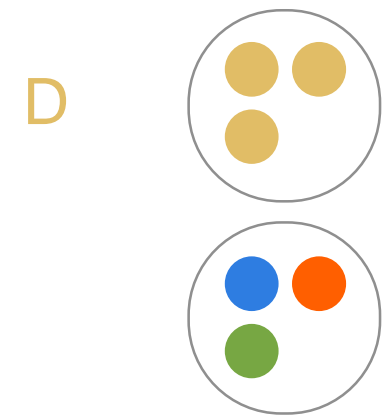
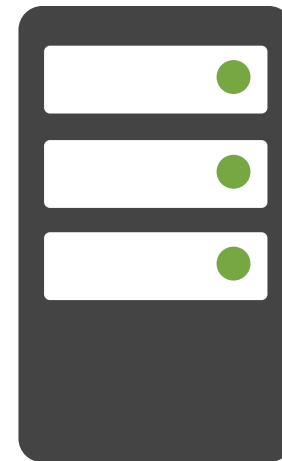
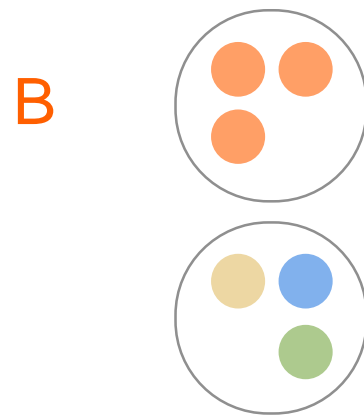
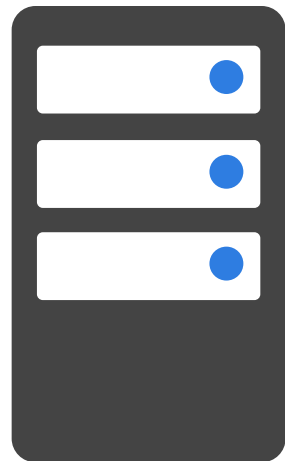
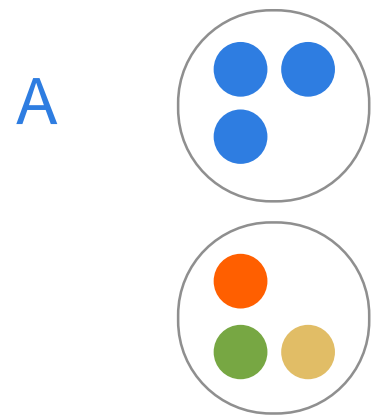
Migration Complete



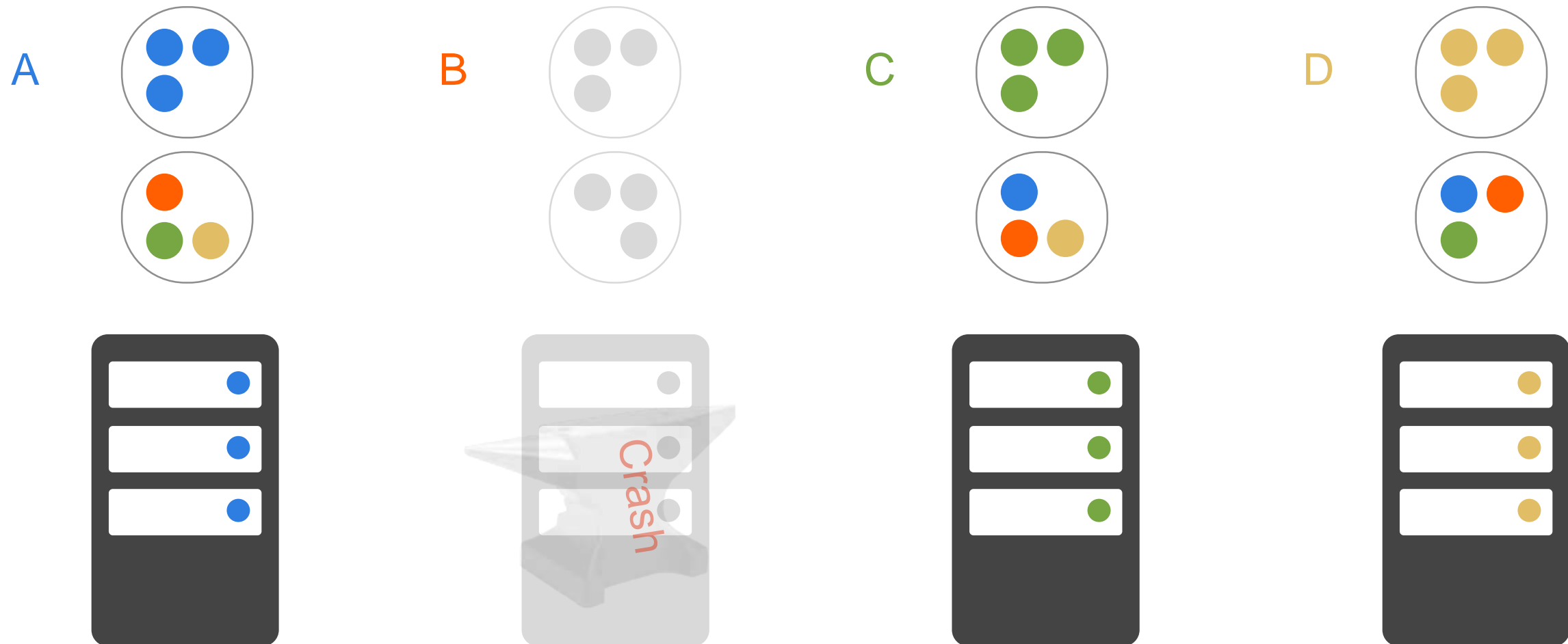


Data Safety when Node Dies

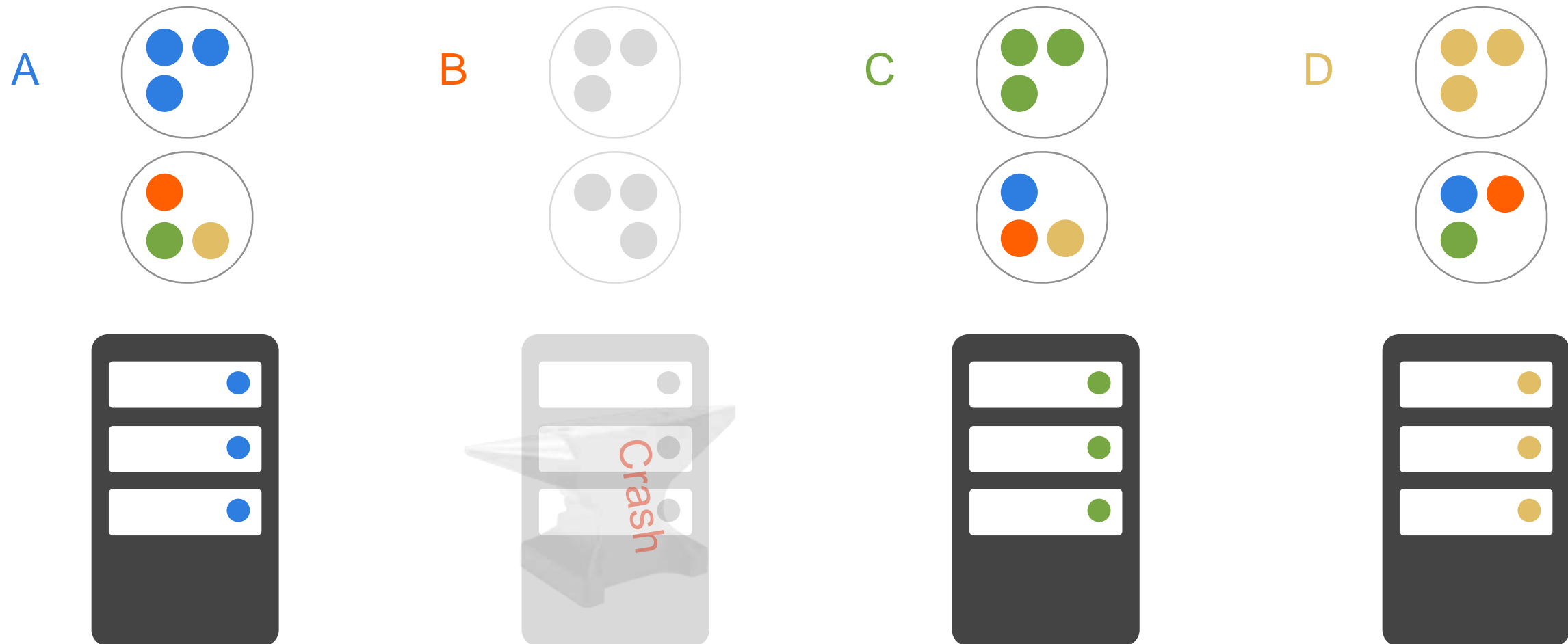
Node Crashes



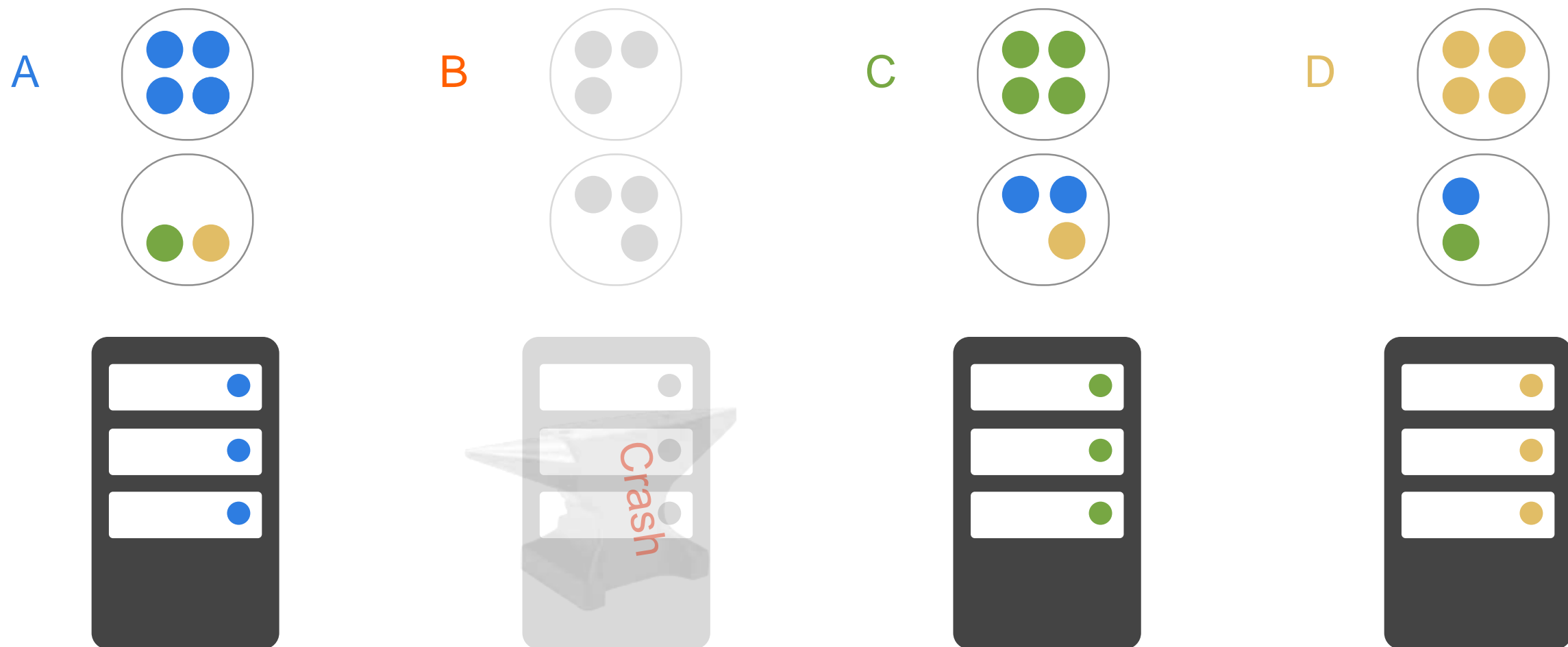
Backups Are Restored



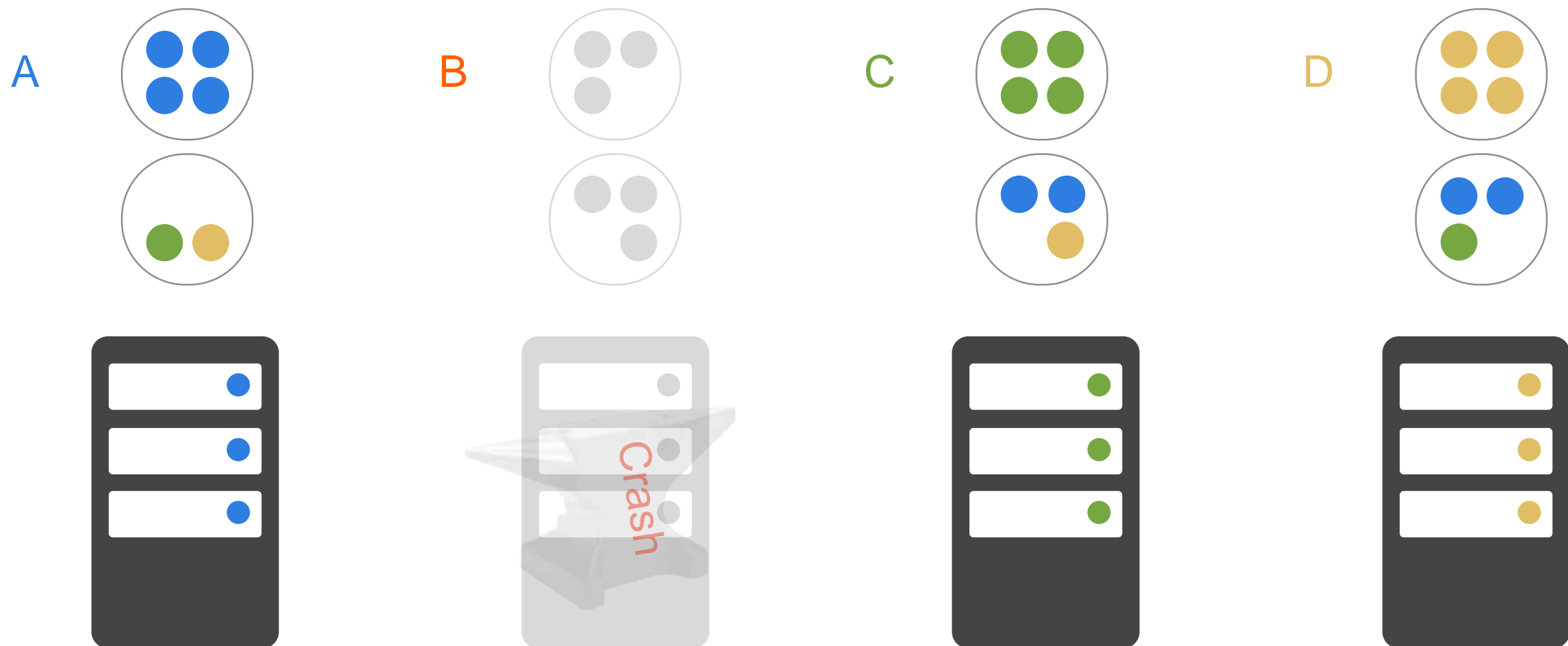
Backups Are Restored



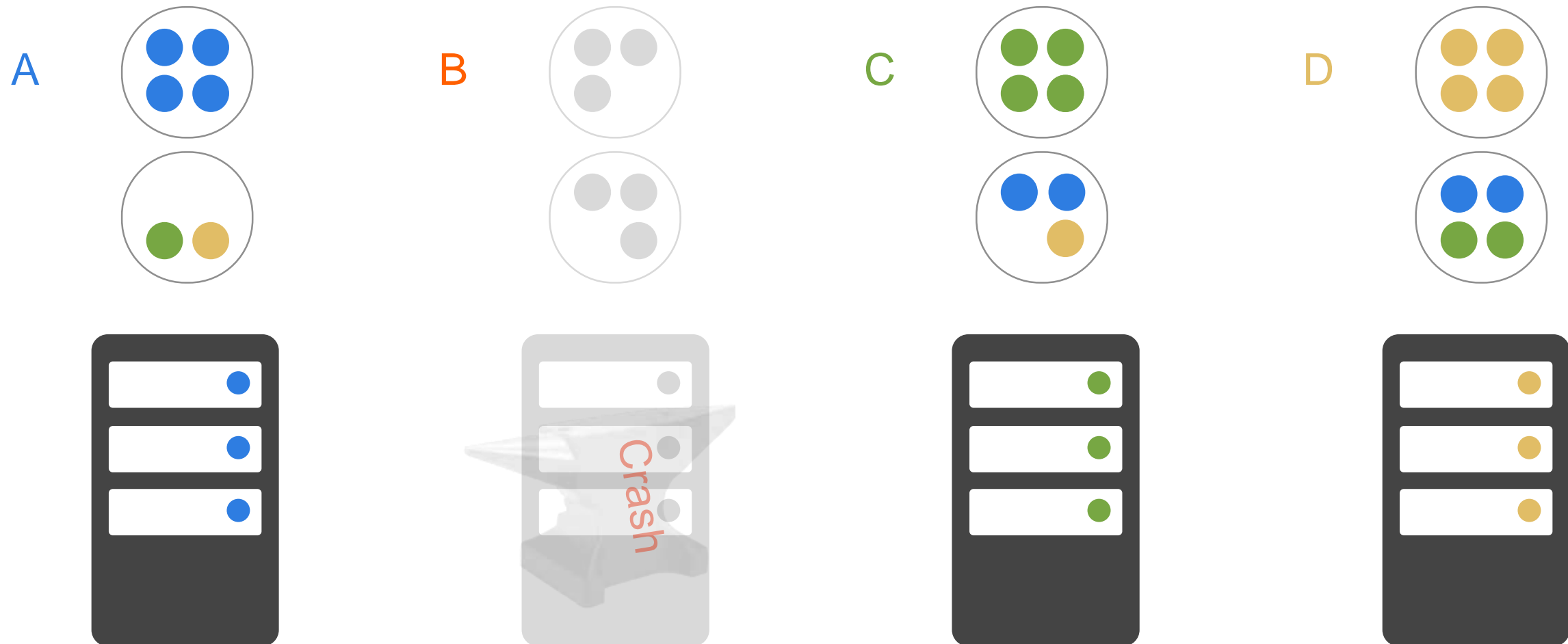
Backups Are Restored



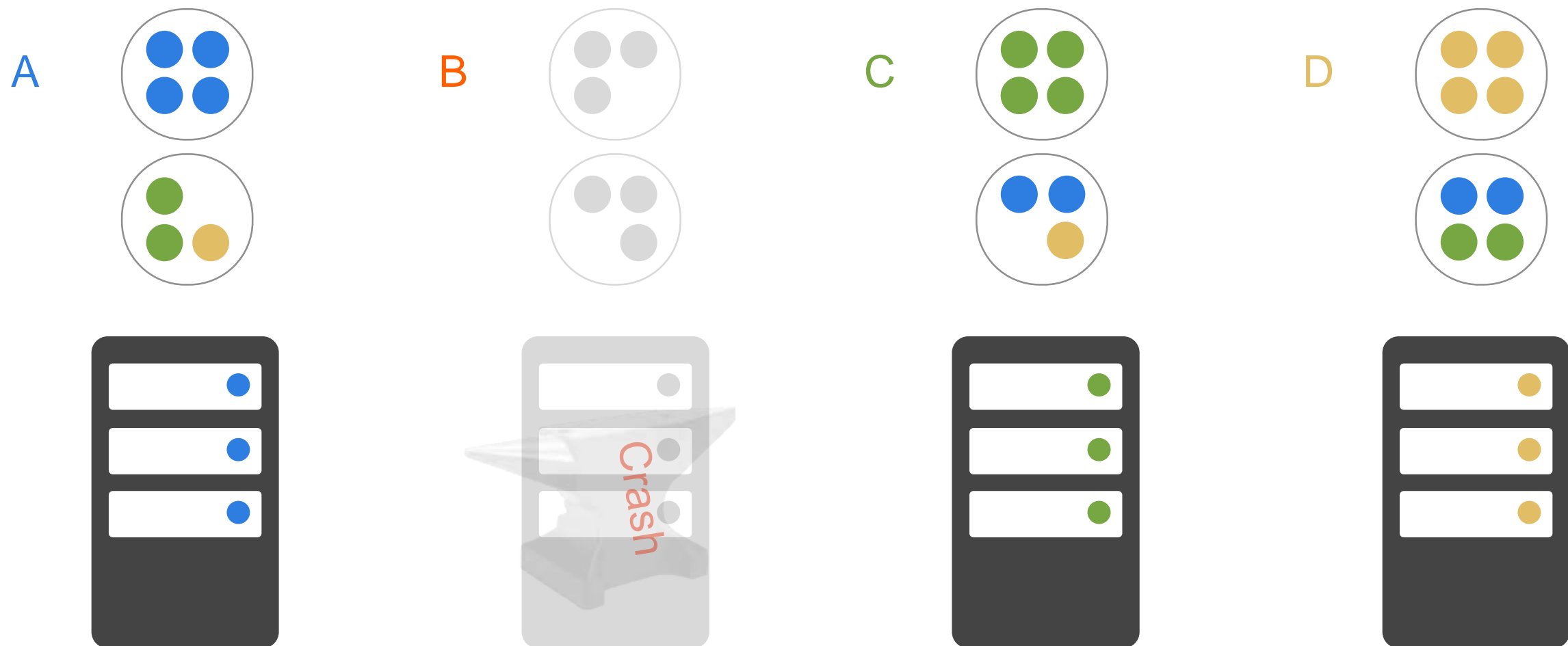
Backups Are Restored



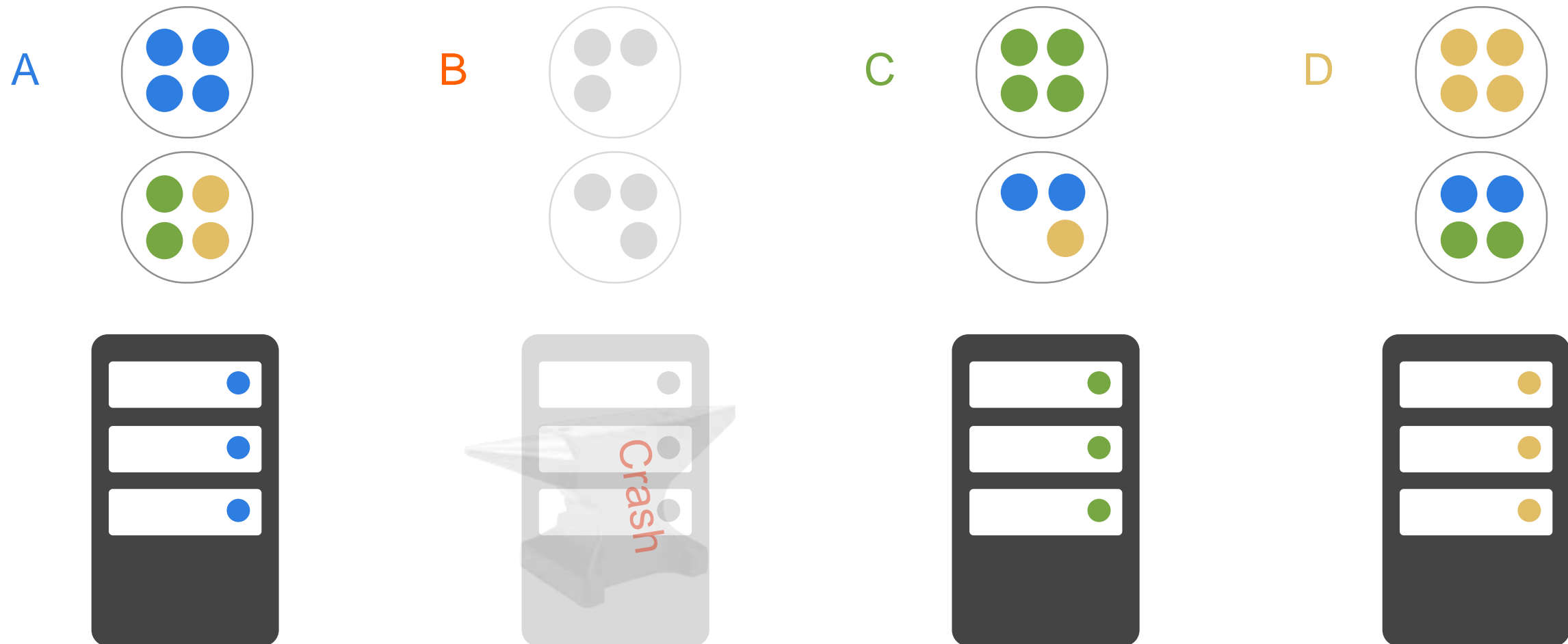
Backups Are Restored



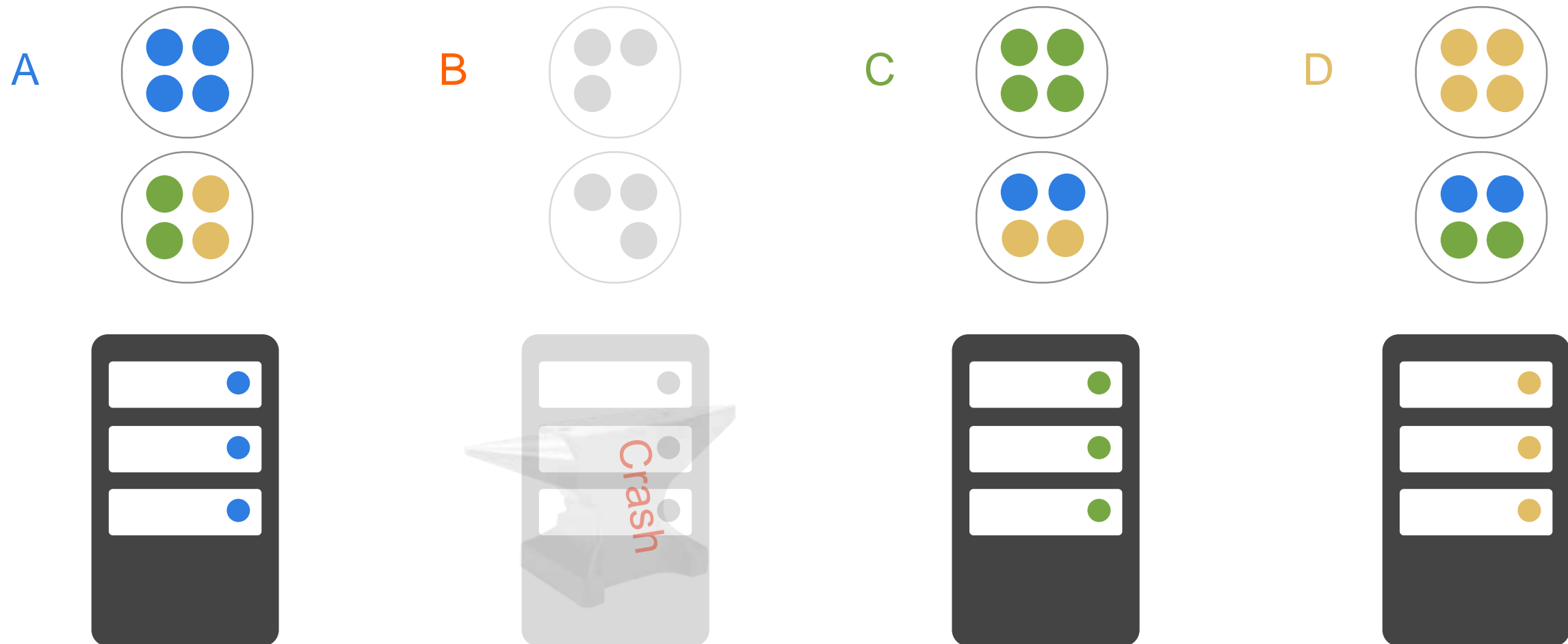
Backups Are Restored



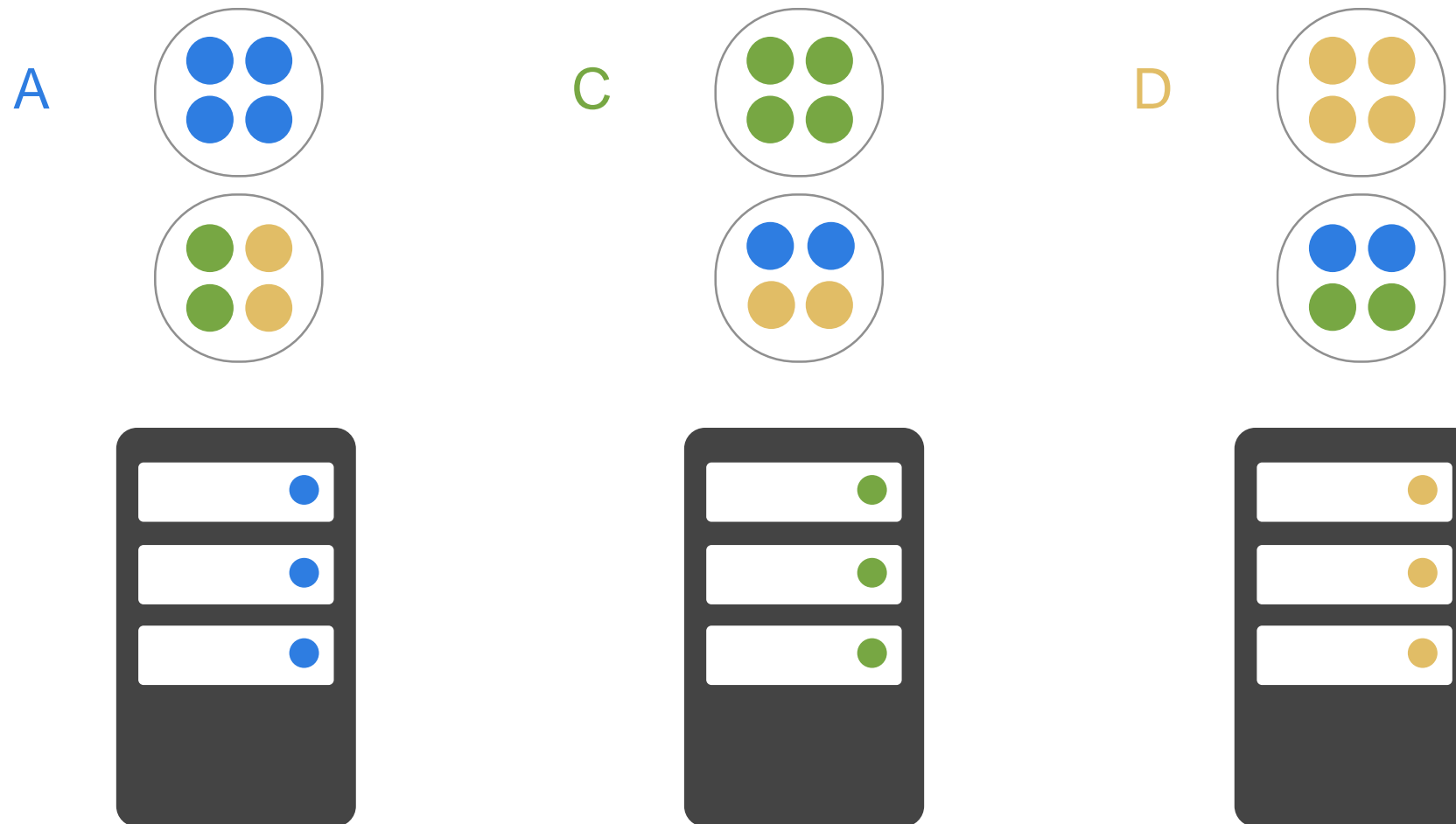
Backups Are Restored



Backups Are Restored



Recovery Is Complete

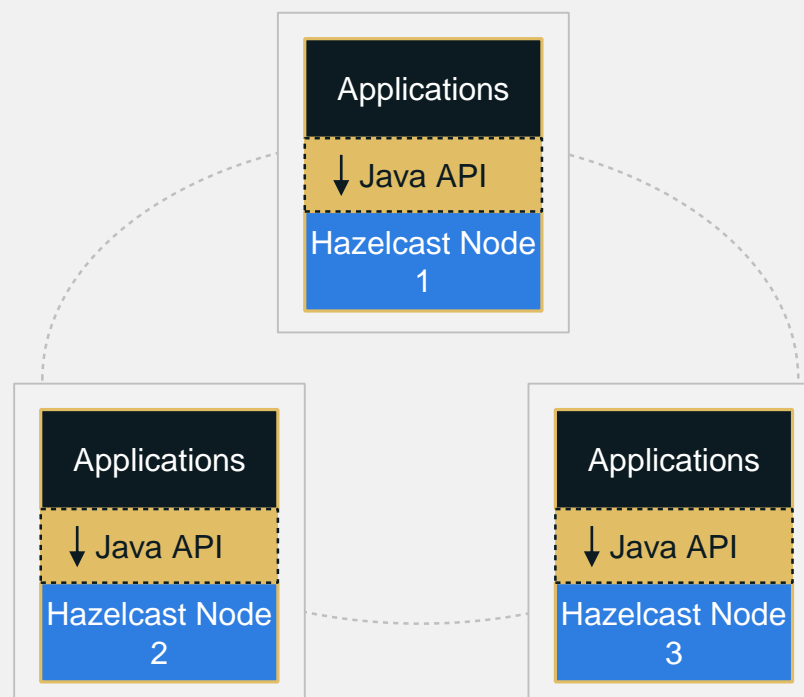


Deployment Options

- ▶ Client/Server Architecture
- ▶ Embedded Architecture

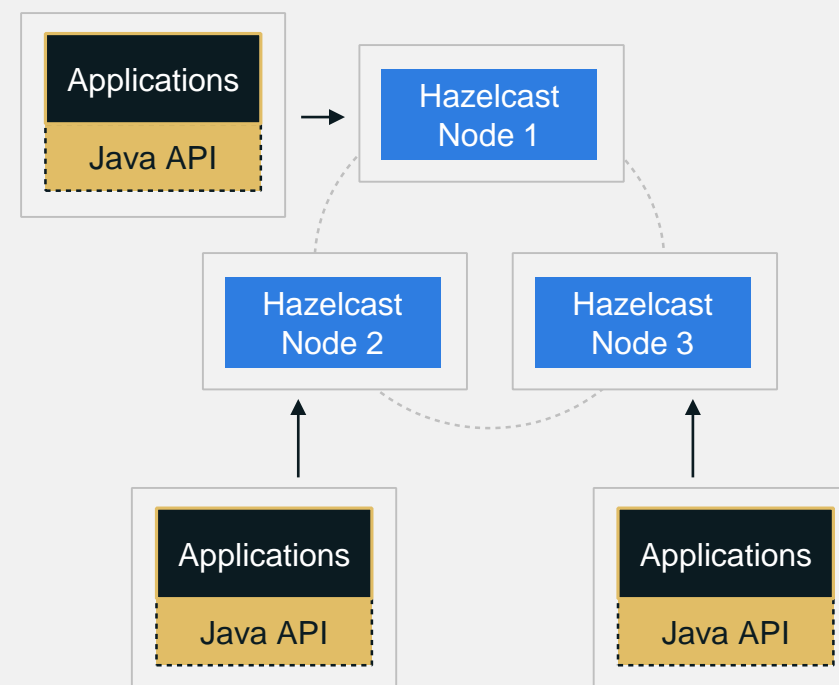
Deployment Options

Embedded Hazelcast



Great for early stages of rapid application development and iteration

Client-Server Mode



Necessary for scale up or scale out deployments – decouples upgrading of clients and cluster for long term TCO

Networking Options

- ▶ Multicast (default)
- ▶ TCP/IP
- ▶ AWS

Hazelcast Features

Data Store Features

Java Collection API: Map, List, Set, Queue

JCache

High Density Memory Store

Hibernate 2nd Level Cache

Web Session Replication: Tomcat, Jetty

Predicate API: Indexes, SQL Query

Persistence: Map/Queue Store & Loader. Write Behind/Through

Eviction

Near Cache

Transactions: Local & XA

WAN Replication

Memcached Interface

Distributed Computing Features

Java Concurrency API

(Lock, Semaphore, AtomicLong, AtomicReference, Executor Service, Blocking Queue)

Entry and Item Listeners

Entry Processor

Aggregators

Map/Reduce

Data Affinity

Continues Query

Map Interceptors

Delta Update

Distributed Messaging Features

Queue

Topic (Pub/Sub)

Event Listeners

Ring Buffers

Hazelcast Integration Modules

- ▶ Spring Cache Manager
- ▶ Hibernate 2nd Level Cache Provider
- ▶ Web Session Replication
- ▶ OSGI Support

Hazelcast Enterprise Features

- ▶ Management Center (free up to 2 nodes)
- ▶ High-Density Memory
- ▶ Tomcat/Jetty Session Replication
- ▶ Enterprise WAN Replication
- ▶ Security
- ▶ Native Clients (.NET/C++)

Thank you ! :)

any questions ?

 emrah@hazelcast.com



<http://www.zenika.com/formation-hazelcast-essentials.html>

30Th November - Free Training

