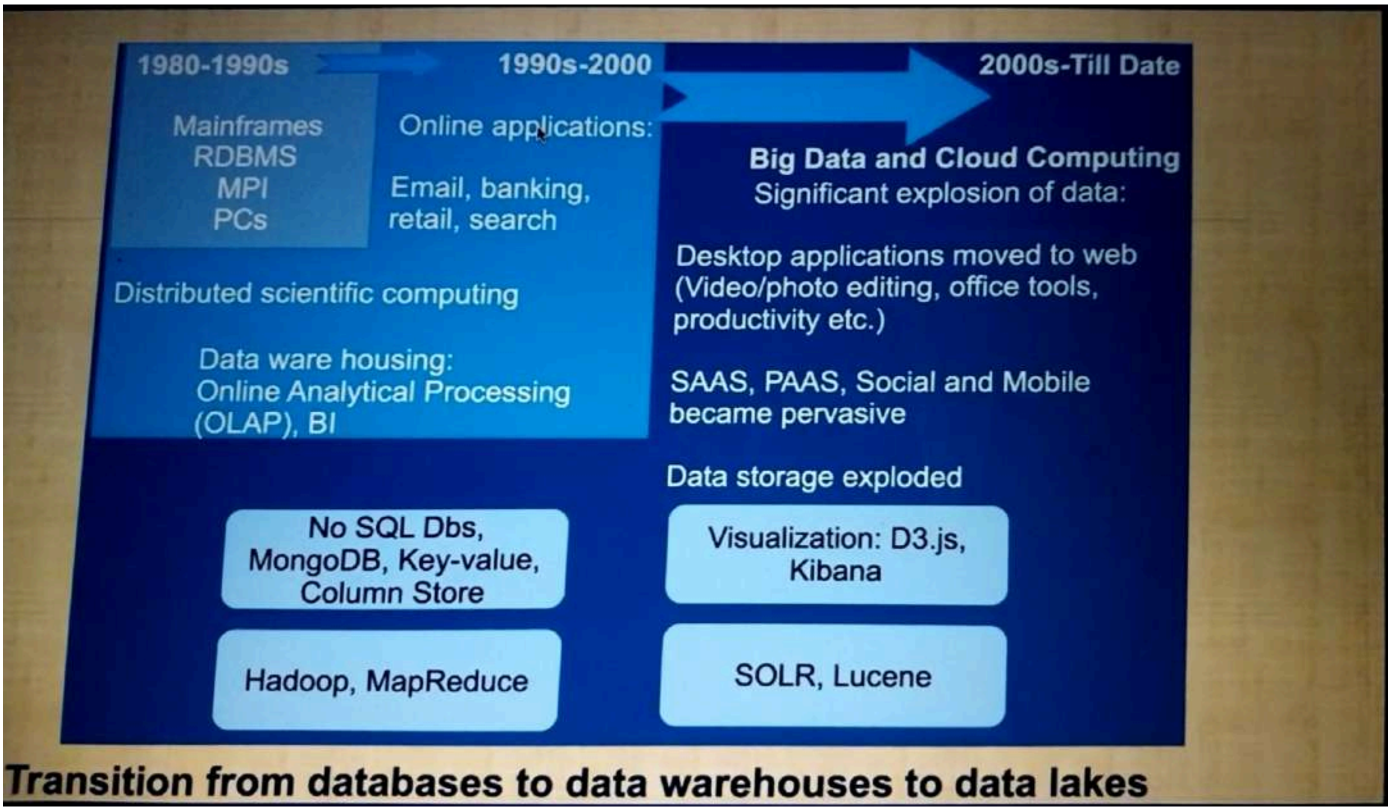


1980-1990s

Mainframes
RDBMS
MPI
PCs

1990s-2000

Online applications:

Email, banking,
retail, search

Distributed scientific computing

Data ware housing:
Online Analytical Processing
(OLAP), BI

2000s-Till Date

Big Data and Cloud Computing
Significant explosion of data:

Desktop applications moved to web
(Video/photo editing, office tools,
productivity etc.)

SAAS, PAAS, Social and Mobile
became pervasive

Data storage exploded

No SQL Dbs,
MongoDB, Key-value,
Column Store

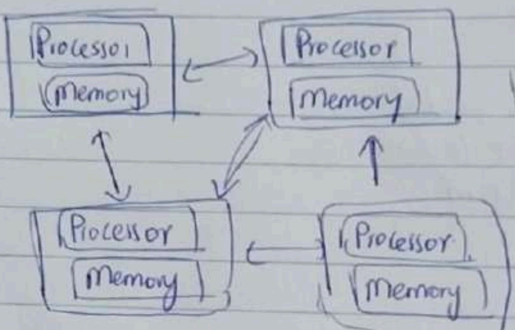
Visualization: D3.js,
Kibana

Hadoop, MapReduce

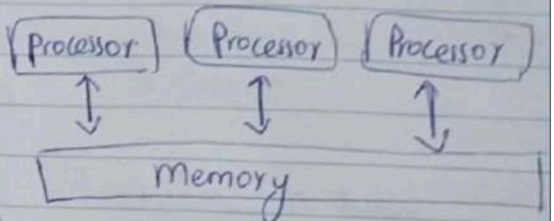
SOLR, Lucene

Transition from databases to data warehouses to data lakes

* Curriculum Discussion -> * Big Data - Hadoop & spark
* Cloud -> Azure & GDP
-> Devop Tools -> Git, Github & Docker.
-> No.SQL - mangoDB. Tool



Distributed Computing

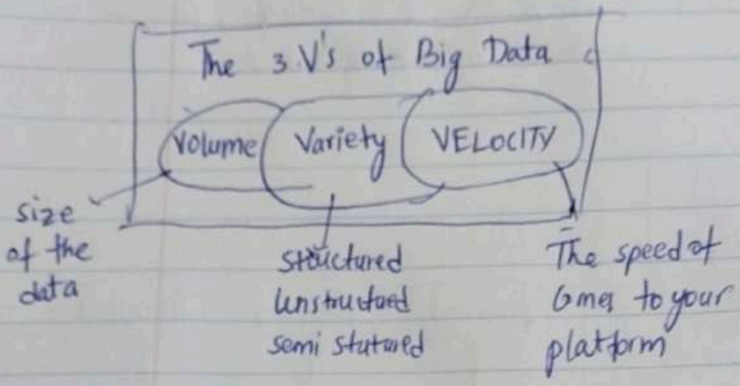


Parallel-computing

Cluster Computing

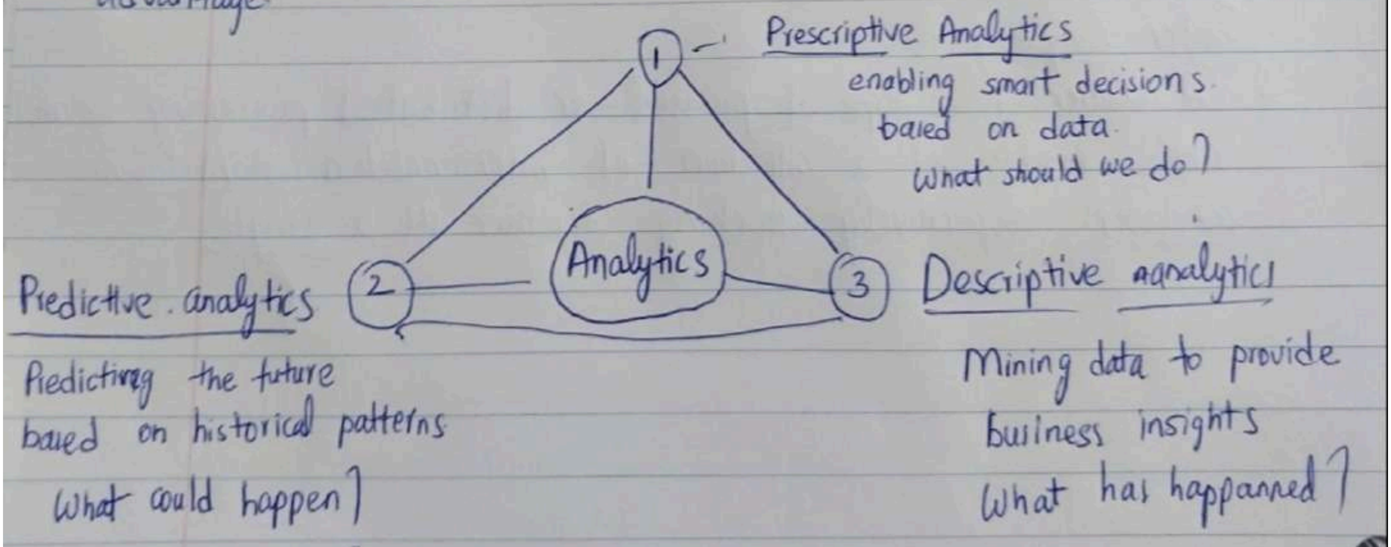
- A cluster is a type of parallel or distributed proccessing sy which consist of a collection of interconnected stand-alone computers cooperatively working together as a single

Big data : It ~~too~~ Data that is Too large & too complex for Conventional data tools to capture, store & analyze



What is analytics?

The scientific process of transforming data into Insight for making better decisions, offering new opportunities for a competative advantage.



Prescriptive Analytics

enabling smart decisions based on data

What should we do?

Predictive analytics

Predicting the future based on historical patterns

What could happen?

Descriptive analytics

Mining data to provide business insights

What has happanned?

Why do airlane prices change every hour?

Ans - Prescriptive analytics
(advice on possible outcomes).

How do grocery cashiers know to hand you coupons you might actually use?

Ans Predictive Analytics
(understanding the future).

How does Netflix frequently recommend just the right movie?

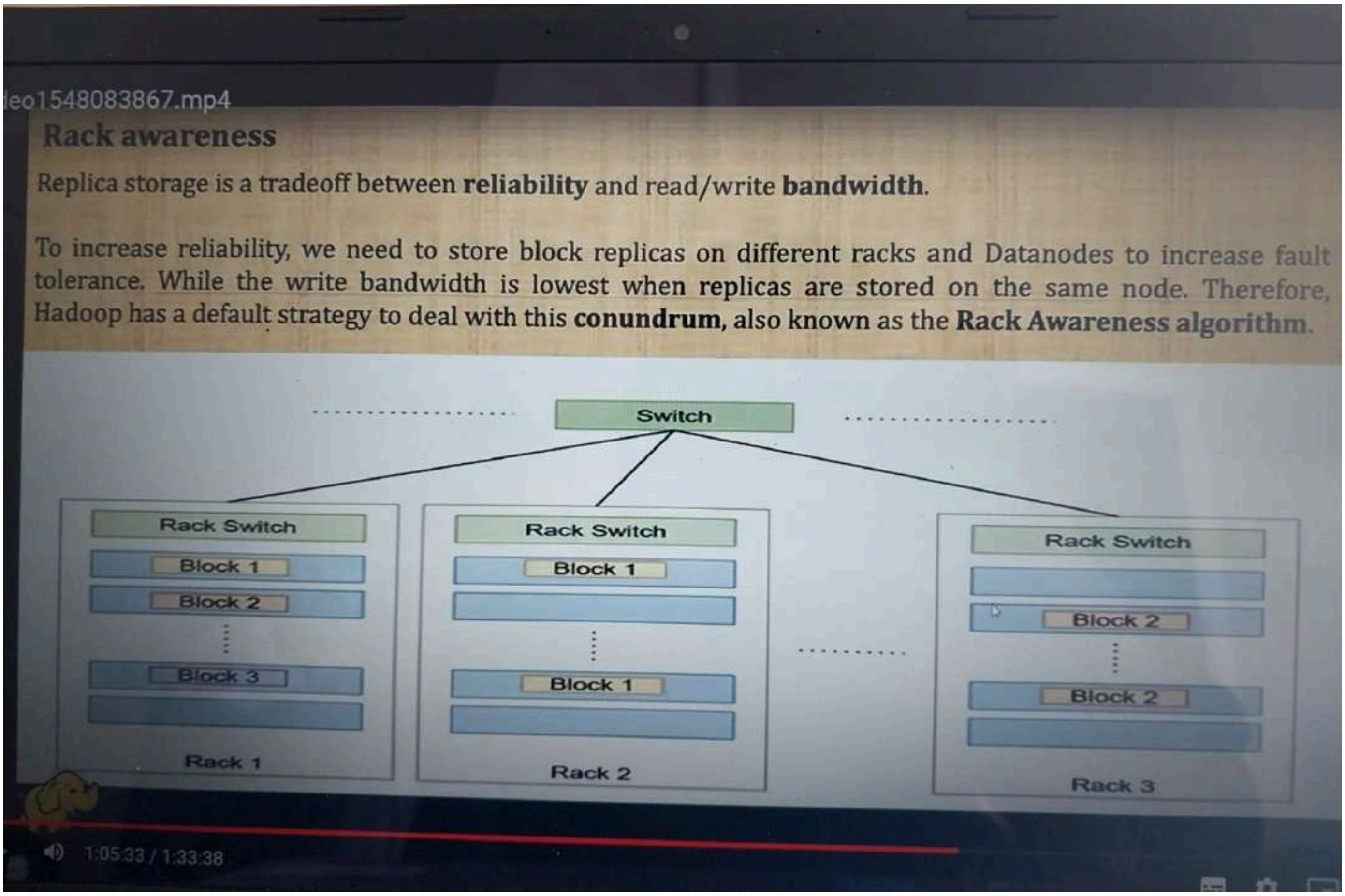
Descriptive Analytics
(insight into the past).

Problems with Traditional approach

1) Storing huge and exponentially growing datasets.

2). Processing data having complex structure (structured, un-structured, -semi-structured)

3) Bringing huge amount of data to computation unit becomes a bottleneck

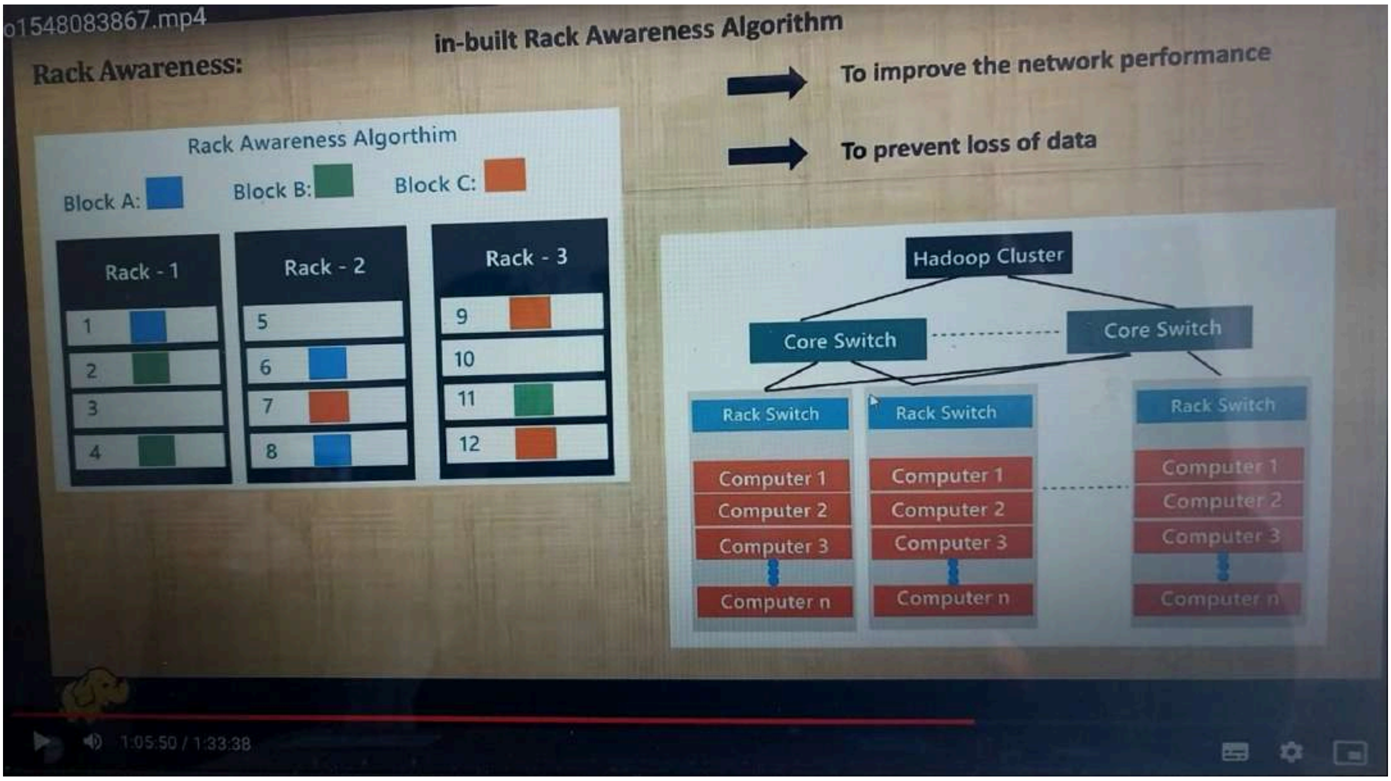


deo1548083867.mp4

Rack awareness

Replica storage is a tradeoff between **reliability** and read/write **bandwidth**.

To increase reliability, we need to store block replicas on different racks and Datanodes to increase fault tolerance. While the write bandwidth is lowest when **replicas are stored** on the same node. Therefore, Hadoop has a default strategy to deal with this **conundrum**, also known as the **Rack Awareness algorithm**.



o1548083867.mp4

Rack Awareness:

in-built Rack Awareness Algorithm

To improve the network performance

To prevent loss of data

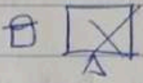
Rack Awareness Algorthim

Block A: 🟦 Block B: 🟩 Block C: 🟧

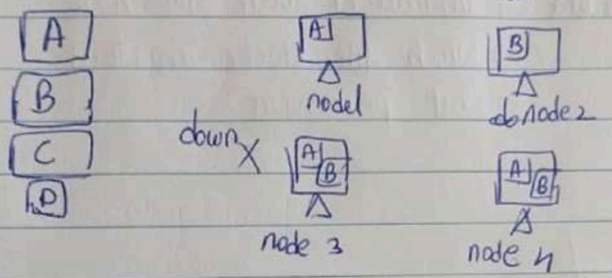
Rack - 1	Rack - 2	Rack - 3
1 🟦	5	9 🟧
2 🟩	6 🟦	10
3	7 🟧	11 🟩
4 🟩	8 🟦	12 🟧

(The data drawing copies) → by default it has 3copies

2). Replication → TA → Hadoop → 3. copies



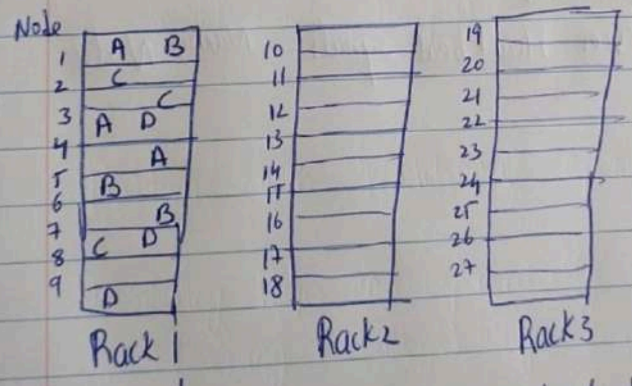
→ Data as loost.



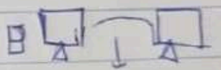
→ How replication works)

Rack awarness Algorithm → Replication → Data Failure
→ Network Performance

A B C D



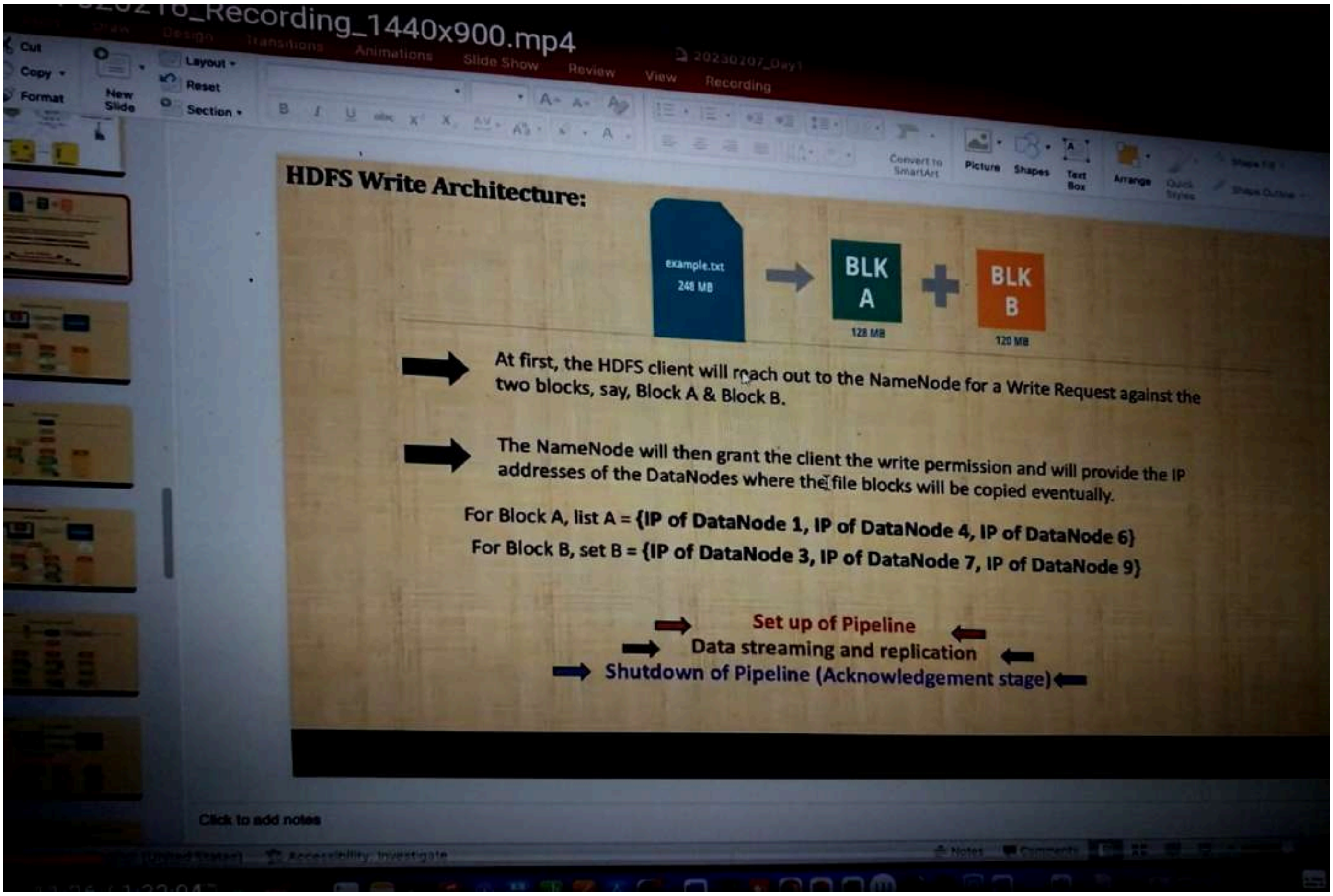
N/w



performance go-down

↳ The performance is high ✓ but if we lost Rack1 we can lose total data so it will store atmost 2 Rack

Rock Awarness algo → ~~set~~ atmost -2 Rocks.



HDFS Write Architecture:

At first, the HDFS client will reach out to the NameNode for a Write Request against the two blocks, say, Block A & Block B.

The NameNode will then grant the client the write permission and will provide the IP addresses of the DataNodes where the file blocks will be copied eventually.

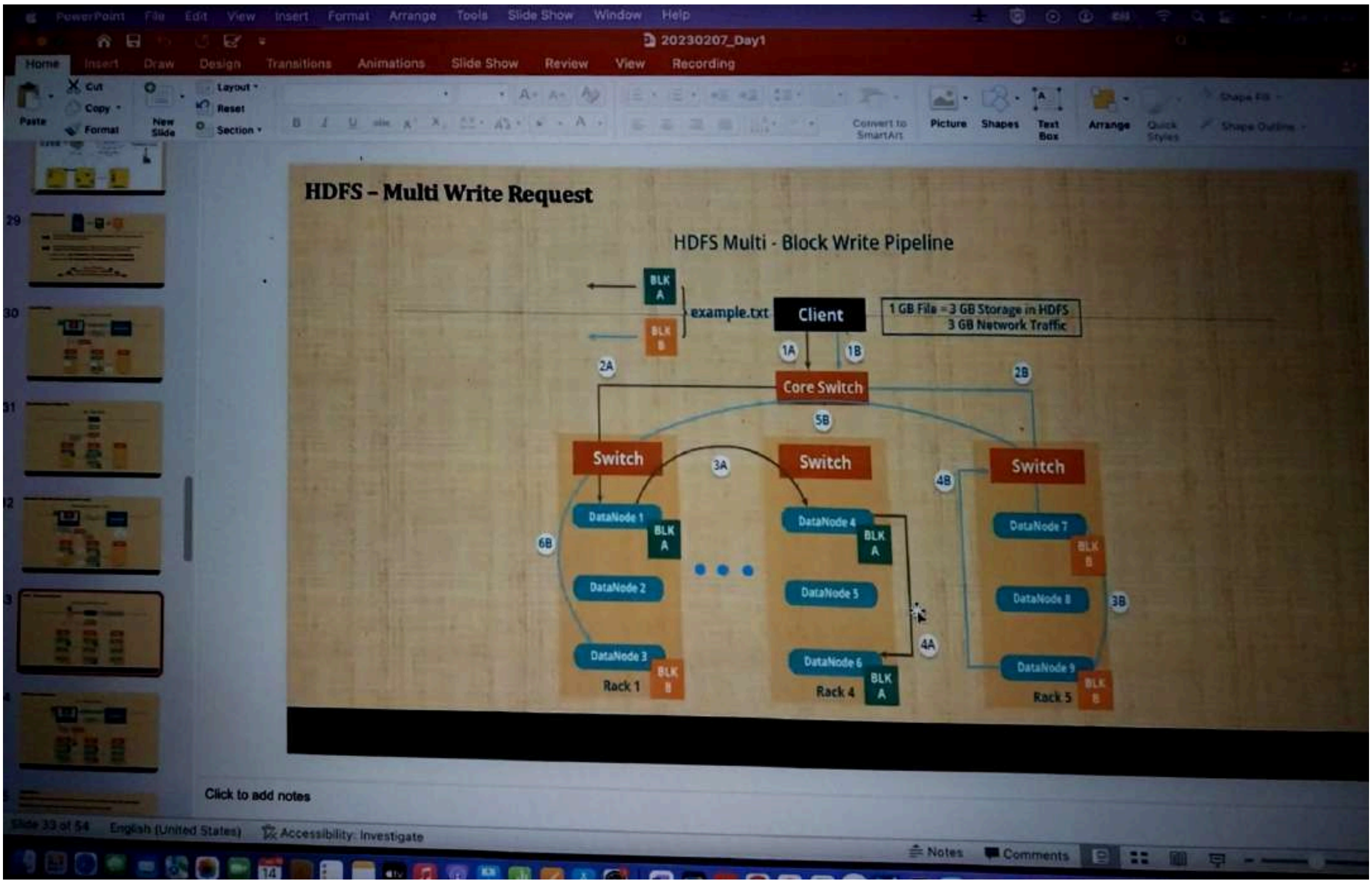
For Block A, list A = **{IP of DataNode 1, IP of DataNode 4, IP of DataNode 6}**

For Block B, set B = **{IP of DataNode 3, IP of DataNode 7, IP of DataNode 9}**

Set up of Pipeline

Data streaming and replication

Shutdown of Pipeline (Acknowledgement stage)



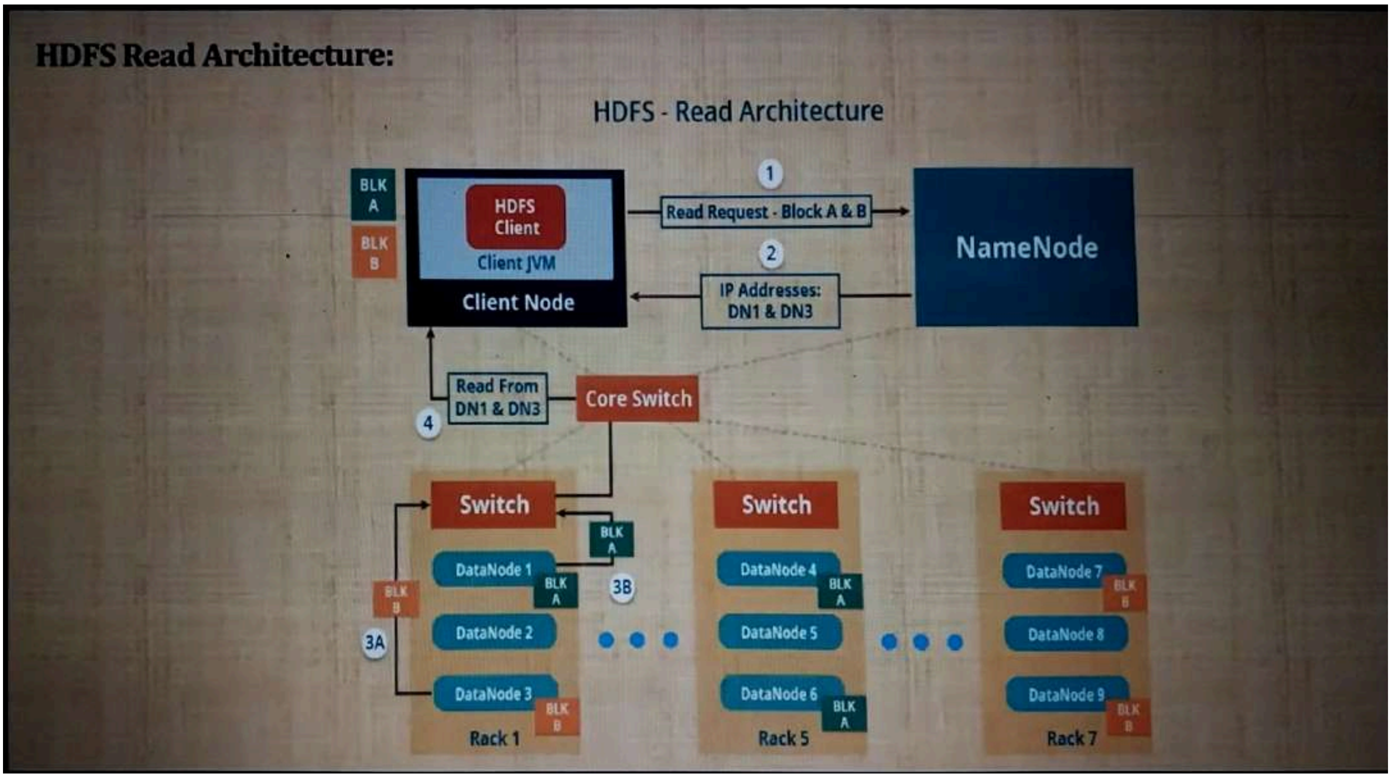
PowerPoint File Edit View Insert Format Arrange Tools Slide Show Window Help
20230207_Day1
Home Insert Draw Design Transitions Animations Slide Show Review View Recording
Paste
Cut
Copy
Format
New Slide
Layout
Reset
Section
Convert to SmartArt
Picture
Shapes
Text Box
Arrange
Quick Styles
Shape Fill
Shape Outline

HDFS – Multi Write Request

HDFS Multi - Block Write Pipeline

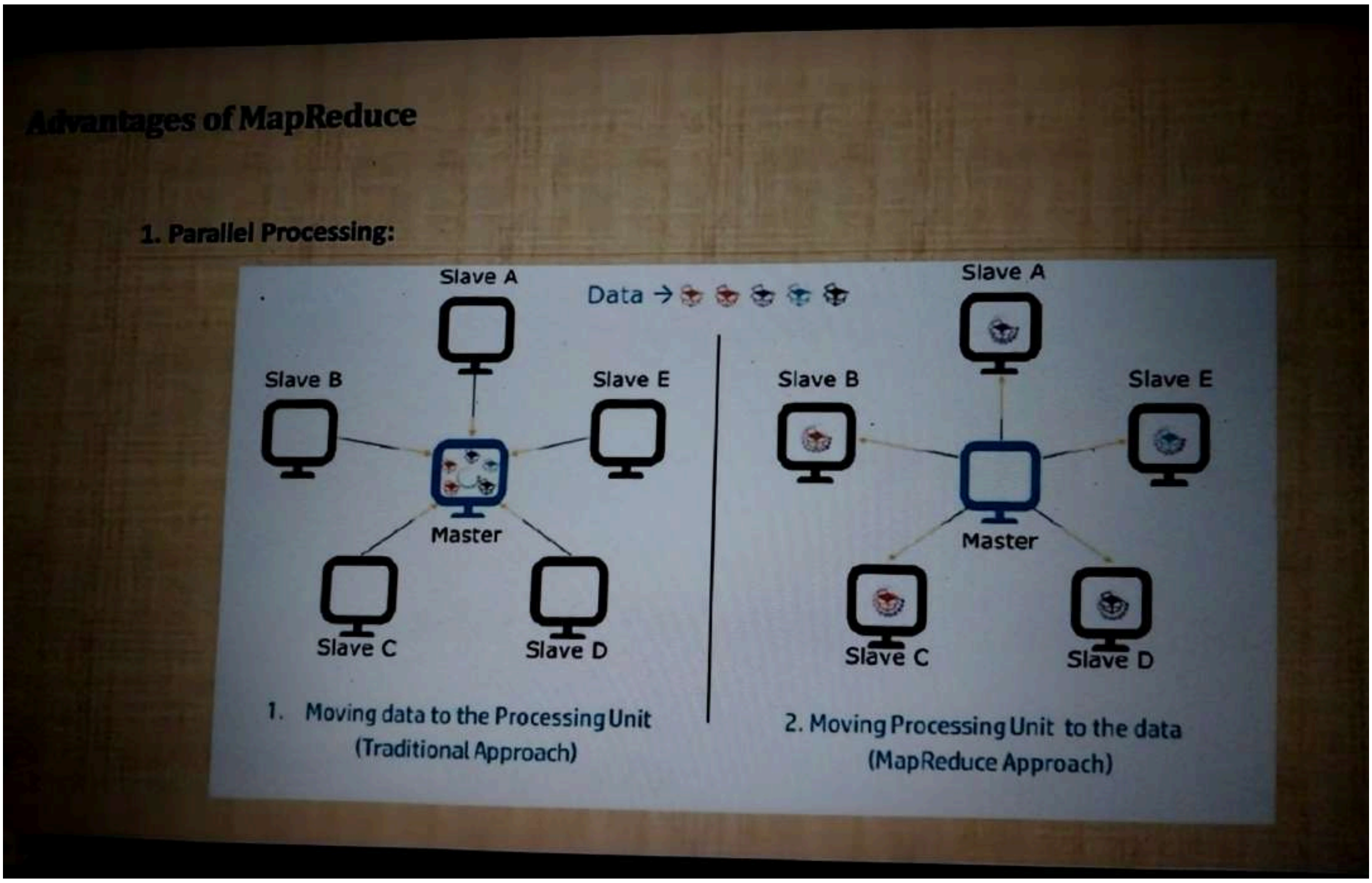
1 GB File = 3 GB Storage in HDFS
3 GB Network Traffic

Click to add notes



HDFS Read Architecture:

HDFS - Read Architecture



Advantages of MapReduce

1. Parallel Processing:

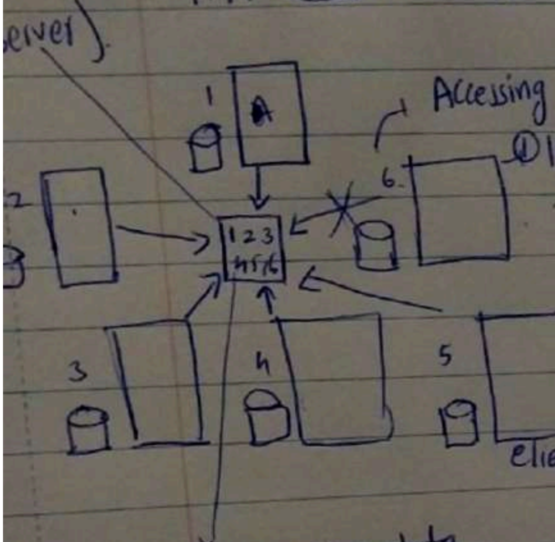
1. Moving data to the Processing Unit (Traditional Approach)

2. Moving Processing Unit to the data (MapReduce Approach)

Map Reduce :

T.A (Tradilatatal approch)

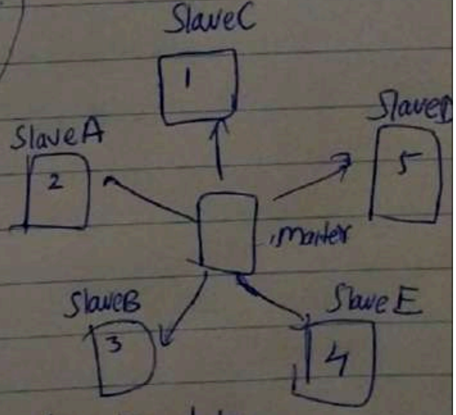
B.D.A (Big Data approch



① In the processing of trafer the data one machine to another machine anthing can happenned.

②

one single server process the entire data.



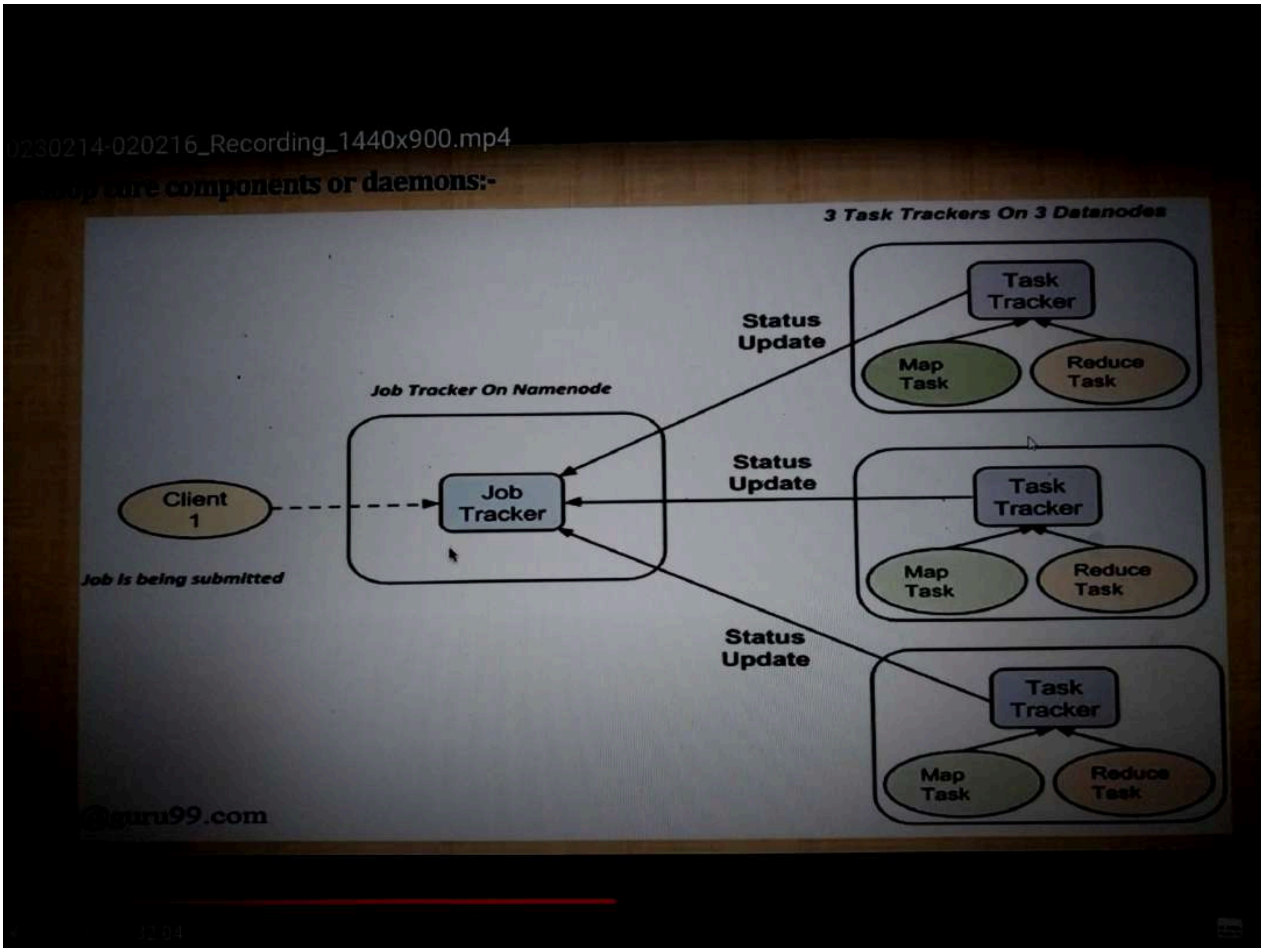
① here the data won't tranfer to one machine to another

Sloved)

② . The processing will take care of each slave.

stovd)

③ . It will only share the results of the Slave's



0230214-020216_Recording_1440x900.mp4

doop core components or daemons:-

3 Task Trackers On 3 Datanodes

Job Tracker On Namenode

Client 1

Job is being submitted

Job Tracker

Status Update

Status Update

Status Update

Task Tracker

Map Task

Reduce Task

Task Tracker

Map Task

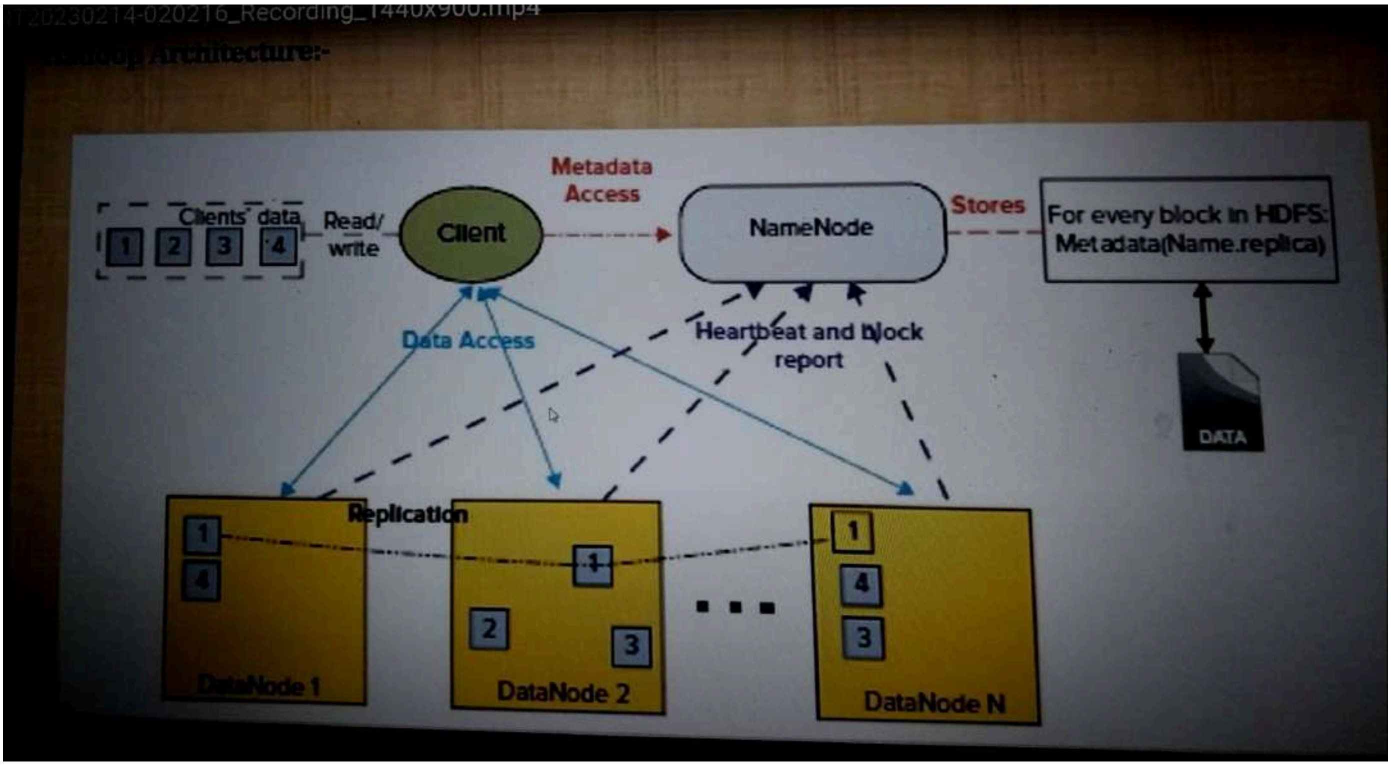
Reduce Task

Task Tracker

Map Task

Reduce Task

@guru99.com



20230214-020216_Recording_1440x900.mp4

Hadoop Architecture:-

Hadoop 1.X Limitation

Main drawback of Hadoop 1.X is that MapReduce Component in it's Architecture.

→ If the Name node server is down than total will down

→ It not suitable for Real-time Data Processing / Data Steaming

→ Job Tracker is the single point of failure

→ it runs only Map / Reduce jobs

Hadoop 2.X

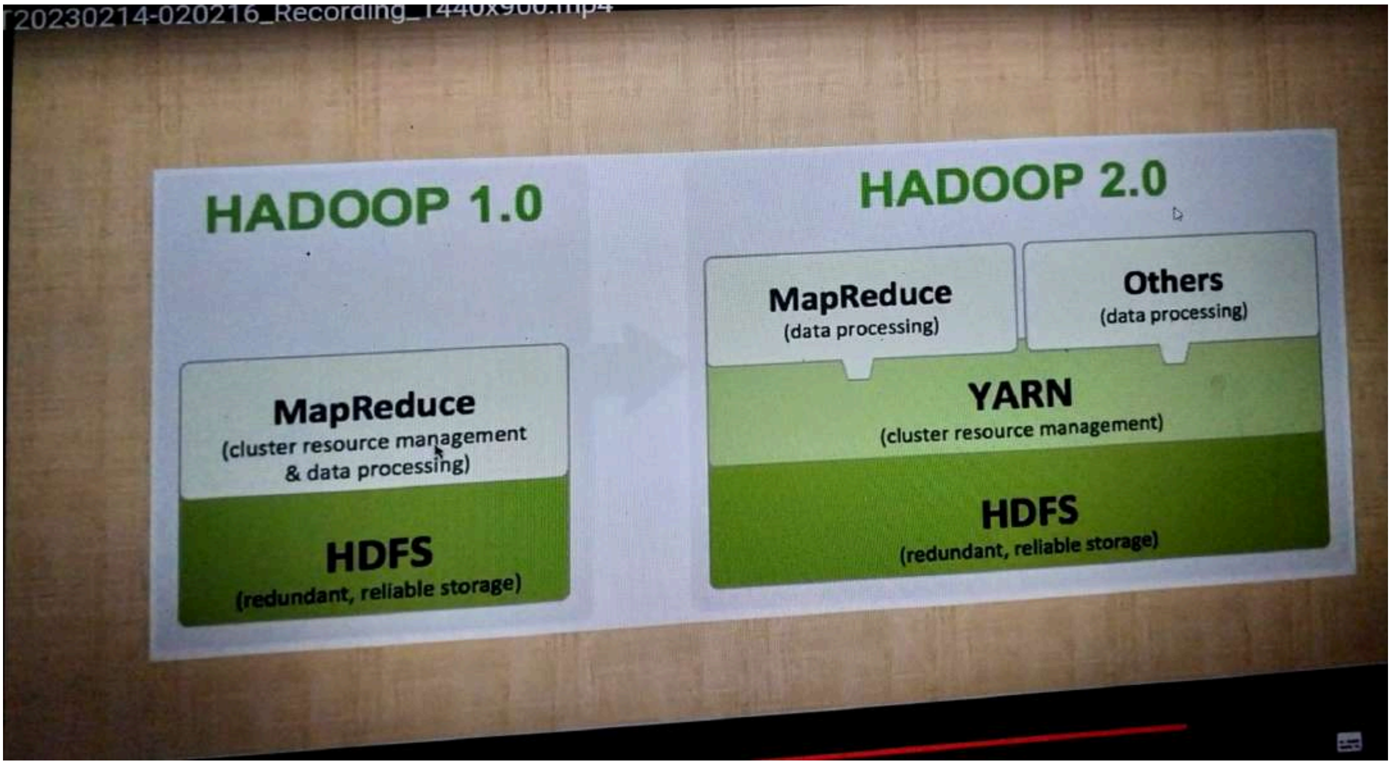
YARN (Yet Another Resource Negotiator)

* Hadoop2.X Allows to work in MR as well as other distributed Computing models like spark, Hama, Giraph, Message Passi

* 2.X Has better scalability, Scalable up to 10000 nodes per clu

* multiple Namenode servers manage multiple namespace

* Can serve as a platform for a wide variety of data analytics - possible to run event processing, streming and real-time operations.



T20230214-020216_Recording_1440x900.mp4

HADOOP 1.0

MapReduce
(cluster resource management & data processing)

HDFS
(redundant, reliable storage)

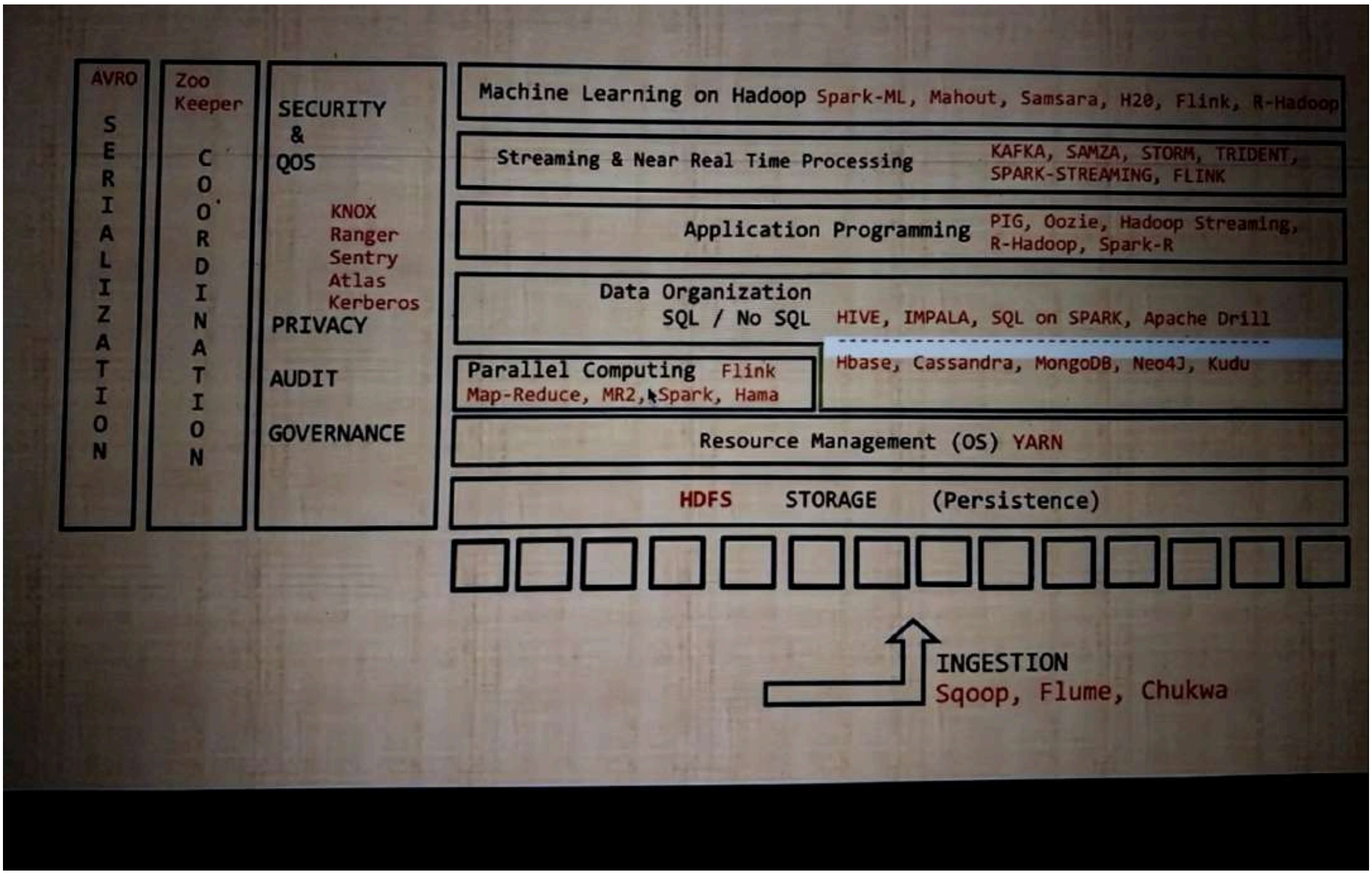
HADOOP 2.0

MapReduce
(data processing)

Others
(data processing)

YARN
(cluster resource management)

HDFS
(redundant, reliable storage)



AVRO

SERIALIZATION

Zoo Keeper

COORDINATION

SECURITY & QOS

KNOX
Ranger
Sentry
Atlas
Kerberos

PRIVACY

AUDIT

GOVERNANCE

Machine Learning on Hadoop Spark-ML, Mahout, Samsara, H2O, Flink, R-Hadoop

Streaming & Near Real Time Processing KAFKA, SAMZA, STORM, TRIDENT, SPARK-STREAMING, FLINK

Application Programming PIG, Oozie, Hadoop Streaming, R-Hadoop, Spark-R

Data Organization
SQL / No SQL HIVE, IMPALA, SQL on SPARK, Apache Drill

Hbase, Cassandra, MongoDB, Neo4J, Kudu

Parallel Computing Flink
Map-Reduce, MR2, Spark, Hama

Resource Management (OS) YARN

HDFS STORAGE (Persistence)

INGESTION
Sqoop, Flume, Chukwa

-1.x — Job tracker

↳ Resource manangment & processing

2.x.V YARN SPARK

Hadoop 2.X Version

↳ YARN → Resource management

↳ SPARK → Processing

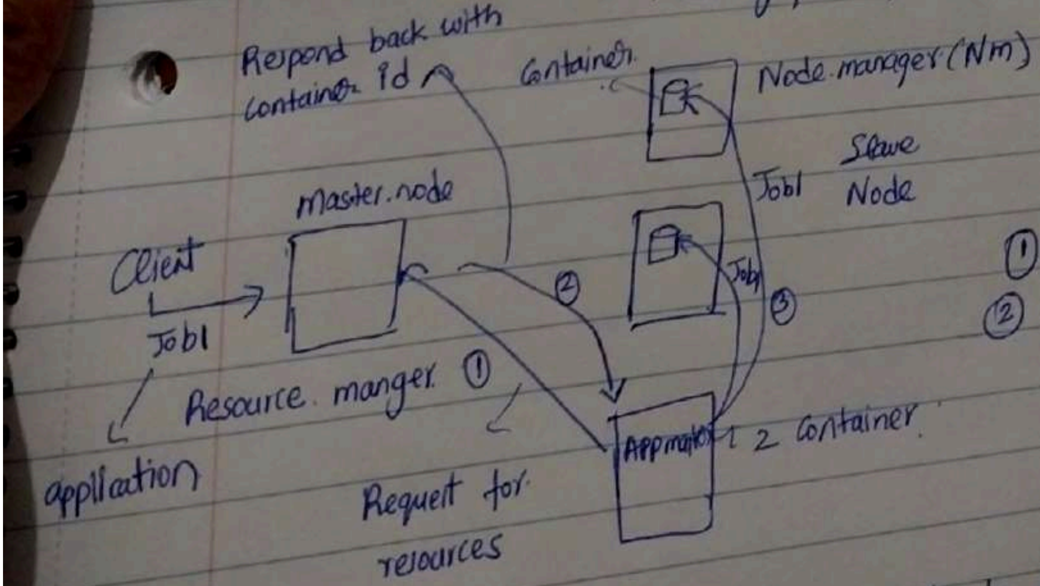
YARN → Yet Another Resource Negotion

→ container Concepts

↳ memory /duck /cpu. cycles

Job:app.matter

1:1 sure.



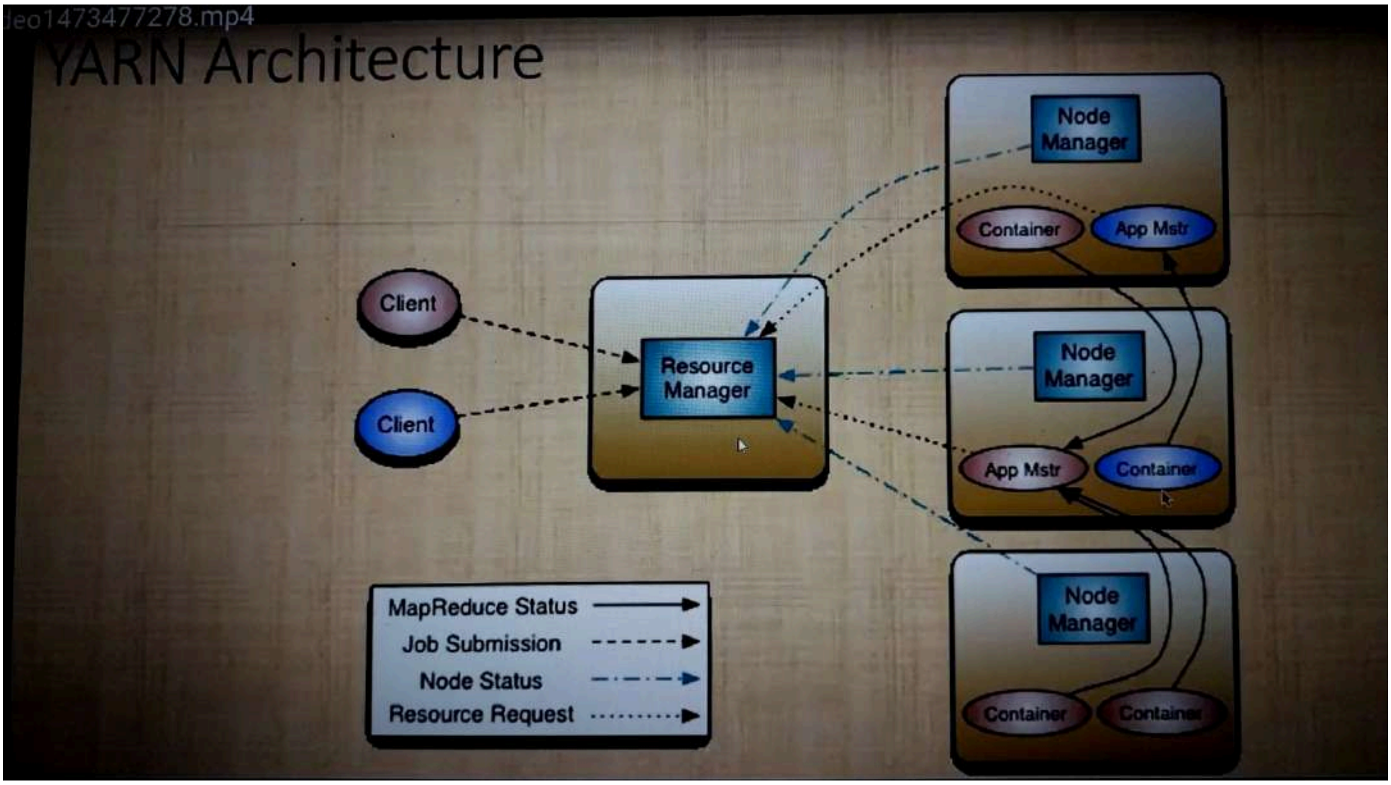
Resoure manager.

① It createl a container

② It create app marter

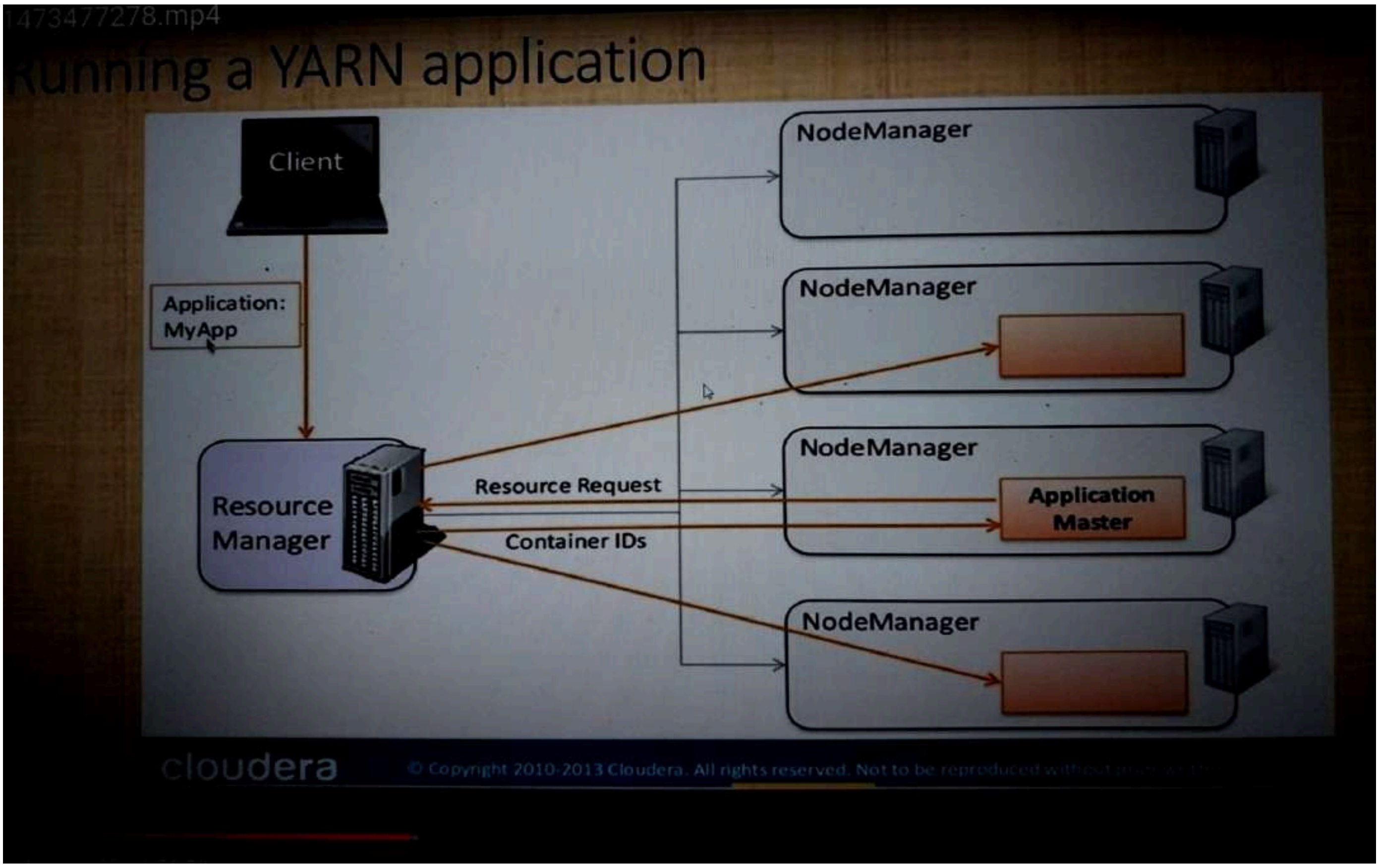
The node manager will give heart beat to the Resource manger.

health of each Node is. updated to node manger to. resoure ma



deo1473477278.mp4

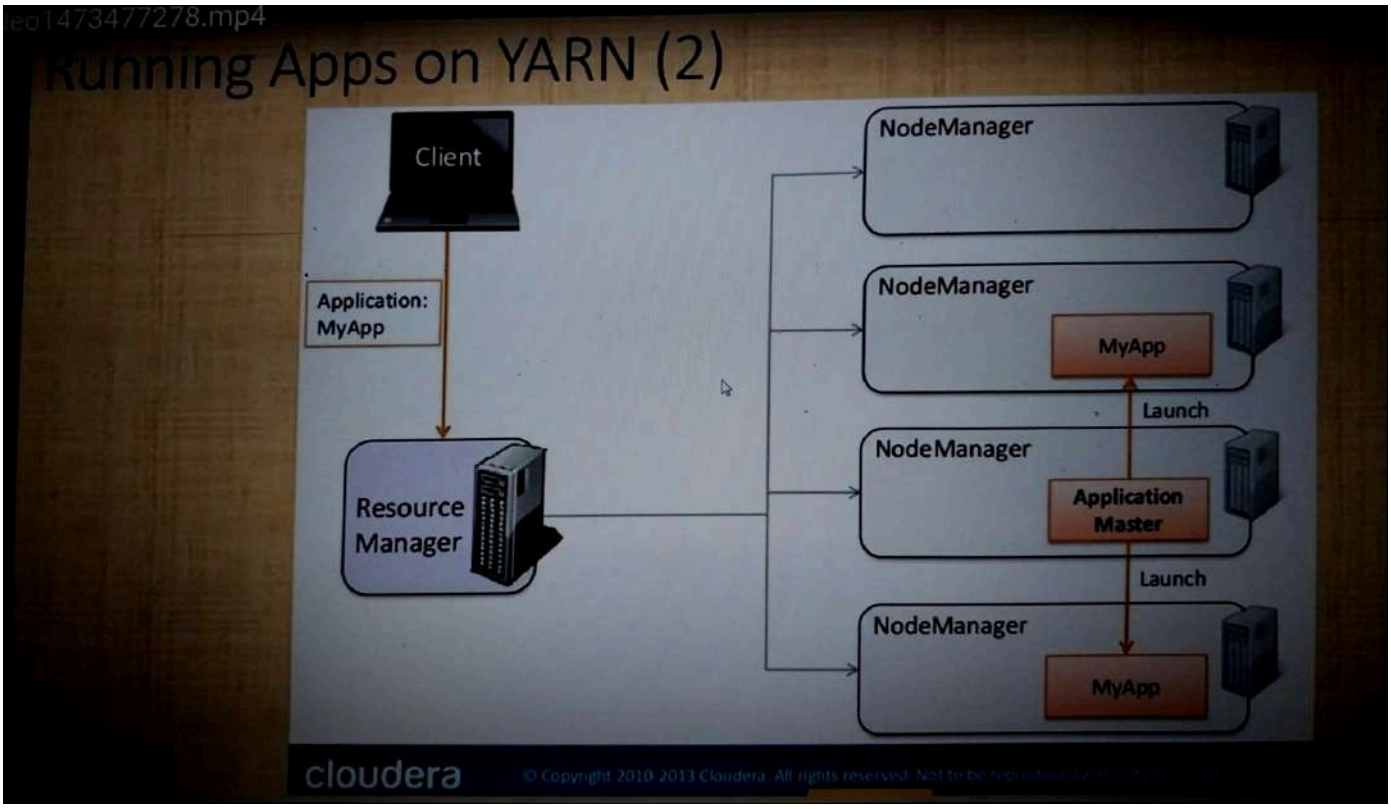
YARN Architecture



1473477278.mp4

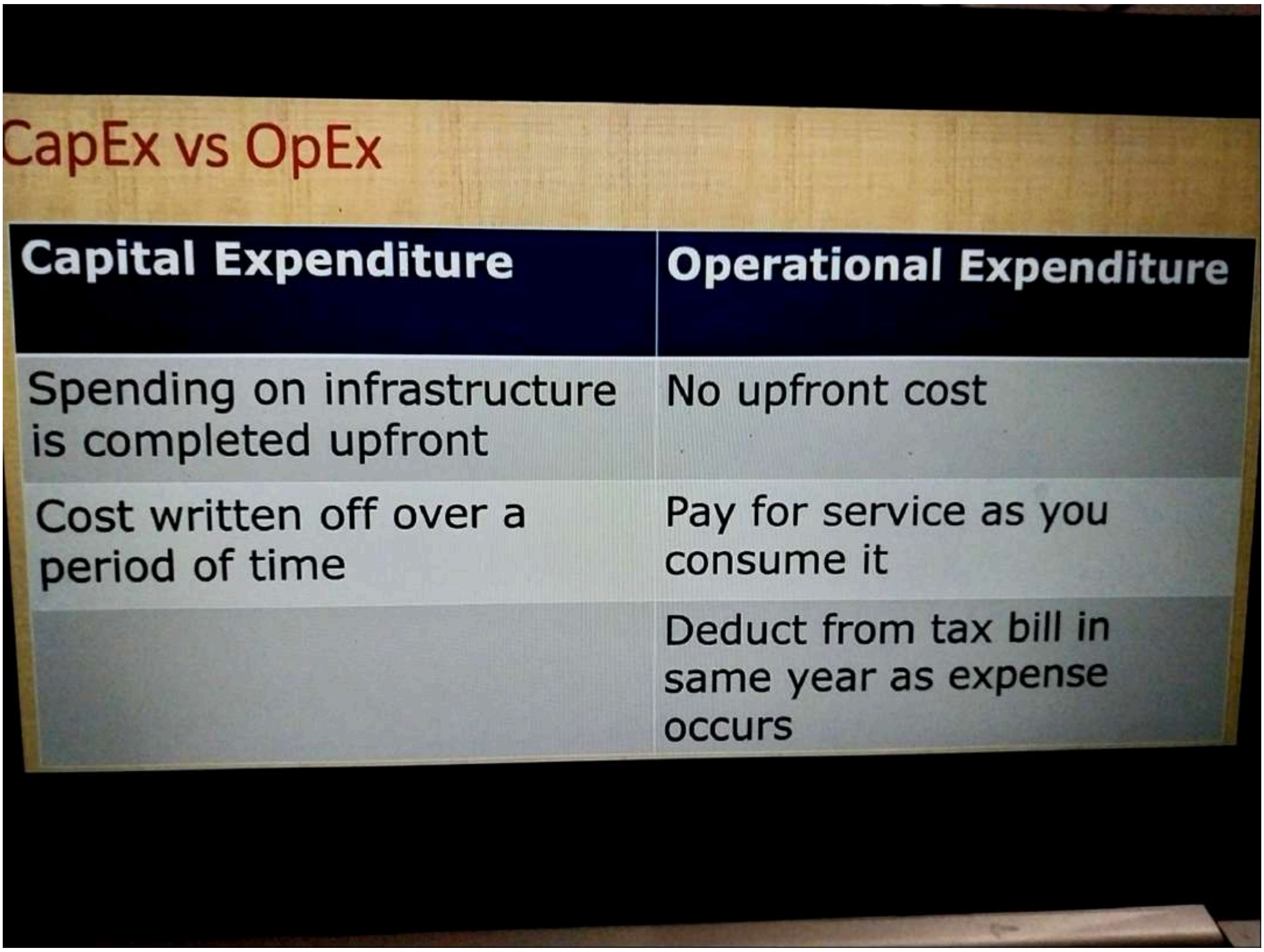
Running a YARN application

cloudera



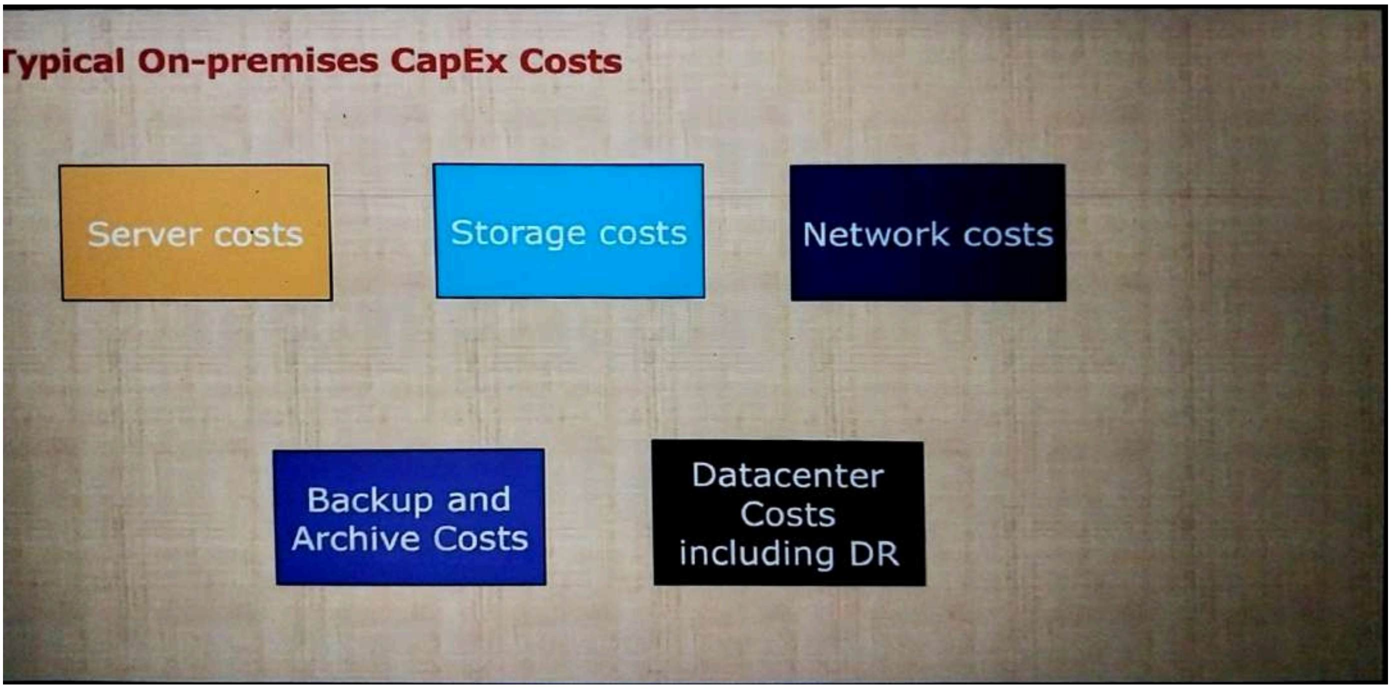
Running Apps on YARN (2)

cloudera



CapEx vs OpEx

Capital Expenditure	Operational Expenditure
Spending on infrastructure is completed upfront	No upfront cost
Cost written off over a period of time	Pay for service as you consume it
	Deduct from tax bill in same year as expense occurs



Typical On-premises CapEx Costs

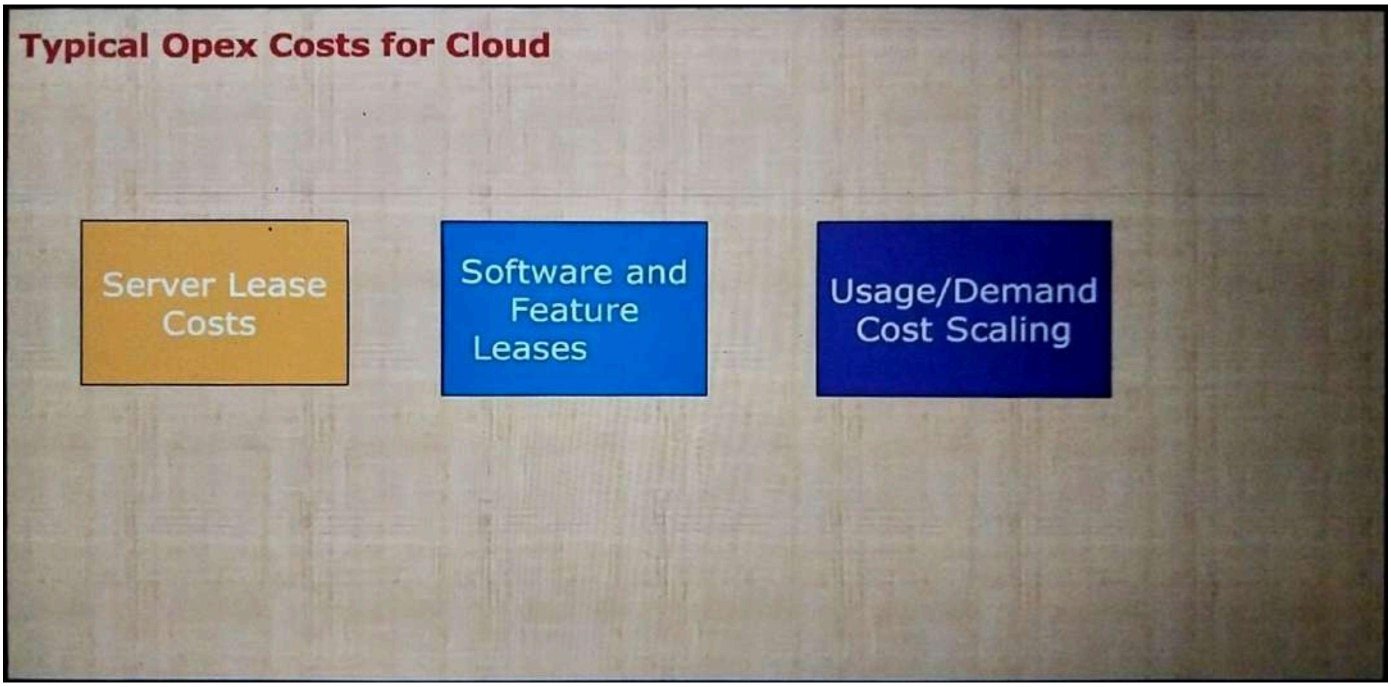
Server costs

Storage costs

Network costs

Backup and Archive Costs

Datacenter Costs including DR

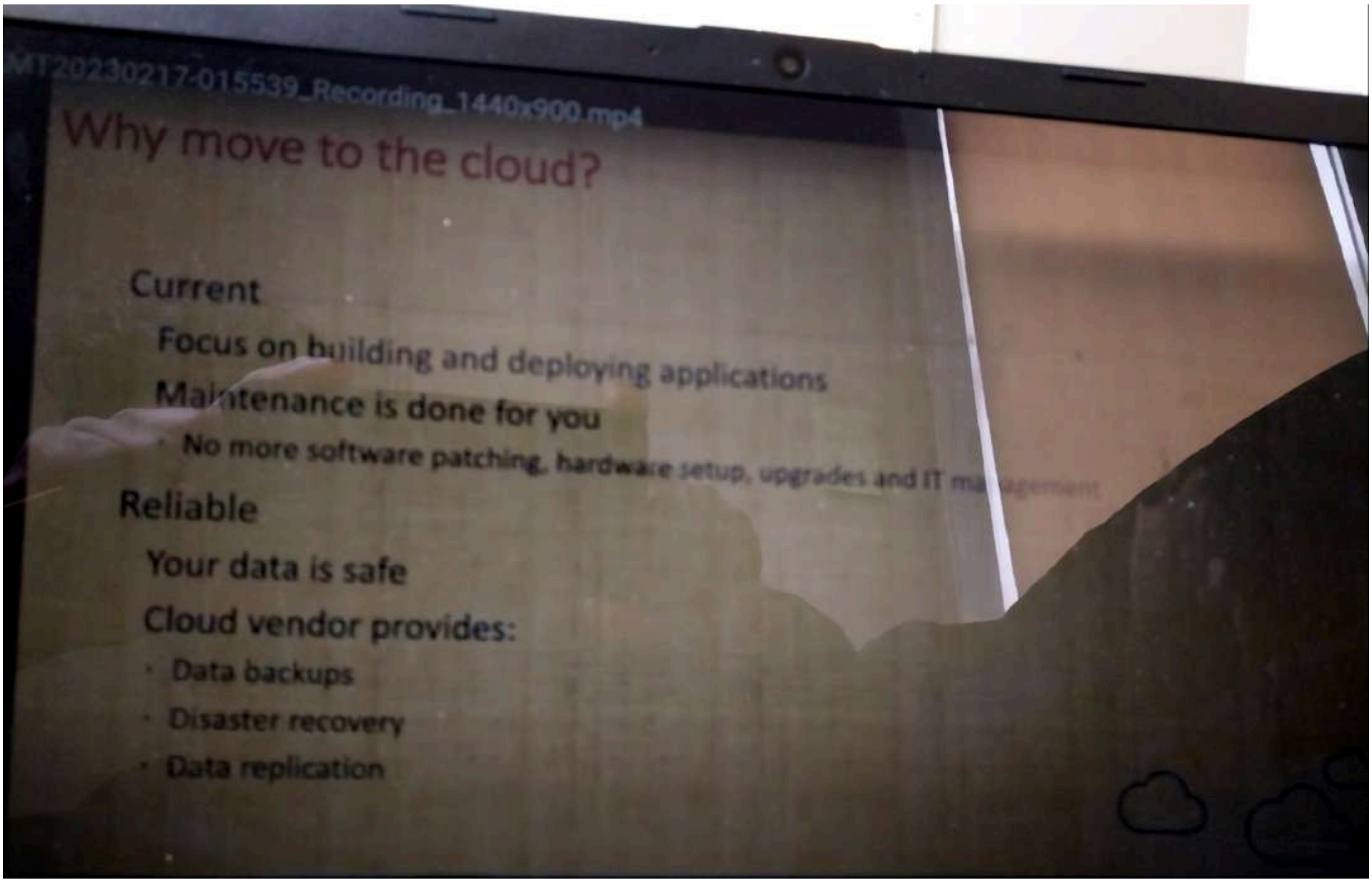


Typical Opex Costs for Cloud

Server Lease Costs

Software and Feature Leases

Usage/Demand Cost Scaling



MT20230217-015539_Recording_1440x900.mp4

Why move to the cloud?

- Current
  - Focus on building and deploying applications
  - Maintenance is done for you
    - No more software patching, hardware setup, upgrades and IT management
- Reliable
  - Your data is safe
  - Cloud vendor provides:
    - Data backups
    - Disaster recovery
    - Data replication

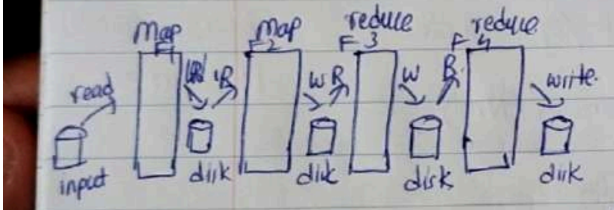
Spark → Parallel processing platforms

(In-memory computation)
↑
Spark

- → multiple Fuctions
- → both batch & real-time data
- → Memory → process the data

M.R (map Reduce)

- → 2 Functions map & Reduce
- → batch processing
- → Disk to process the data



- → multiple Read/write operation
- → Time - processing is more/high

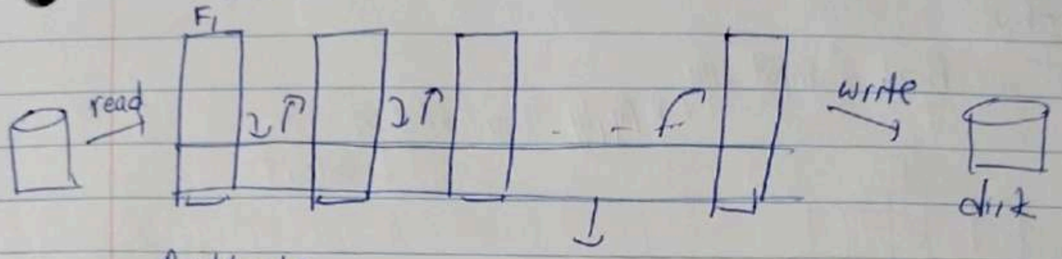
↓

On Disk takes more time

RAM → Memory → we can reduce time

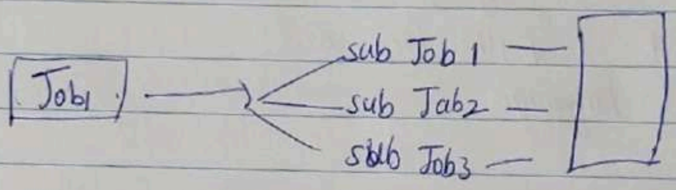
ROM → DISK → Time is huge

Spark (for processing)



Read/write happens in the RAM memory

Inmemory /distribted memory



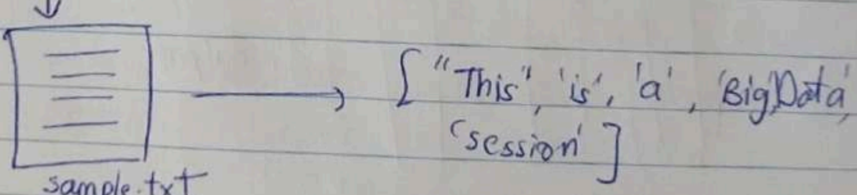
Spark

2 operation

① . Trasformation — Multiple Fuction — Converting one form of data into another form

② . Action → multiple. Fuction

show the result

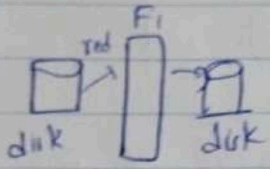


sample.txt

This is a BigData session.

Python

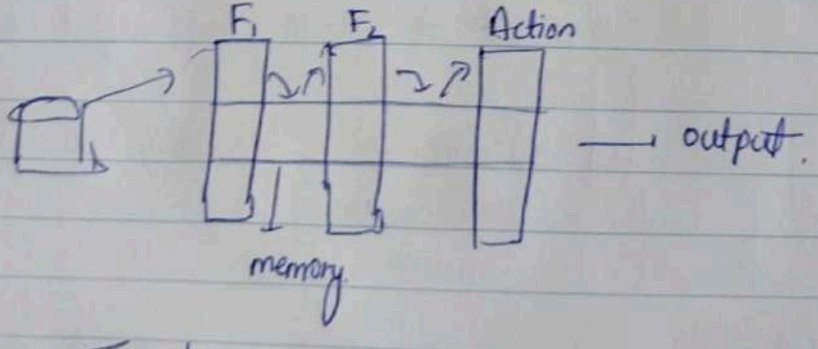
Read a input file
↳ Apply Transformation
↳ output.



Spark + python — writes — Scala Language
↳ pyspark ⇒ python language
↳ language

PySpark

lazy Evaluation { Transformation - unique feature
: will not consume any cpu cycle / ~~execute~~ not execute until. we call action



Trans
- map()
- filter()
- Reduce ByKey()

Action
- collect()
- show()
- take()

} shows output

spark → API's

spark → 3.X.V

1.4 ≤ 1.4 >

↓ ↓

RDD API DF ← RDD

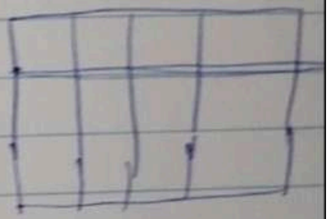
& SQL

output

RDD → unstructured Data — [.......]

↓

DF → structured way of processing Data. —



& SQL

└ Hand-on — cloud

	Master.	Slave
HDFS -	Name Node	Data Node
MR -	Job Traker.	Task Traker
YARN -	Resource manager	Node manger.
SPARK -	Drivers	Executors.

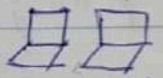
we care - cloud ? why I should move to cloud

start up – in developing a mobile App – Food ordering . App

End – End product Development

client → Development → Deliver (customer)
launch it

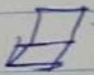
Infrastructure:- (server, computers) → server
OS
storage
software stack < tool
funcality
↓
Avg Computing power.



P Laptops.

10-developer's

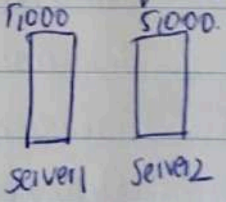
Benglure – 10,000 customer



→ Itrastructure for production (launch it) is different compare to development.

Day1 Target customer 10,000

Infra required



IT. Team 2 members

No. of customer - 2k - 5k

Day5 – 9k customer → works fine

Day6 — 15k Customer → Accell the App

? - server down

- customer - lose & bad Feedback

Add new servers → 2 server - Budget
- configure

Day10 20,000 4 server's - 3 IT Team

No issuers ✓

Day20 → 30,000 6 server - 4 IT Team

↓ 10day - No. issues ✓

Day 30.

Day32 -	No. of user's	- 5K	?	Cost is fixed ?
33	"	- 6K		
"	"	- 10K		

observation

- → maintance issues - costly, Time
- → server cost
- → IT Team.
- → cost is fixed

Focus
?
Developing New product with less budget.

3rd party - comes to you
(They take of? all
Infastructure (serve's)
Accepted

after — cloud.
↓
pay for what you use.

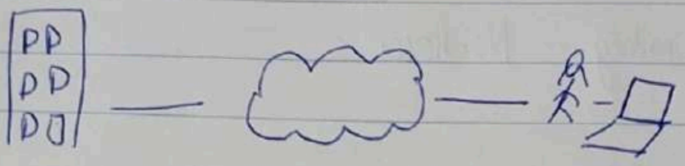
Day - 4hrs - pay for 4hrs
1hr → " "

Where can infrastructure be hosted?

- For an enteprise, it is on a Data Centers

What is cloud Computing

- Delivery of computing services over internet



-) | aaccessing someone else ~~over~~ computer over the internet for my computer and pay only for. cloud services you use.

What is cloud Computing?

* Renting resources vs Purchasing the hardware

# Pay for what you use

* Run your applications in Someone else's datacenter

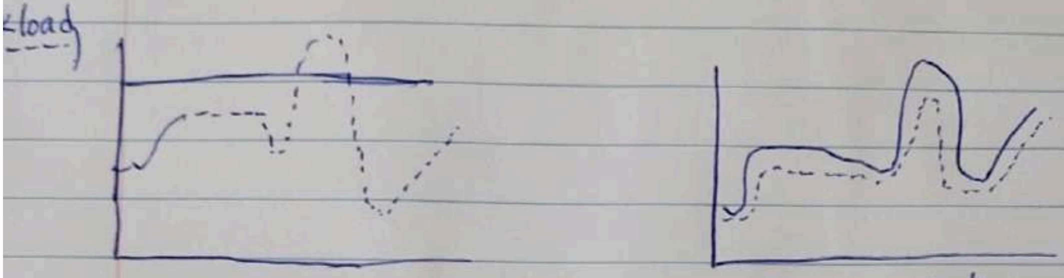
* Cloud provider is responsible for the physical hardware and facilities necessary to execute your work

* Cloud provider is responsible for keeping the servies to uptod

Elaiticity)

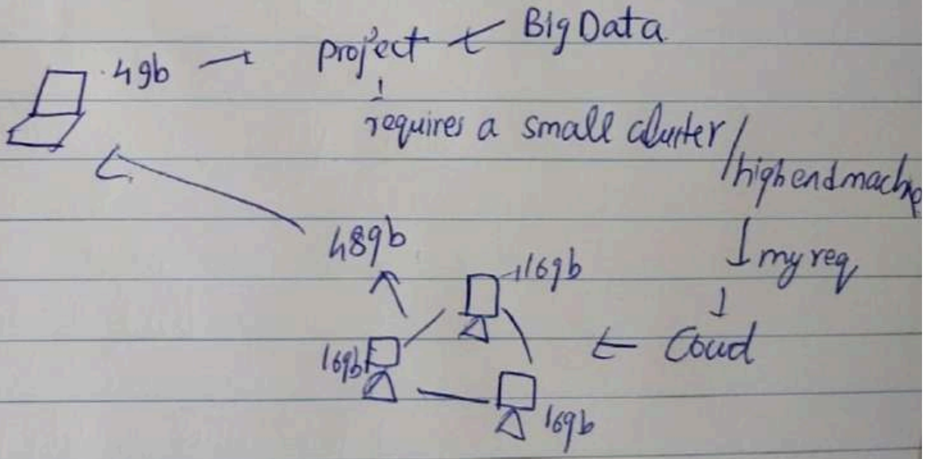
- As your workload changes, resources can be changed to compensate (up or down)

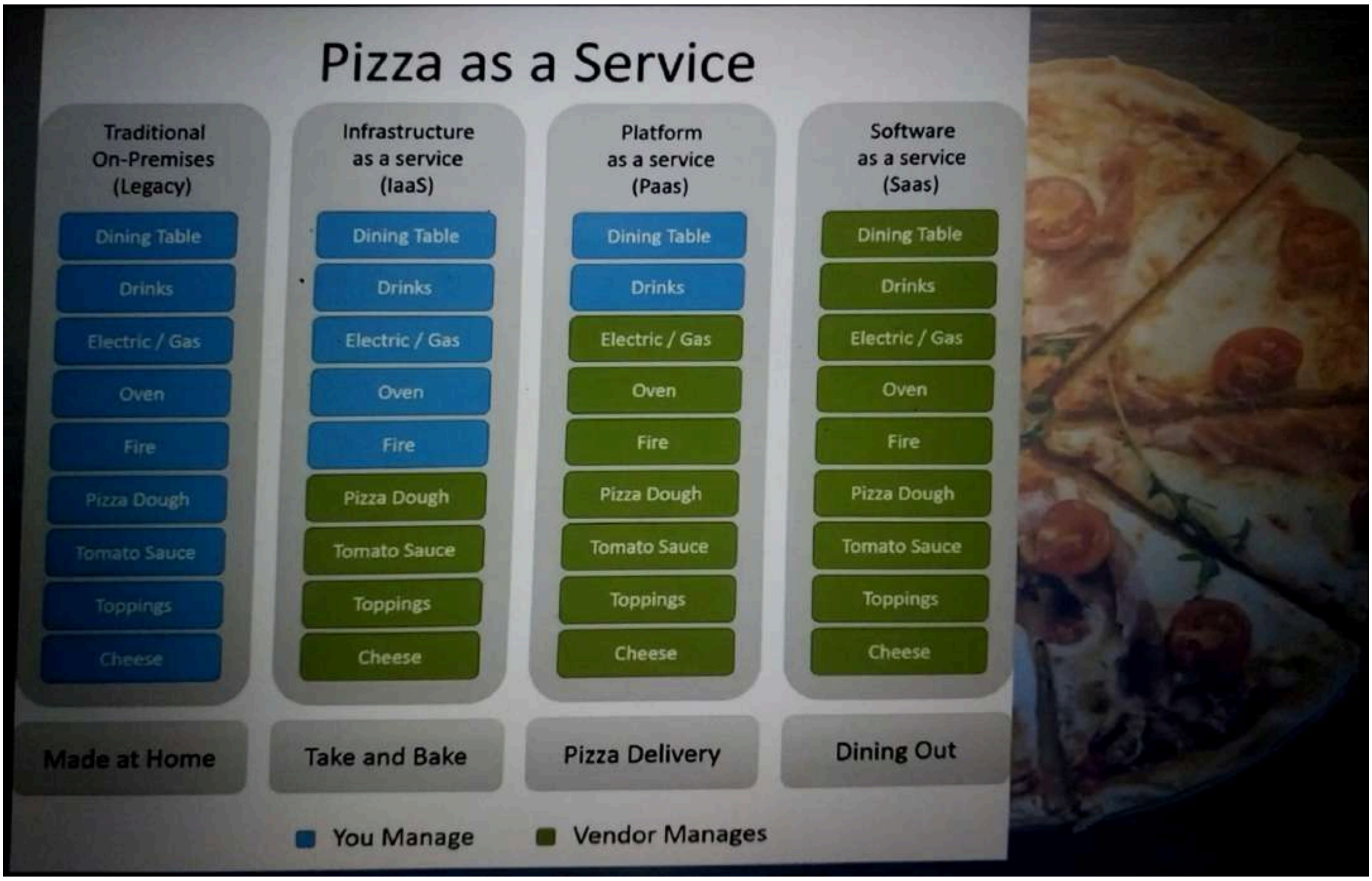
example : Seasonal Demand for retail website Black Fiday



Static

Elastic Scaling



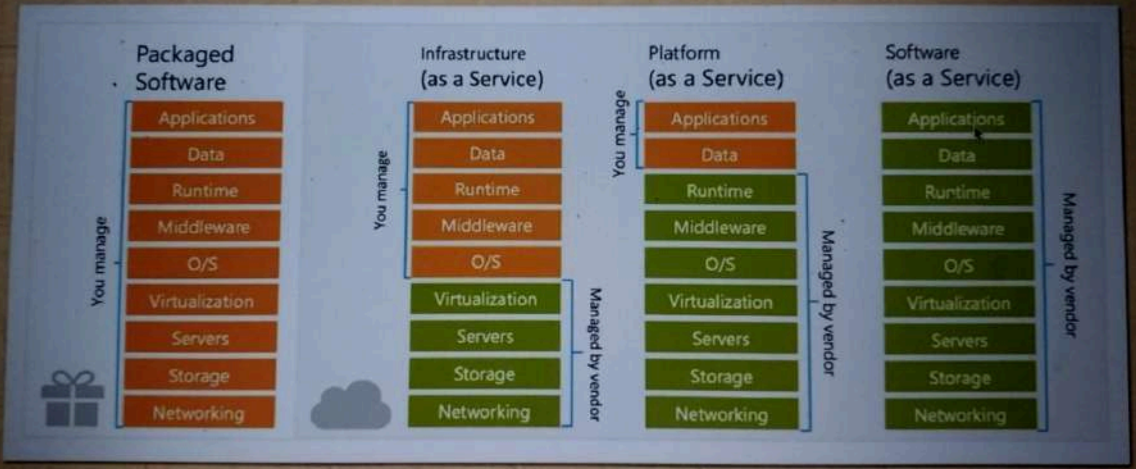


Pizza as a Service

Traditional On-Premises (Legacy)	Infrastructure as a service (IaaS)	Platform as a service (Paas)	Software as a service (Saas)
🟦 Dining Table	🟦 Dining Table	🟦 Dining Table	🟩 Dining Table
🟦 Drinks	🟦 Drinks	🟦 Drinks	🟩 Drinks
🟦 Electric / Gas	🟦 Electric / Gas	🟩 Electric / Gas	🟩 Electric / Gas
🟦 Oven	🟦 Oven	🟩 Oven	🟩 Oven
🟦 Fire	🟦 Fire	🟩 Fire	🟩 Fire
🟦 Pizza Dough	🟩 Pizza Dough	🟩 Pizza Dough	🟩 Pizza Dough
🟦 Tomato Sauce	🟩 Tomato Sauce	🟩 Tomato Sauce	🟩 Tomato Sauce
🟦 Toppings	🟩 Toppings	🟩 Toppings	🟩 Toppings
🟦 Cheese	🟩 Cheese	🟩 Cheese	🟩 Cheese
Made at Home	Take and Bake	Pizza Delivery	Dining Out

🟦 You Manage 🟩 Vendor Manages

XaaS (What can be rented?)



Packaged Software

You manage: Applications, Data, Runtime, Middleware, O/S, Virtualization, Servers, Storage, Networking

Infrastructure (as a Service)

You manage: Applications, Data, Runtime, Middleware, O/S

Managed by vendor: Virtualization, Servers, Storage, Networking

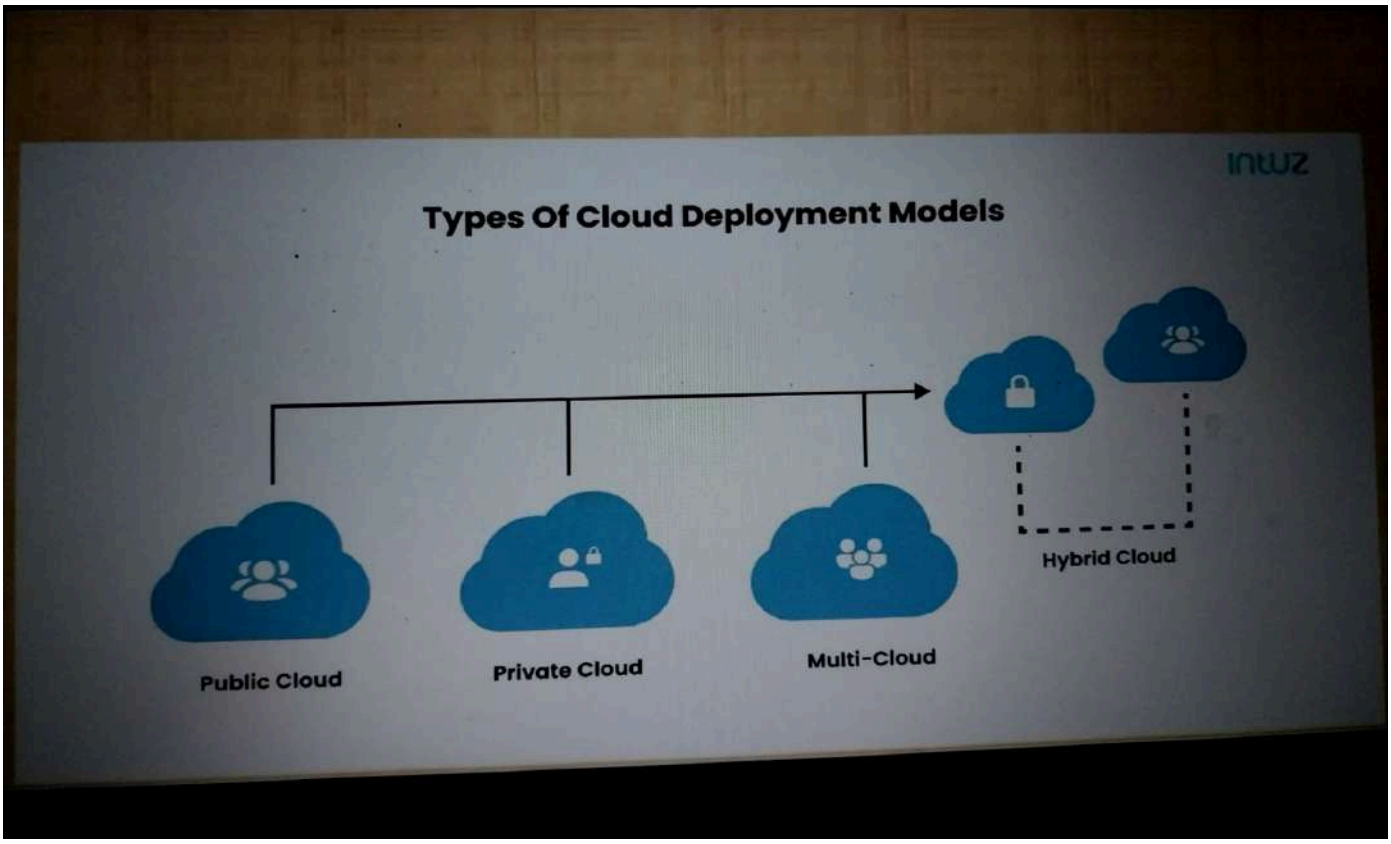
Platform (as a Service)

You manage: Applications, Data

Managed by vendor: Runtime, Middleware, O/S, Virtualization, Servers, Storage, Networking

Software (as a Service)

Managed by vendor: Applications, Data, Runtime, Middleware, O/S, Virtualization, Servers, Storage, Networking



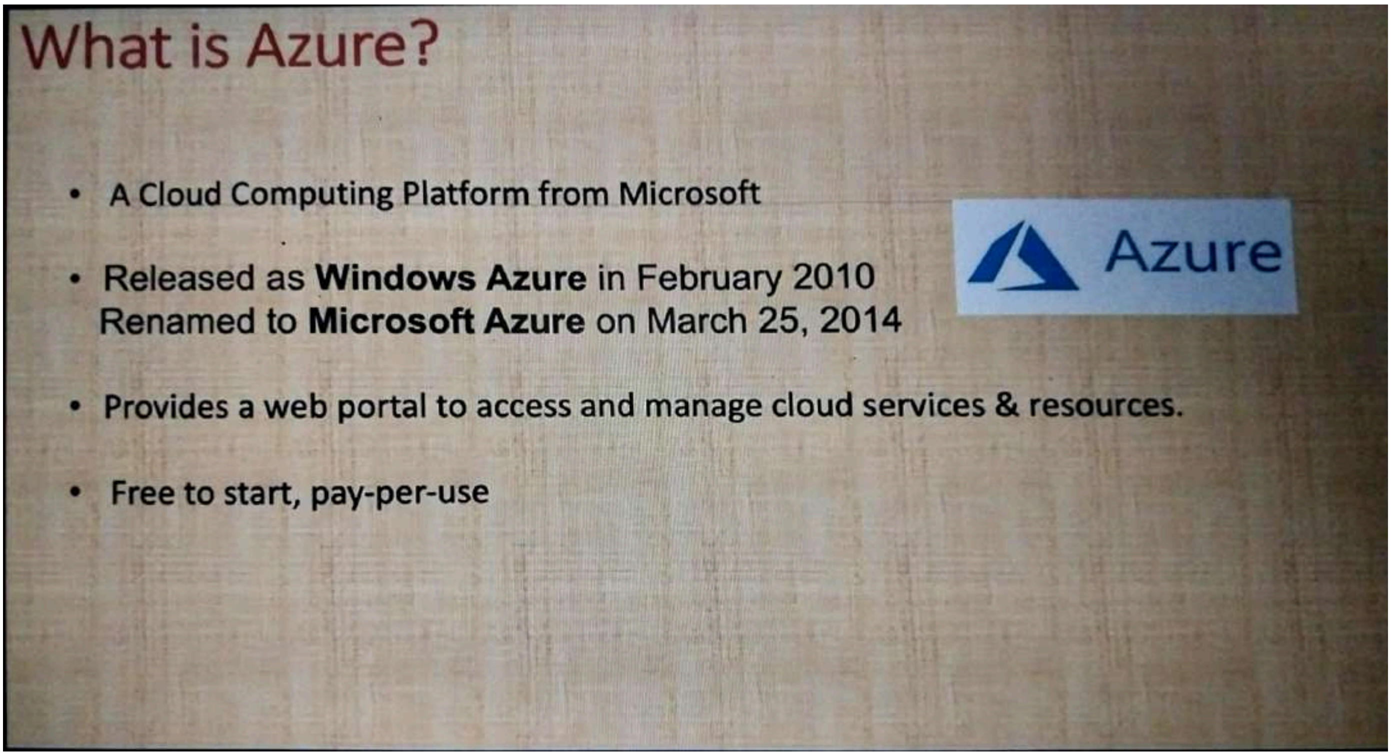
Types Of Cloud Deployment Models

Hybrid Cloud

Public Cloud

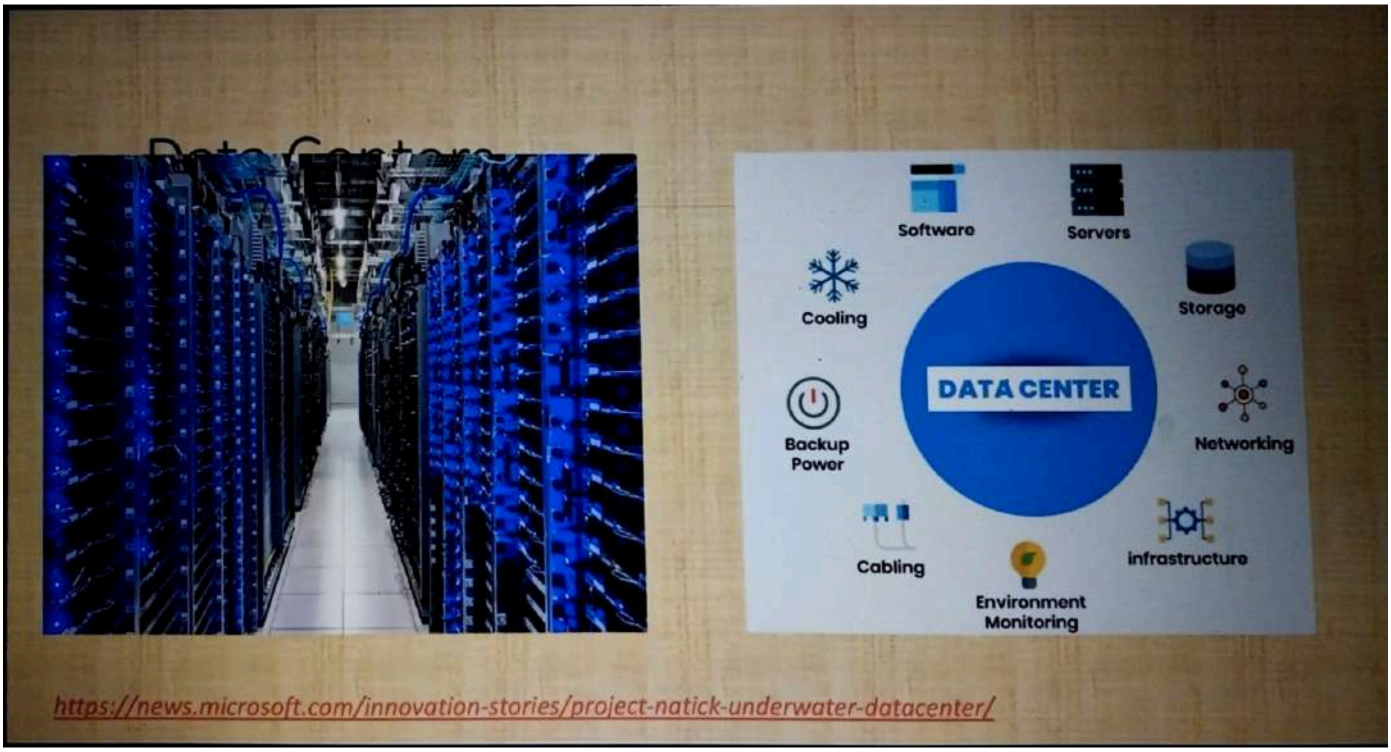
Private Cloud

Multi-Cloud



What is Azure?

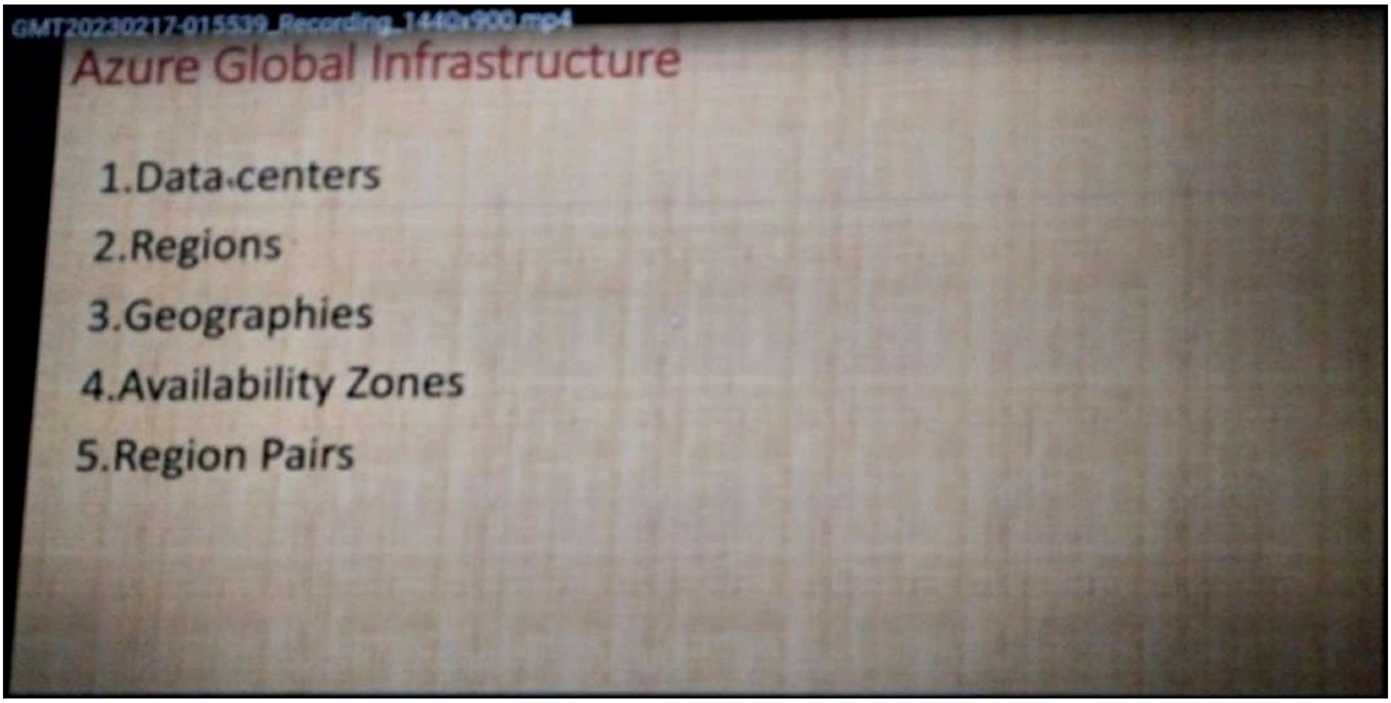
- A Cloud Computing Platform from Microsoft
- Released as **Windows Azure** in February 2010
  Renamed to **Microsoft Azure** on March 25, 2014
- Provides a web portal to access and manage cloud services & resources.
- Free to start, pay-per-use



Data Centers

Software
Servers
Cooling
Storage
DATA CENTER
Backup Power
Networking
Cabling
Infrastructure
Environment Monitoring

https://news.microsoft.com/innovation-stories/project-natick-underwater-datacenter/



GMT20230217-015539_Recording_1440x900.mp4

Azure Global Infrastructure

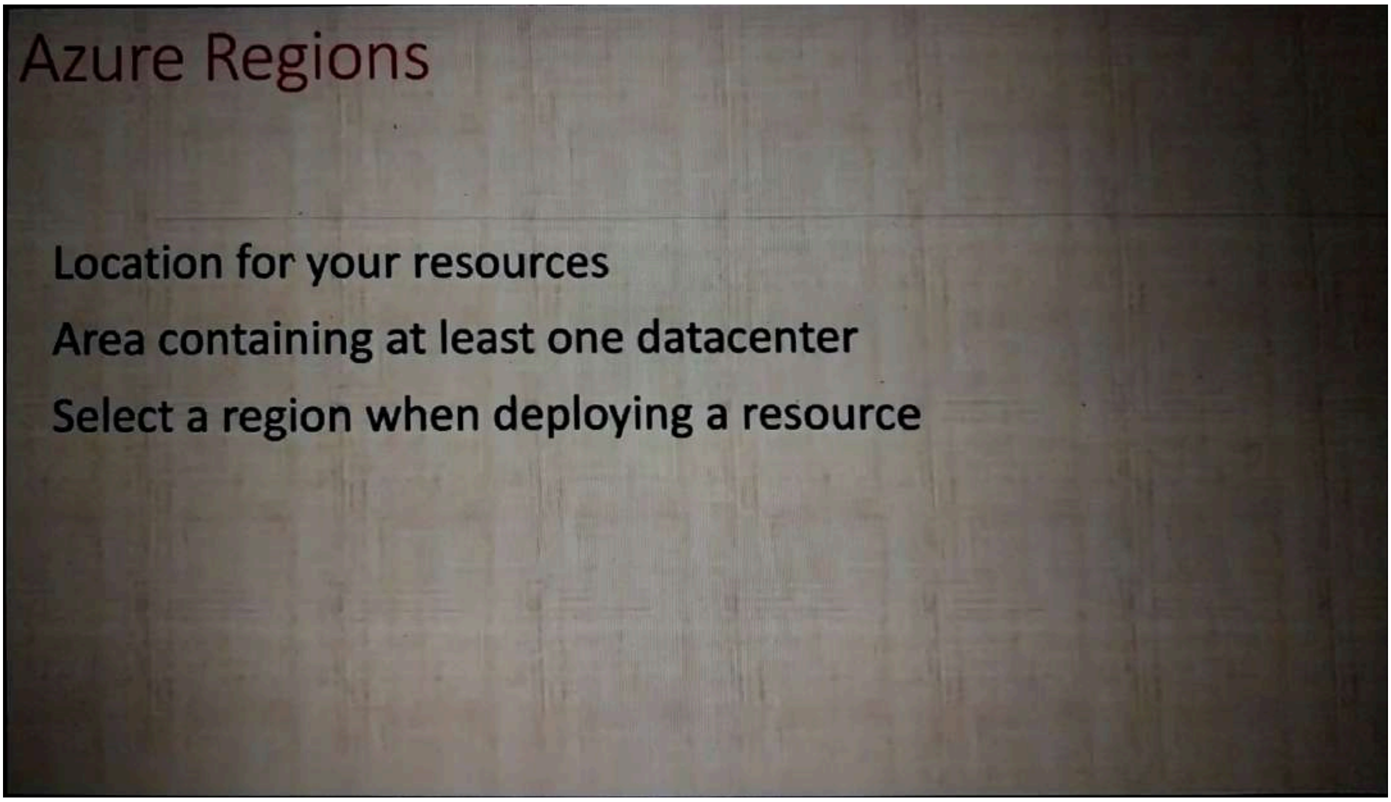
1.Data centers

2.Regions

3.Geographies

4.Availability Zones

5.Region Pairs

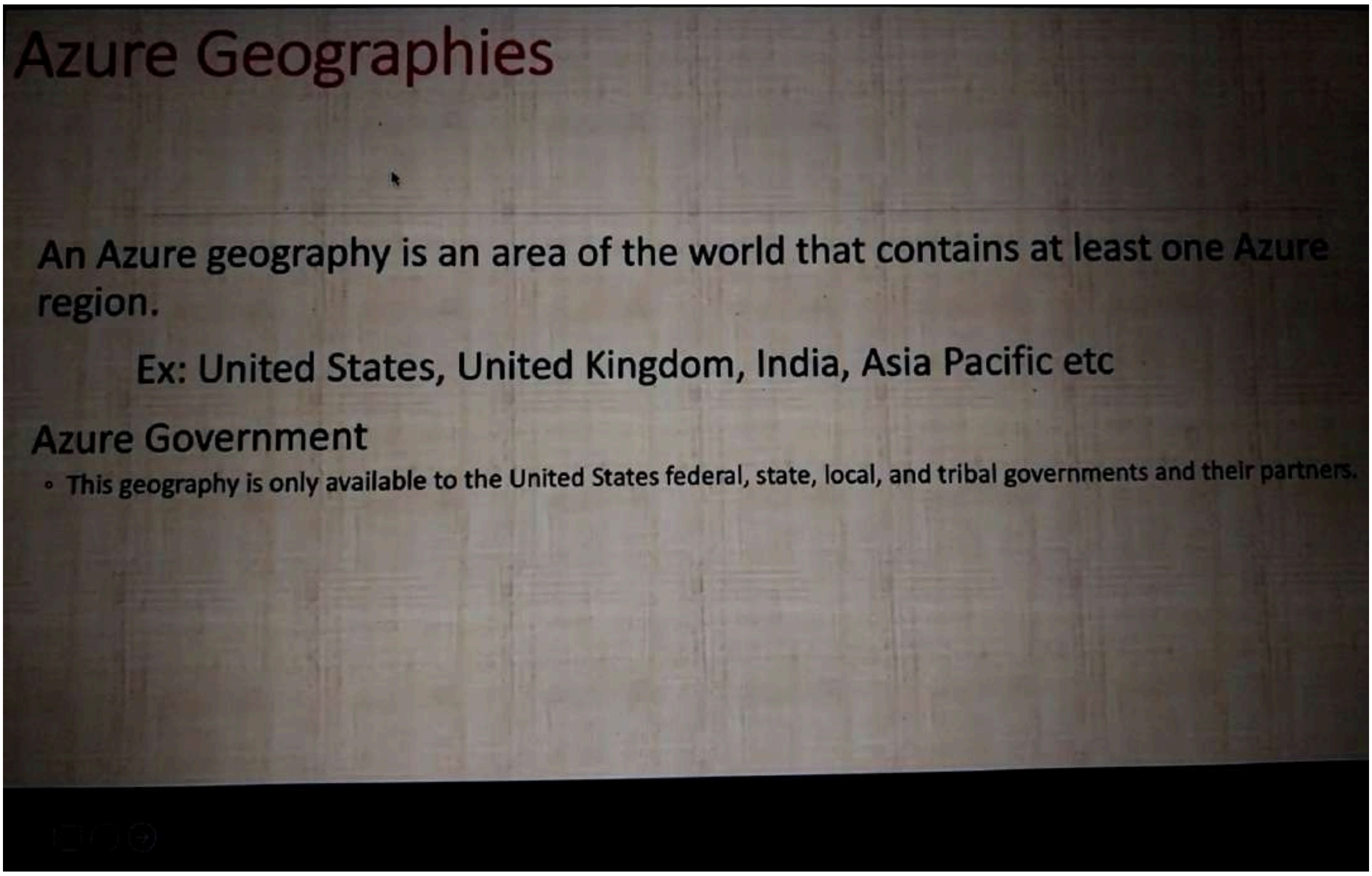


Azure Regions

Location for your resources

Area containing at least one datacenter

Select a region when deploying a resource



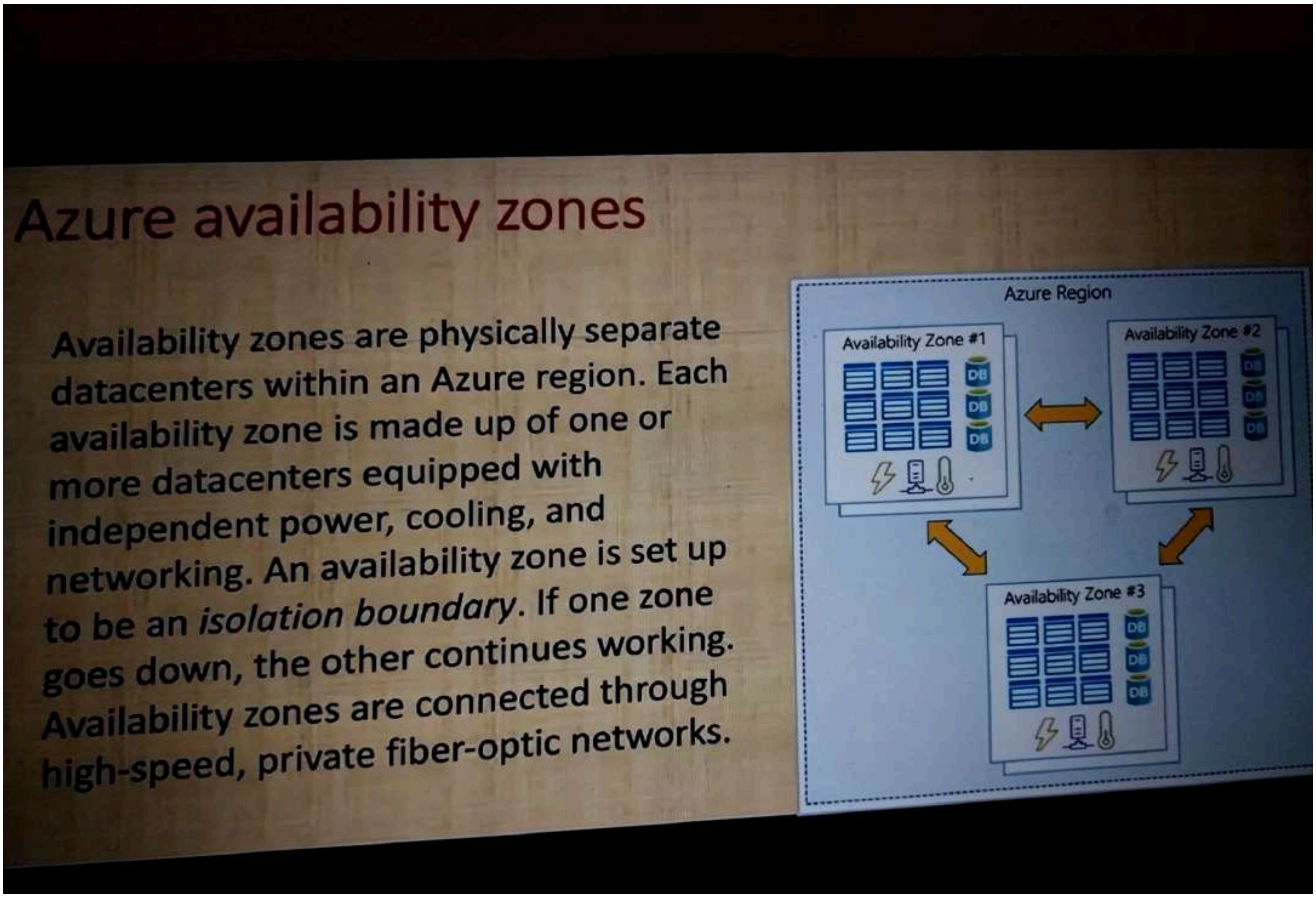
Azure Geographies

An Azure geography is an area of the world that contains at least one Azure region.

Ex: United States, United Kingdom, India, Asia Pacific etc

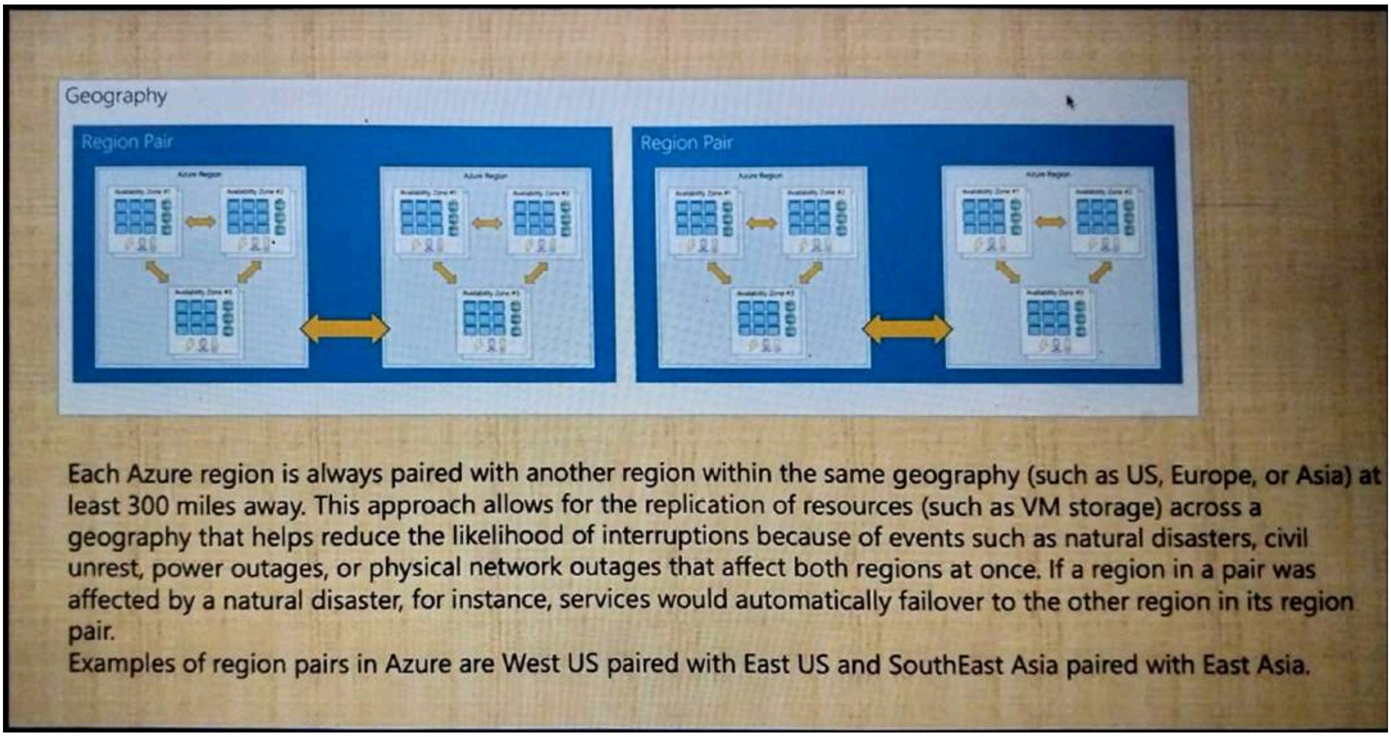
Azure Government

- This geography is only available to the United States federal, state, local, and tribal governments and their partners.



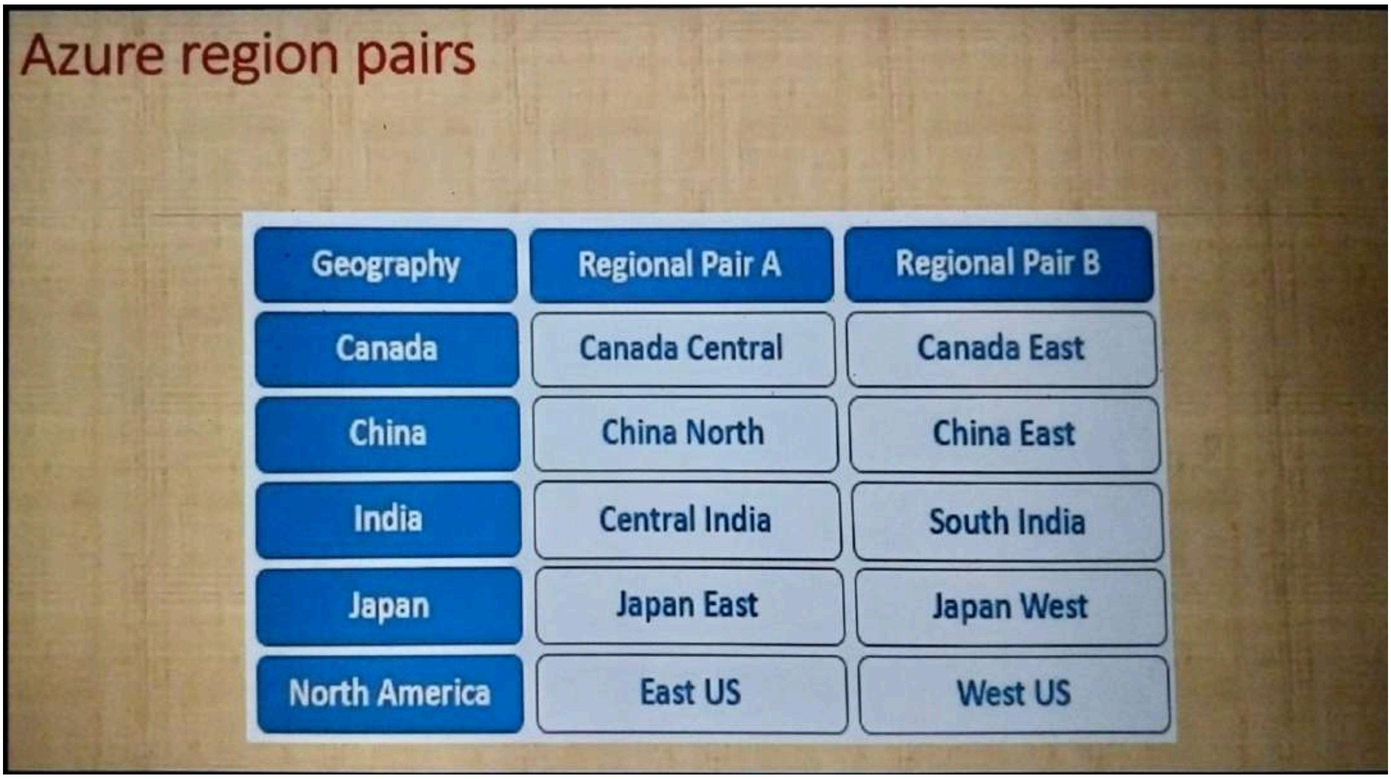
Azure availability zones

Availability zones are physically separate datacenters within an Azure region. Each availability zone is made up of one or more datacenters equipped with independent power, cooling, and networking. An availability zone is set up to be an *isolation boundary*. If one zone goes down, the other continues working. Availability zones are connected through high-speed, private fiber-optic networks.



Each Azure region is always paired with another region within the same geography (such as US, Europe, or Asia) at least 300 miles away. This approach allows for the replication of resources (such as VM storage) across a geography that helps reduce the likelihood of interruptions because of events such as natural disasters, civil unrest, power outages, or physical network outages that affect both regions at once. If a region in a pair was affected by a natural disaster, for instance, services would automatically failover to the other region in its region pair.

Examples of region pairs in Azure are West US paired with East US and SouthEast Asia paired with East Asia.

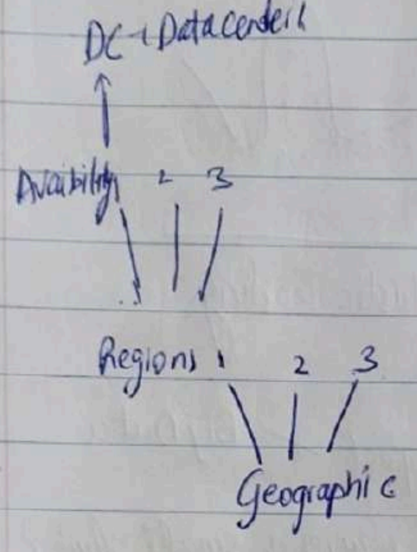


Azure region pairs

Geography	Regional Pair A	Regional Pair B
Canada	Canada Central	Canada East
China	China North	China East
India	Central India	South India
Japan	Japan East	Japan West
North America	East US	West US

Core claud services

Compute Storage Networking
App Services Analytics



India → Geografic
↑
South Iindia → Region
↑
DataCentor → DataCentor

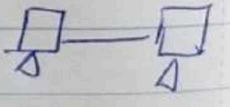
IAAS

→ Compute Engine
→ Storage services
→ Network services

Compute engin
↓
→ VM
→ Containers
→ Kubernetes

Storage servic
↓
Structure data → Azure SQL
Semi-Structure data → Cosmas DB
un-structured data → ADL & Blob

network servies

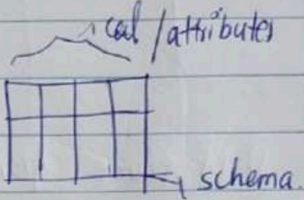


→ NIC
→ IP. address
→ subnets

Resource Group

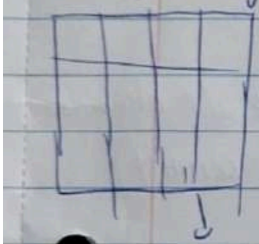
Service — Azure

R.G ......
- VM
- storage



* Storage services

store any data



- Structured Data - Azure
- Unstructured Data → Blob - ADLS
  - No filesystem
  - Flat file
- Semi-structured Data → ~~or~~ Cosmos DB

Relational Data — Schema on write

→ RDBMS

→ 1) Create the schema before / load / store data

non-relational data — Schema on read

→ NoSQL - Not only SQL

can store SQL data and also

→ flexibility in the schema

Structured data → semi-structed data

①

create table students (Sid number, name varchar(20) ....); Mango.DB

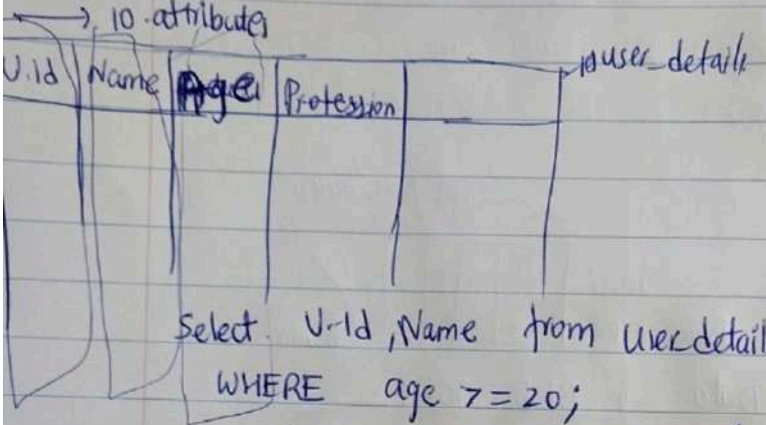
→ NoSQL → (JSON Java script object Notation)

id	Name	H_1	H_2	H_3	....	H_{10}
1	Anu	-	-	7	?	
2	-	-	-	?	?	-
3	-	-	-		?	?

→ only have H_3. all are empty so waste of storage

Limitation are in the back

1) Every row/record is a table have same no.of attribute

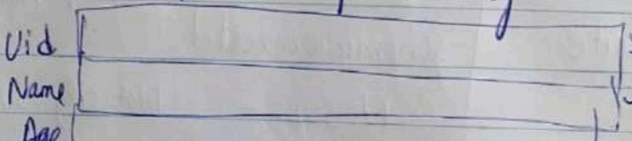


Select U-Id, Name from userdetails
WHERE age >= 20;

it process in the Traditinal way (because it check all attributes it was not necessary).

② SQL is not an efficient for reading

NOSQLs
Mango-DB
- JSON
Java script object Notation



- it change column to rows
- Columnar Data store

```
{
  U-id:
  Name:
  hobby: ["", "", ""]
}
```

```
{
  U-id:
  Nam:
  hobby: [ ....... ];
}
```

- we can take how many you want

BigData
- processing . ADLS - Big Data processing
required to have
file system

Flat Files
=

Blob

* we need not to enable

Storage Account → ☒ to enable

to hir namespace

↑

container – space

↑

store our data

□

BigData processing
=

ADLS

→ Storage

☑ to enable the hir.

namespace

↑

container

↑

can create a diretory fil

↑

store our data

□

Access

Hot – Use more

cool – Not use more

Archive –

Strorage servcies

→ DB
→ SQLserver

Structured — Azure SQL
semi structured — Cosmos DB → mango.DB
un-structured → blob & ADLS
↑
JSON doc

Azure Databricks

Bigdata → Hands on

Spark → run all spark job – Introduce the service Name. Azure Databricks

— scala
→ programing Language – pyspark

→ platform – spark – 3. X-version

→ service – data. bricks

Azure Databricks

1) Create a Databricks workspace
2) Create a cluster – single Node Cluster
3) create a Notebook – write spyspark Code
4) DBFS → Data bricks file system
L. create DBFS & storege datasets

spark

Version ≤1.4 → Spark RDD

Version >1.4 → Spark DF & SQL — object spark session → spark

3.X

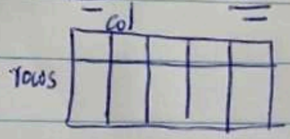
Spark DF & SQL ↑ spark RDD — spark context → SC

Processing the data in the structured & semi sturture

spark RDD → unstructured list of list [[ ]]

Pyspark → DF

SQL ← pyspark - programing + SQL - query langyge



Spark

2option

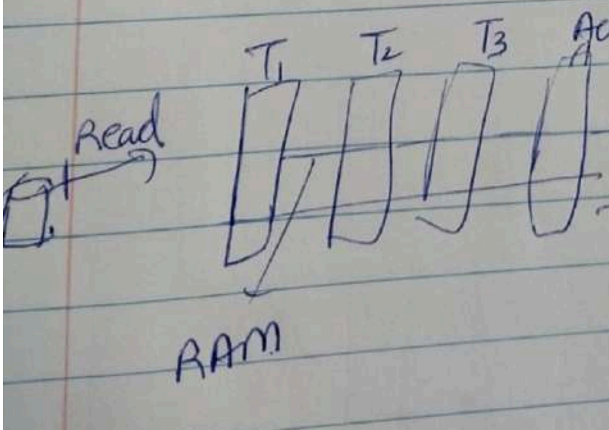
1) Tranformation → a, b, c — it will not execute

↓ Fuction

['a','b','c']

2) Action

not easy memory



→ it Action Triggerd then only all transform
will run