JOURNEY PPT

Anubhav Bagri, Batch 8, Room 103

Week 1 Foundation

18 Aug 2023 – 25 Aug 2023

- Business Analysis
- Agile
- DBMS
- Software Testing
- DevOps & Github
- Cloud

Software Development Life Cycle

SDLC

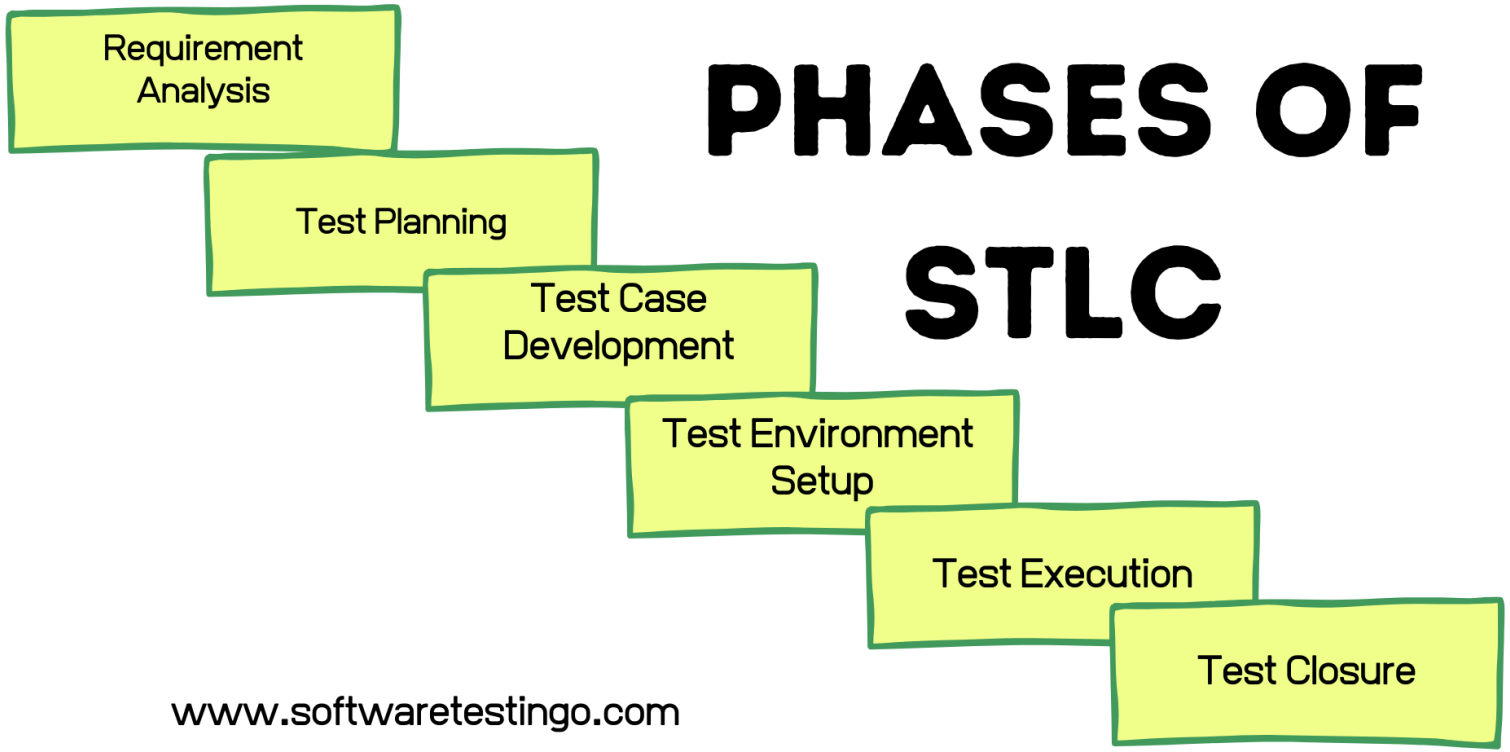
- Requirements gathering & Analysis
- Design
- Coding
- Testing
- Deployment
- Maintenance

SDLC Models

Waterfall Model	V Model	Prototype Model
Spiral Model	RAD Model	Agile

STLC

- Requirement analysis
- Test Planning
- Test case development
- Test environment Setup
- Test execution
- Test cycle closure



Waterfall Model

- Cannot go back to the previous phase i.e. Backtracking not possible.
- Client can see resulting product at the end only.
- So, client not satisfied.

Agile Methodology

- Entire process is broken down into sprints.
- Client can see product at the end of the sprint so feedback can be given continuously.
- Incremental development can be seen; very flexible & adaptable.

SCRUM OVERVIEW

Roles

- Product Owner
- Scrum Master
- Dev Team

Issues

- Epic
- User Story
- Bug
- Task

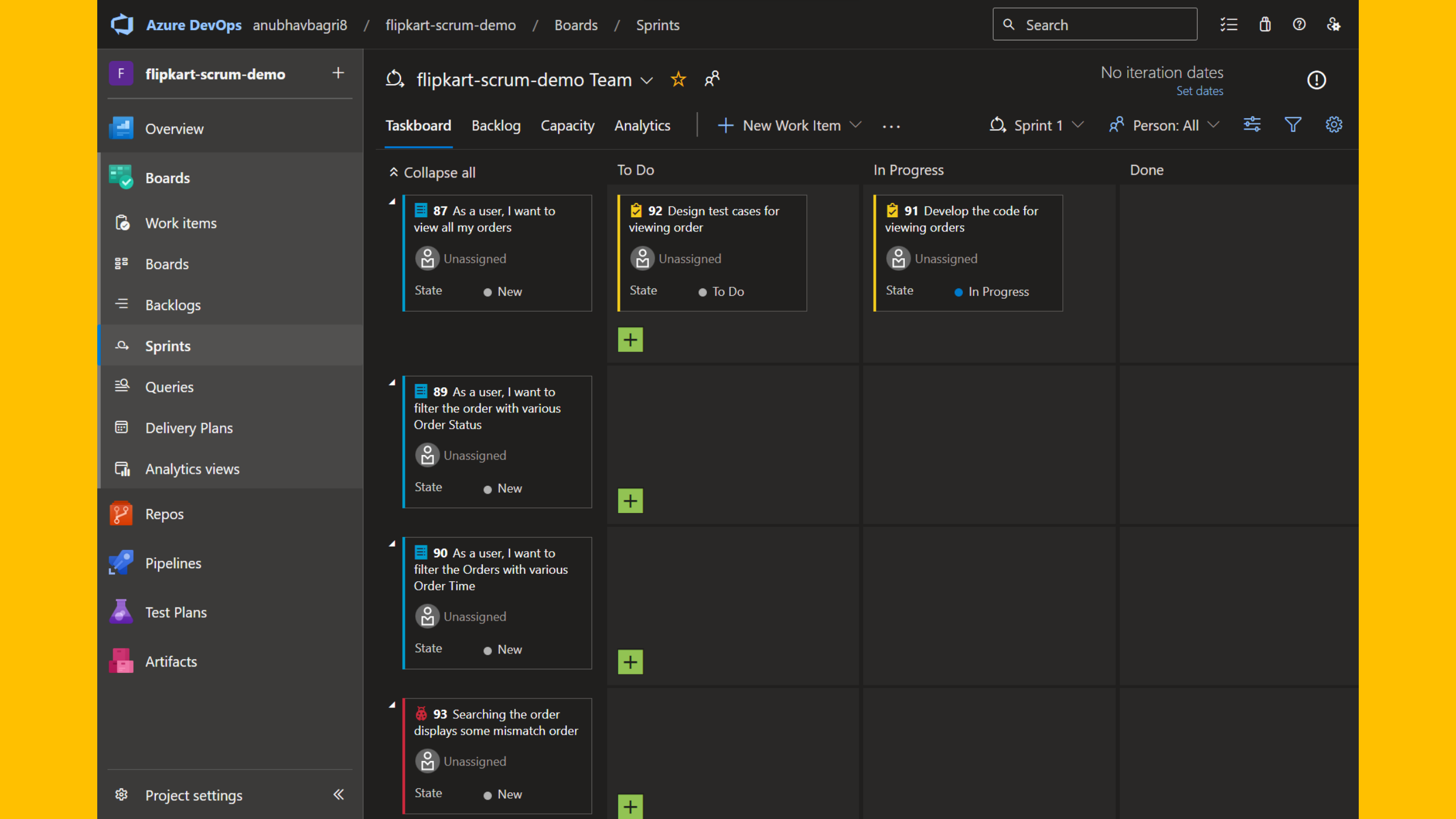
Backlogs

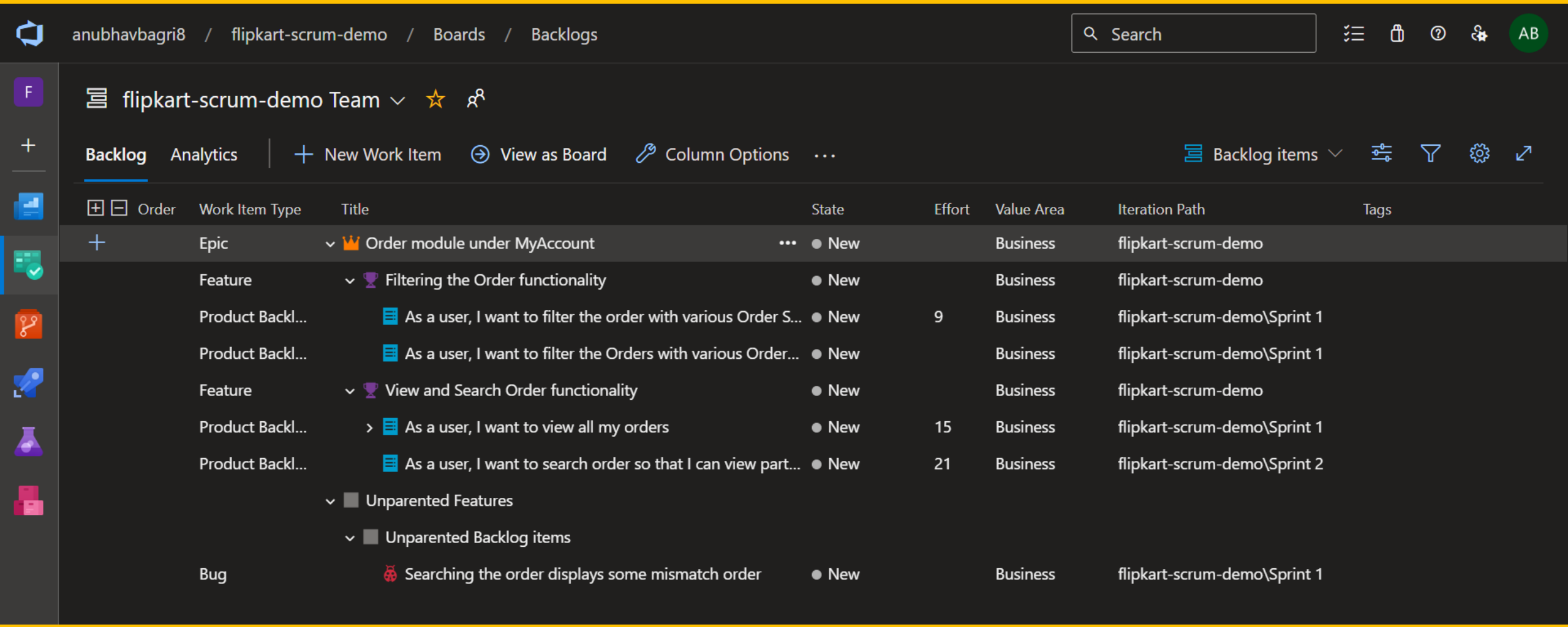
- Product Backlogs
- Sprint Backlogs

Events/Stages

- Release Planning
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

- Burn Up Chart
- Burn Down Chart


Azure DevOps anubhavbagri8 / flipkart-scrum-demo / Boards / Sprints
Search
flipkart-scrum-demo
Overview
Boards
Work items
Boards
Backlogs
Sprints
Queries
Delivery Plans
Analytics views
Repos
Pipelines
Test Plans
Artifacts
Project settings
flipkart-scrum-demo Team
No iteration dates
Set dates
Taskboard
Backlog
Capacity
Analytics
New Work Item
Sprint 1
Person: All
Collapse all
To Do
In Progress
Done
87 As a user, I want to view all my orders
Unassigned
State New
92 Design test cases for viewing order
Unassigned
State To Do
91 Develop the code for viewing orders
Unassigned
State In Progress
89 As a user, I want to filter the order with various Order Status
Unassigned
State New
90 As a user, I want to filter the Orders with various Order Time
Unassigned
State New
93 Searching the order displays some mismatch order
Unassigned
State New


anubhavbagri8 / flipkart-scrum-demo / Boards / Backlogs
Search
AB
flipkart-scrum-demo Team
Backlog
Analytics
New Work Item
View as Board
Column Options
Backlog items
Order
Work Item Type
Title
State
Effort
Value Area
Iteration Path
Tags
Epic
Order module under MyAccount
New
Business
flipkart-scrum-demo
Feature
Filtering the Order functionality
New
Business
flipkart-scrum-demo
Product Backl...
As a user, I want to filter the order with various Order S...
New
9
Business
flipkart-scrum-demo\Sprint 1
Product Backl...
As a user, I want to filter the Orders with various Order...
New
Business
flipkart-scrum-demo\Sprint 1
Feature
View and Search Order functionality
New
Business
flipkart-scrum-demo
Product Backl...
As a user, I want to view all my orders
New
15
Business
flipkart-scrum-demo\Sprint 1
Product Backl...
As a user, I want to search order so that I can view part...
New
21
Business
flipkart-scrum-demo\Sprint 2
Unparented Features
Unparented Backlog items
Bug
Searching the order displays some mismatch order
New
Business
flipkart-scrum-demo\Sprint 1

DBMS

PostgreSQL

- Open-source object-relational DB management platform
- Compatible with multiple datatypes and supports data integrity.
- JSON (non-relational) & SQL(relational) is supported by Postgresql

Normalization

- Process of effectively organizing data into multiple tables to minimize data redundancy
- First Normal Form(1NF), second normal form (2NF), third normal form(3NF), 3.5NF, 4NF and 5NF

Joins

- Join statements is used to combine data or rows from two or more tables based on common field between them
- Types of Joins – Inner, Left, Right, Outer, Cross and Equi join

```sql

-- create
CREATE TABLE Customer(
  Cust_ID int PRIMARY KEY,
  Cust_Name text,
  mobile int,
  Email_ID char(30),
  address char(30)
);

-- -- insert
INSERT INTO Customer VALUES (100, 'Bala', 9876543, 'sen@gmail.com', 'bangalore');
INSERT INTO Customer VALUES (101, 'Paapaaji', 454323, 'paapaaji@gmail.com', 'Pind');
INSERT INTO Customer VALUES (102, 'Wrik', 122312, 'wrik@gmail.com', 'bengal');

ALTER TABLE Customer add pan char(10);

\d Customer

UPDATE Customer set pan='BAD37283D' where cust_id=100;
UPDATE Customer set pan='GOODPA34D' where cust_id=101;
UPDATE Customer set pan='GRT7239D5' where cust_id=102;

-- -- fetch
SELECT * FROM Customer;

SELECT cust_name FROM Customer;


```

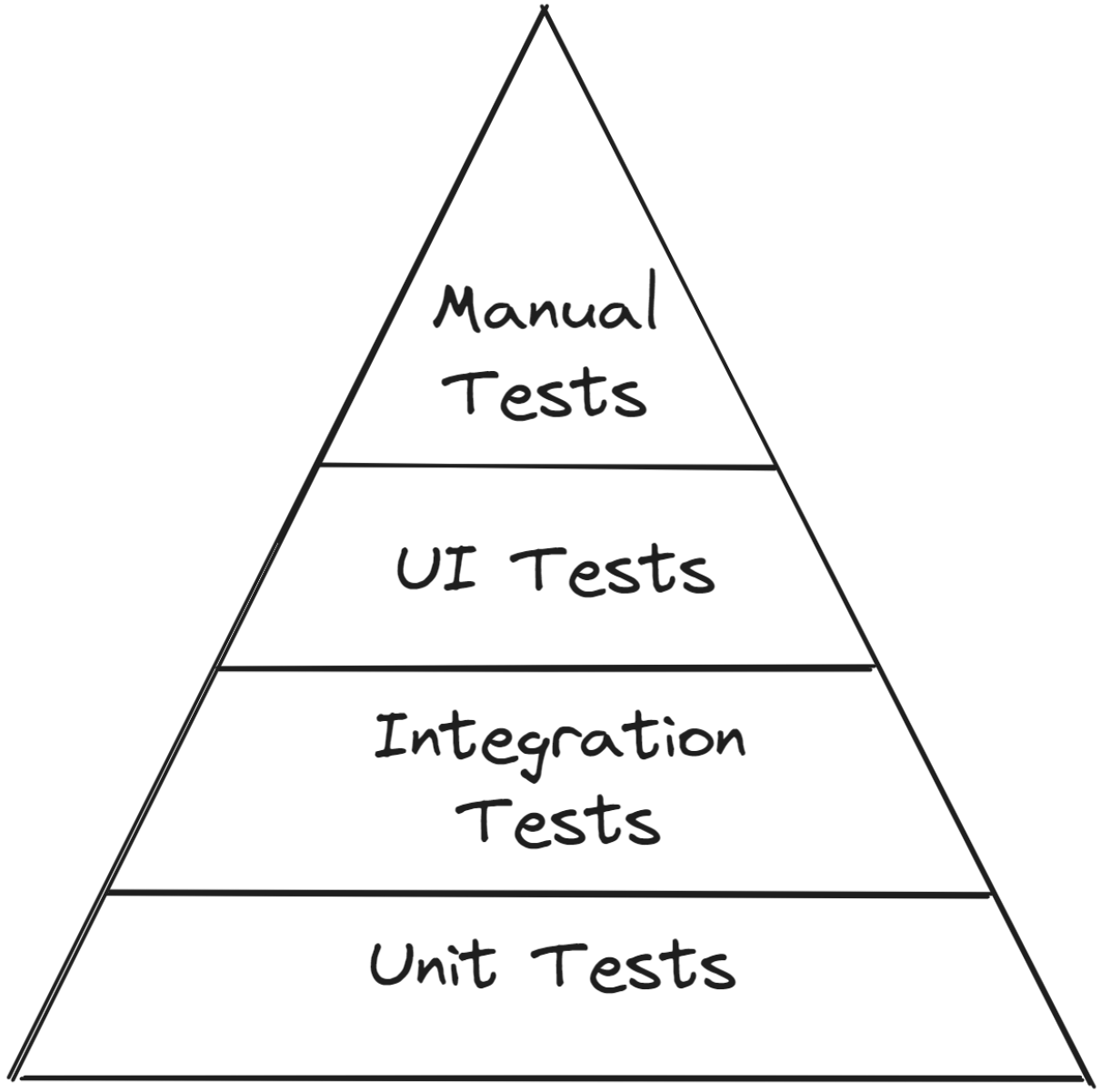
STDIN

Input for the program (Optional)

Output:

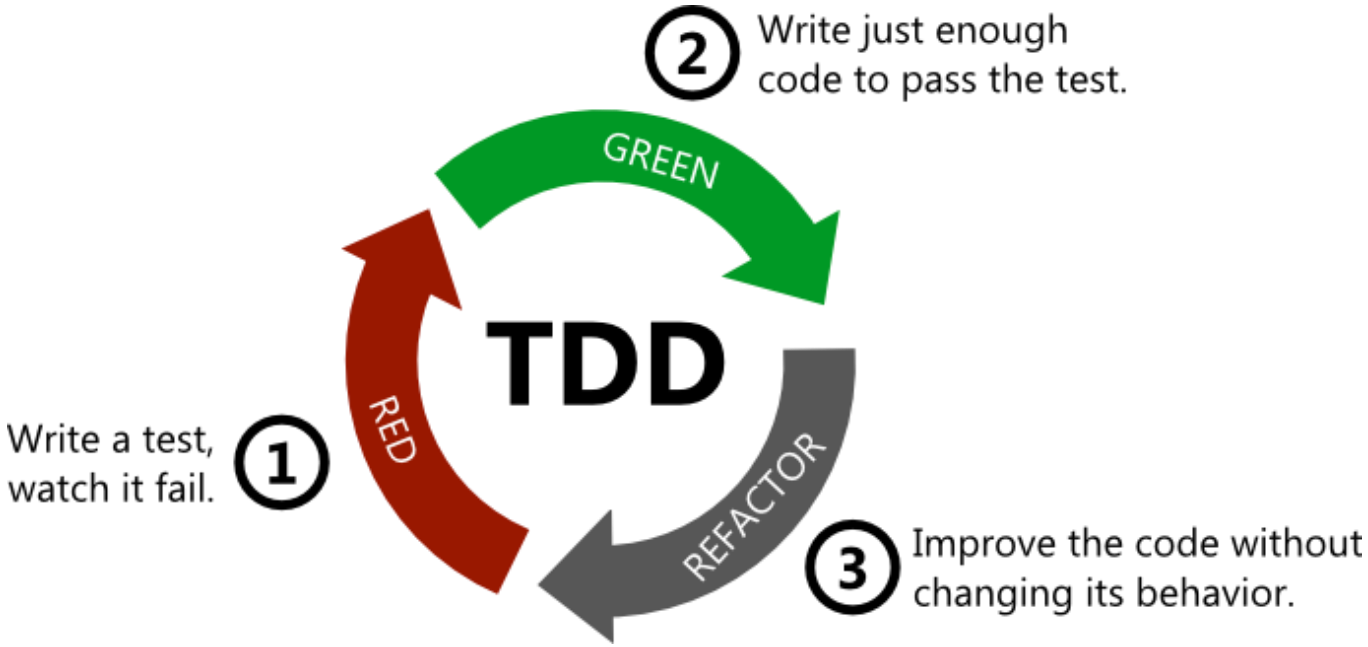
```
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
ALTER TABLE
                  Table "public.customer"
  Column   |     Type      | Collation | Nullable | Default
-----------+---------------+-----------+----------+---------
 cust_id   | integer       |           | not null |
 cust_name | text          |           |          |
 mobile    | integer       |           |          |
 email_id  | character(30) |           |          |
 address   | character(30) |           |          |
 pan       | character(10) |           |          |
Indexes:
    "customer_pkey" PRIMARY KEY, btree (cust_id)

UPDATE 1
UPDATE 1
UPDATE 1
 cust_id | cust_name | mobile  |            email_id            |
---------+-----------+---------+--------------------------------+-------
     100 | Bala      | 9876543 | sen@gmail.com                  | bangal
     101 | Paapaaji  |  454323 | paapaaji@gmail.com             | Pind
     102 | Wrik      |  122312 | wrik@gmail.com                 | bengal
(3 rows)

 cust_name
-----------
 Bala
 Paapaaji
 Wrik
(3 rows)
```

Software Testing

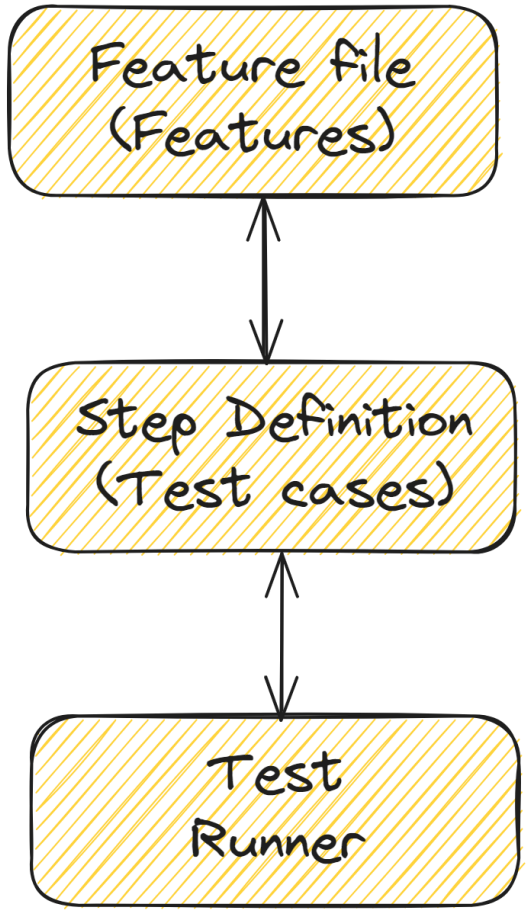


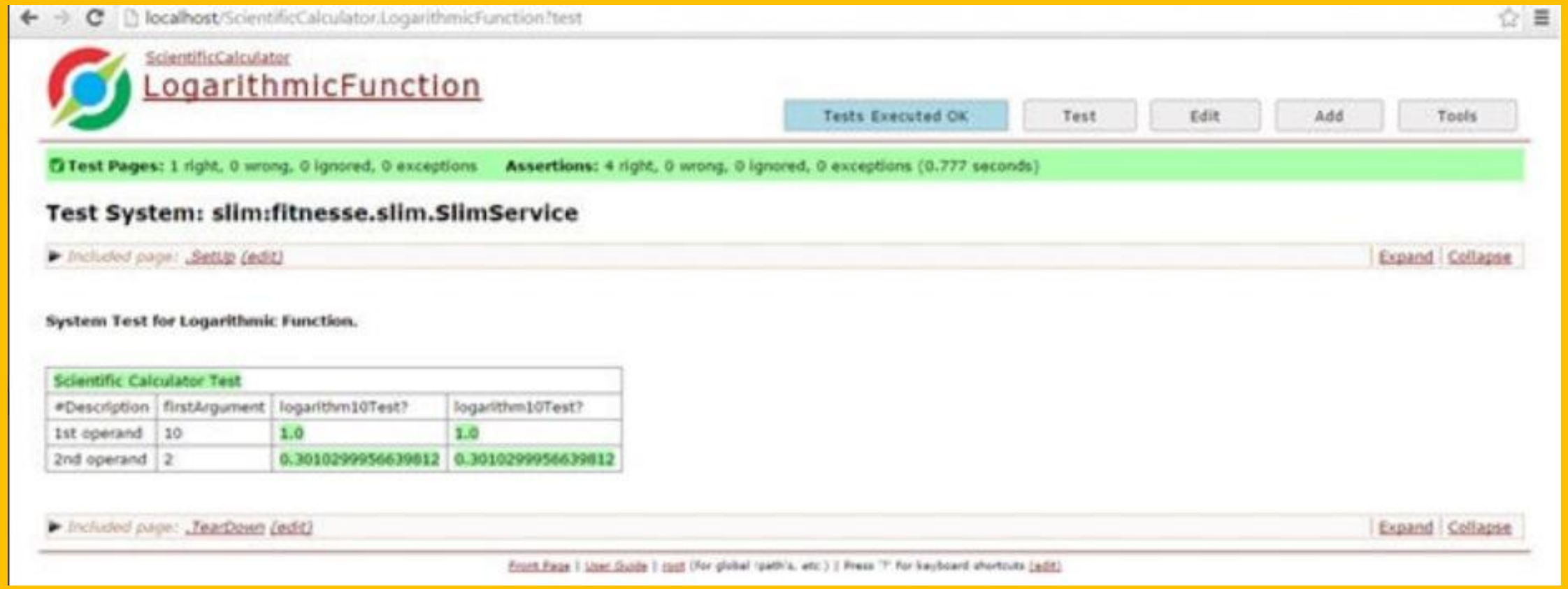
Testing Triangle

Test Driven Development



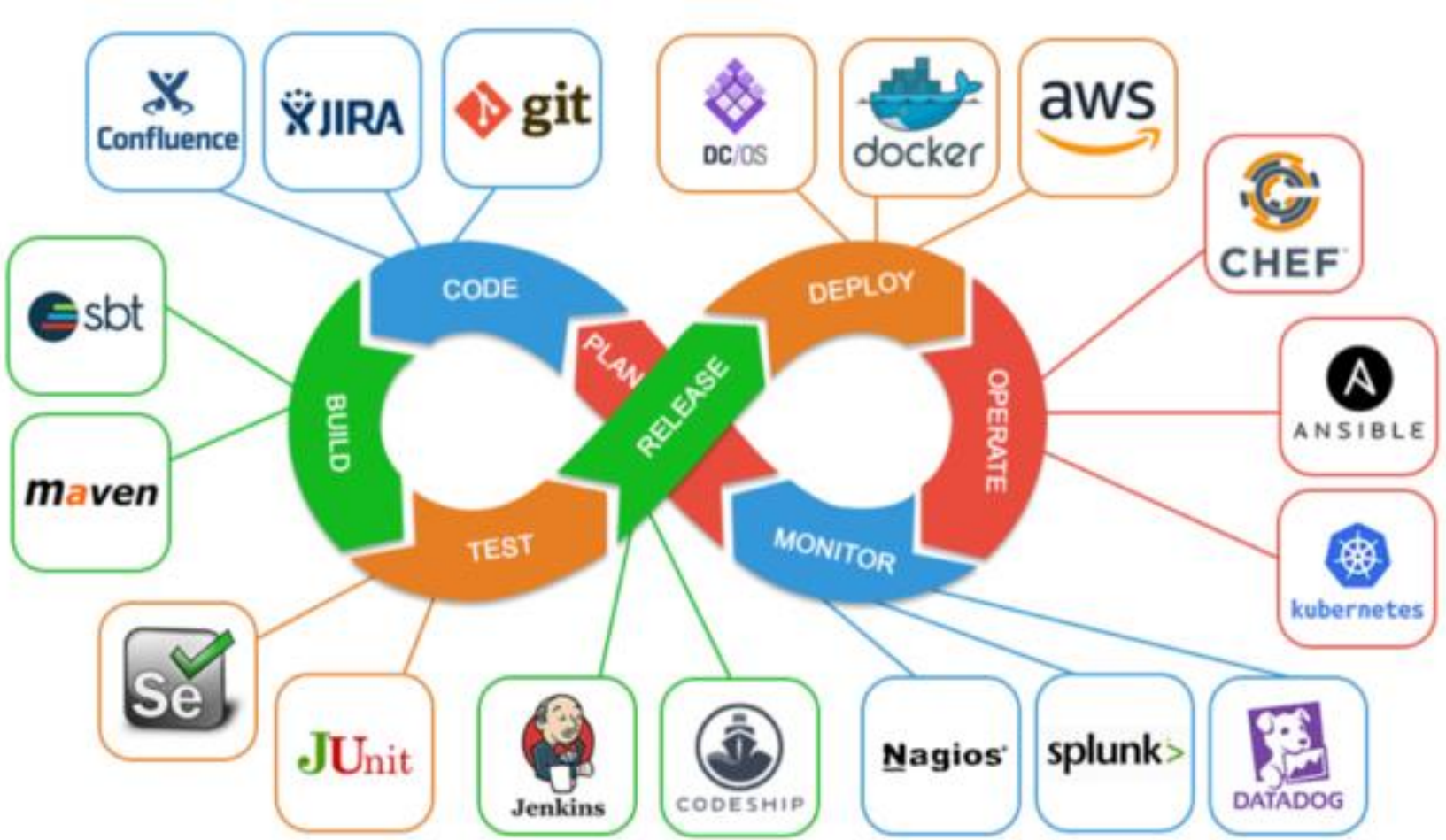
Behavioral Driven Development

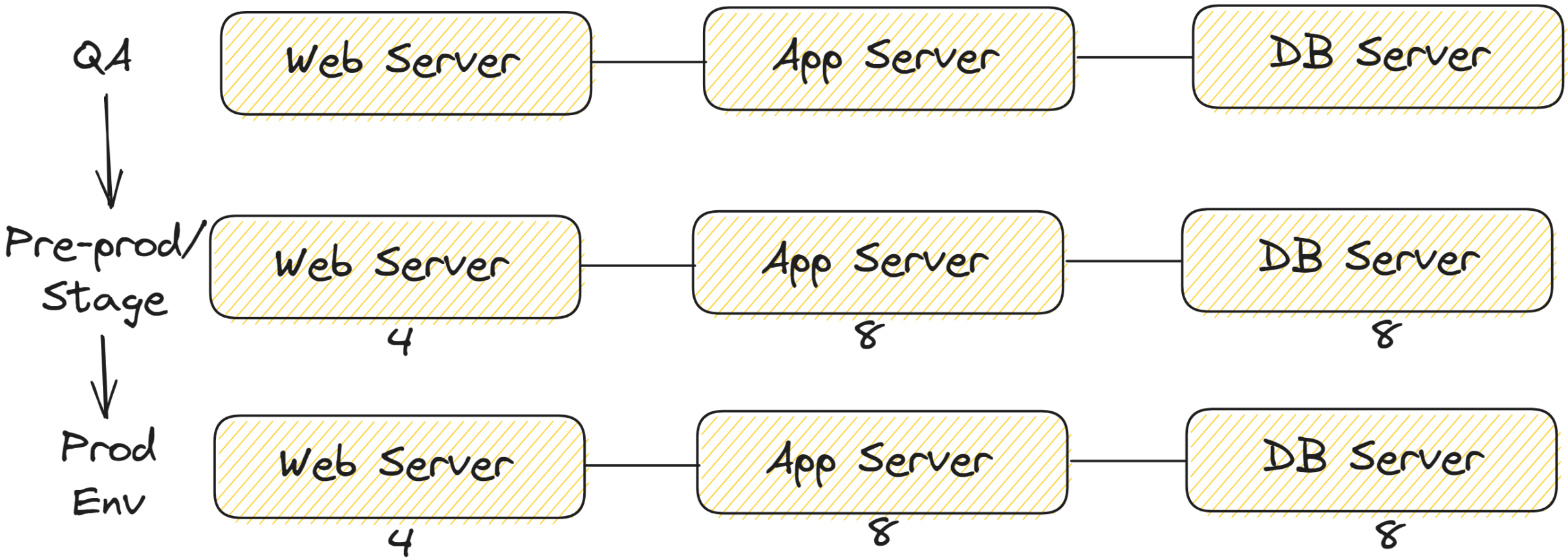



localhost/ScientificCalculator.LogarithmicFunction?test
ScientificCalculator
LogarithmicFunction
Tests Executed OK
Test
Edit
Add
Tools
Test Pages: 1 right, 0 wrong, 0 ignored, 0 exceptions
Assertions: 4 right, 0 wrong, 0 ignored, 0 exceptions (0.777 seconds)
Test System: slim:fitnesse.slim.SlimService
Included page: .SetUp (edit)
Expand
Collapse
System Test for Logarithmic Function.
Scientific Calculator Test
#Description
firstArgument
logarithm10Test?
logarithm10Test?
1st operand
10
1.0
1.0
2nd operand
2
0.3010299956639812
0.3010299956639812
Included page: .TearDown (edit)
Expand
Collapse

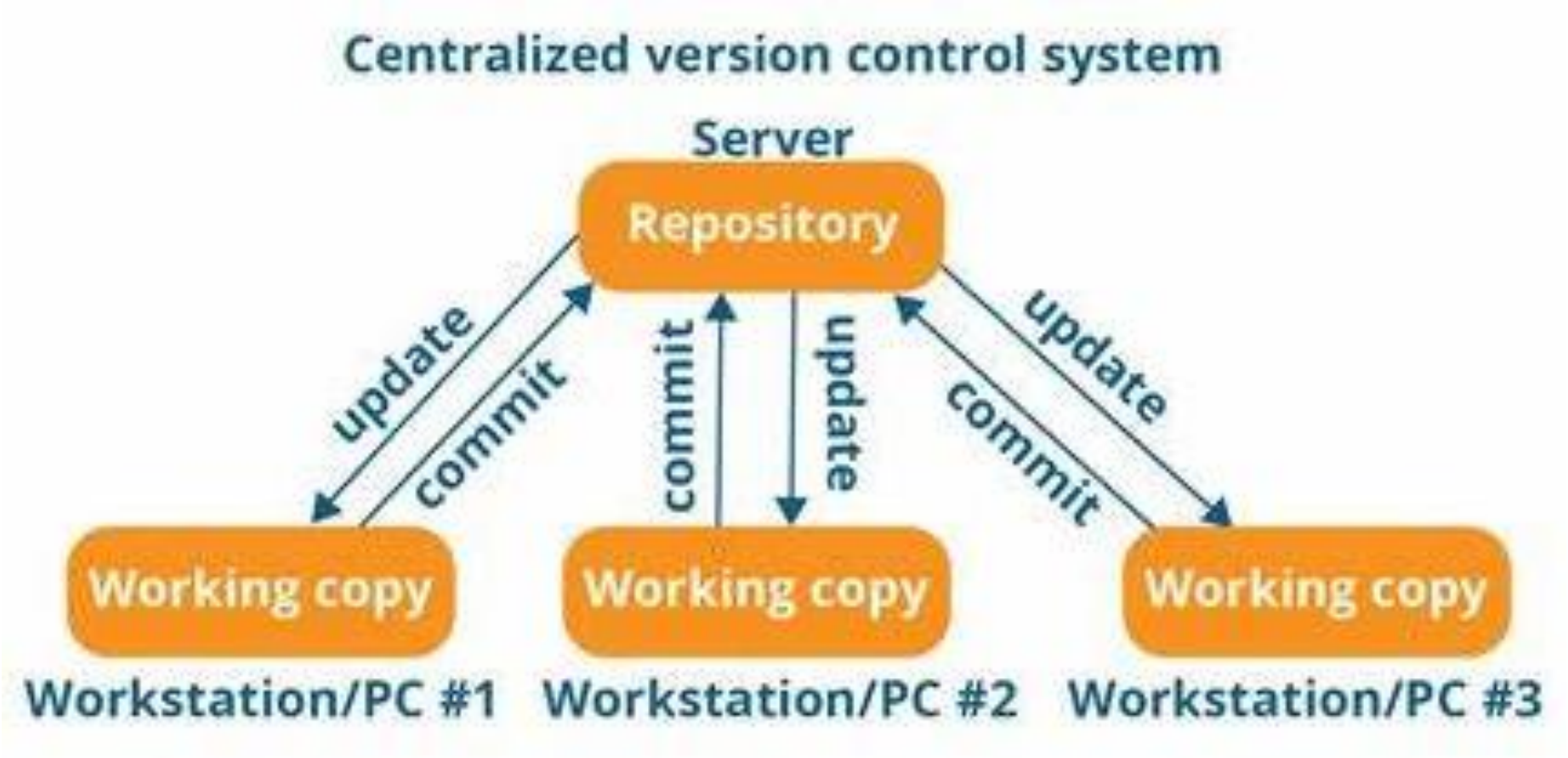
DevOps

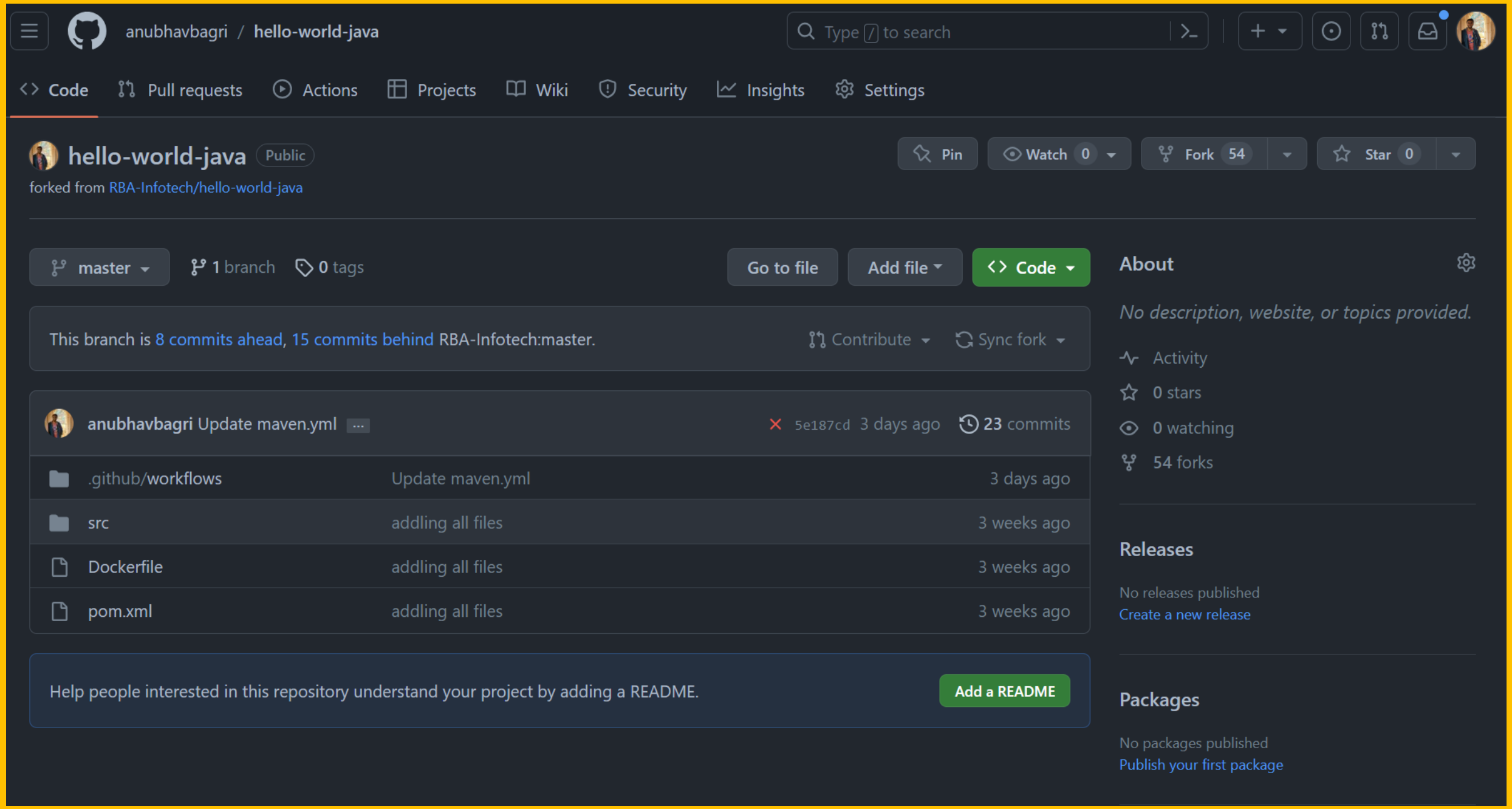
- Plan
- Code
- Build
- Test
- Release
- Deploy
- Operate
- Monitor




QA
Web Server
App Server
DB Server
Pre-prod/
Stage
Web Server
4
App Server
8
DB Server
8
Prod
Env
Web Server
4
App Server
8
DB Server
8

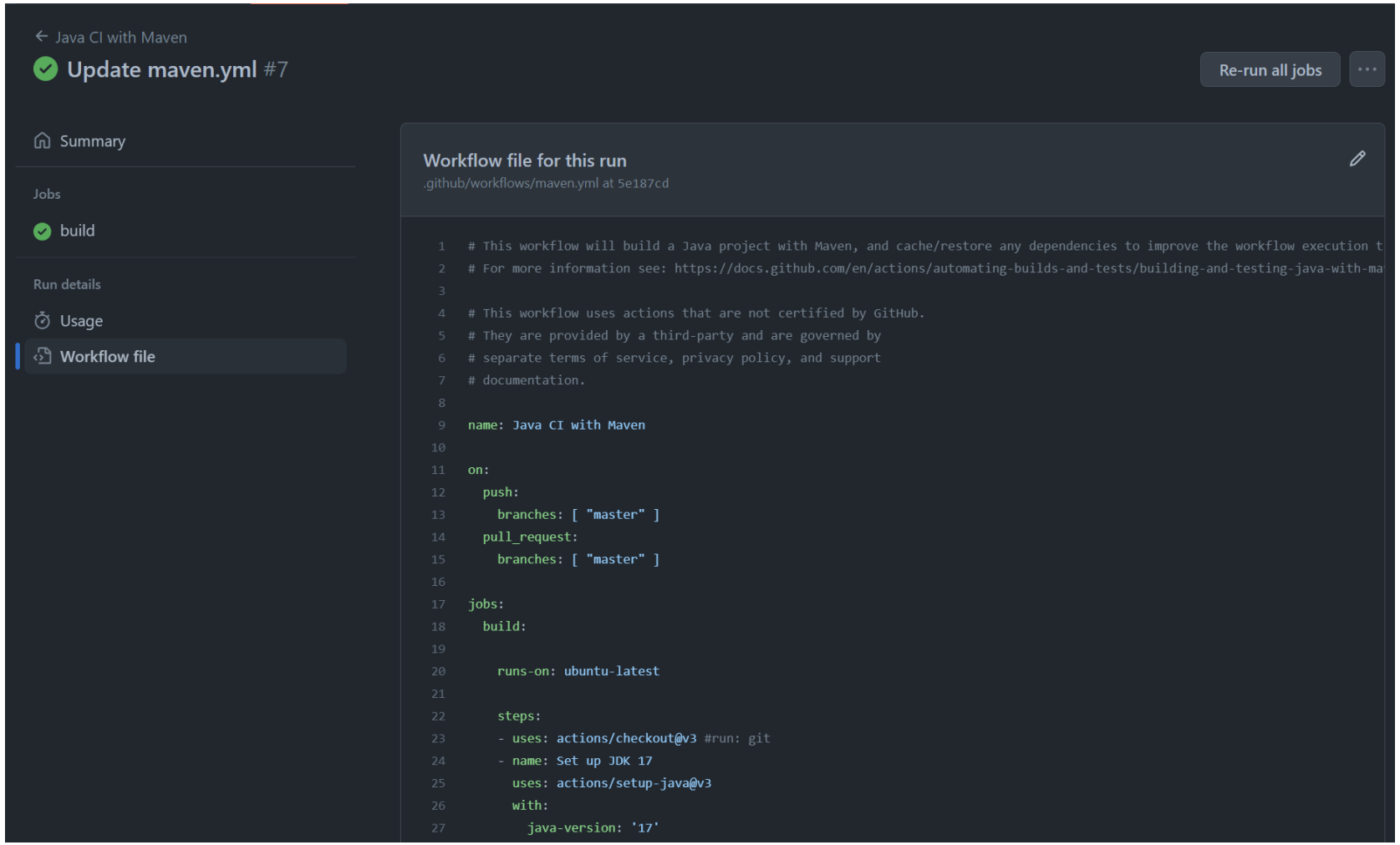
GIT VCS


Centralized version control system
Server
Repository
update
commit
commit
update
update
commit
Working copy
Working copy
Working copy
Workstation/PC #1
Workstation/PC #2
Workstation/PC #3


anubhavbagri / hello-world-java
Type / to search
Code
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
hello-world-java
Public
forked from RBA-Infotech/hello-world-java
Pin
Watch 0
Fork 54
Star 0
master
1 branch
0 tags
Go to file
Add file
Code
This branch is 8 commits ahead, 15 commits behind RBA-Infotech:master.
Contribute
Sync fork
anubhavbagri Update maven.yml
5e187cd 3 days ago
23 commits
.github/workflows
Update maven.yml
3 days ago
src
addling all files
3 weeks ago
Dockerfile
addling all files
3 weeks ago
pom.xml
addling all files
3 weeks ago
Help people interested in this repository understand your project by adding a README.
Add a README
About
No description, website, or topics provided.
Activity
0 stars
0 watching
54 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package

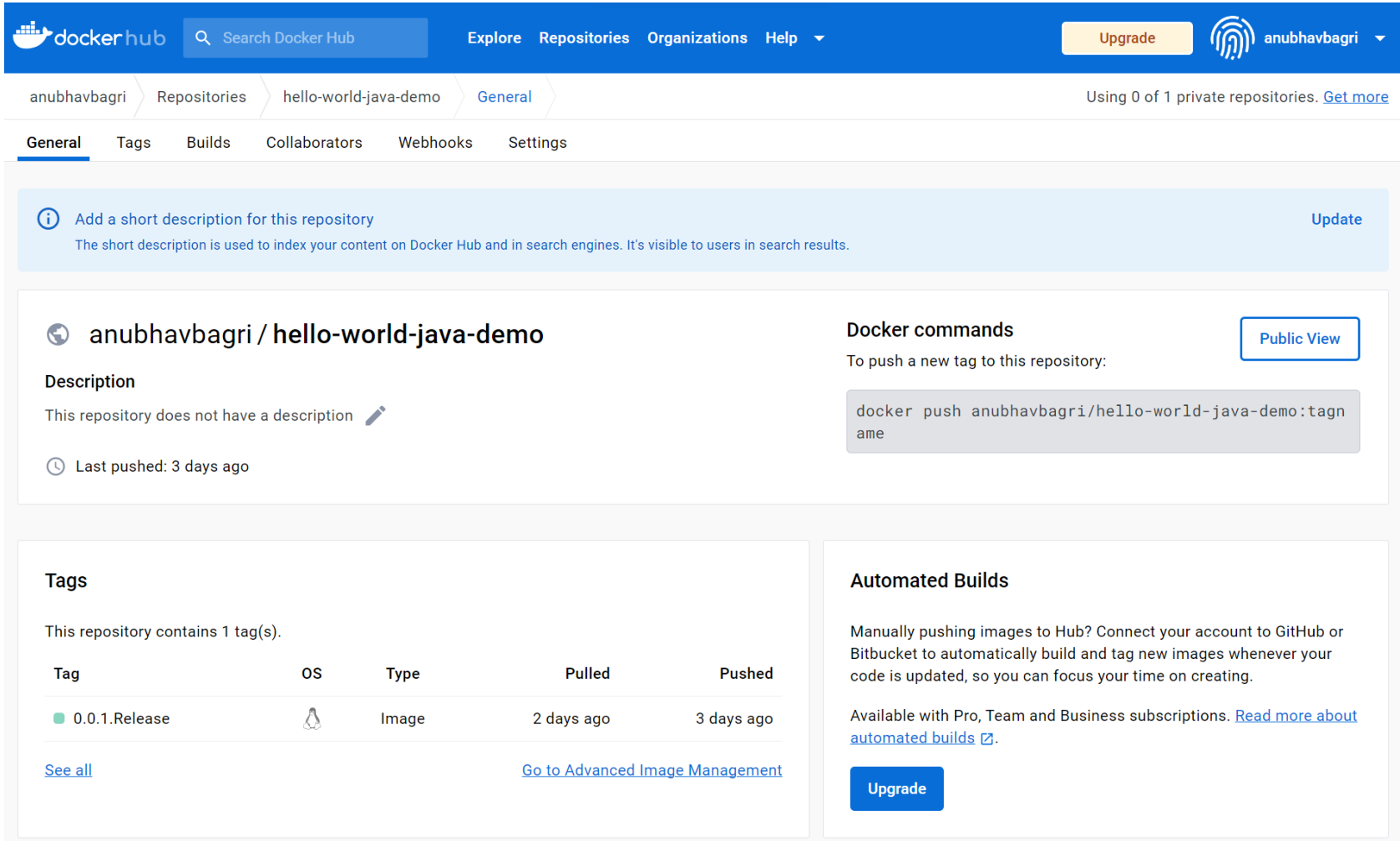
Github Actions

- Workflows
- Events
- Jobs
- Actions
- Runners



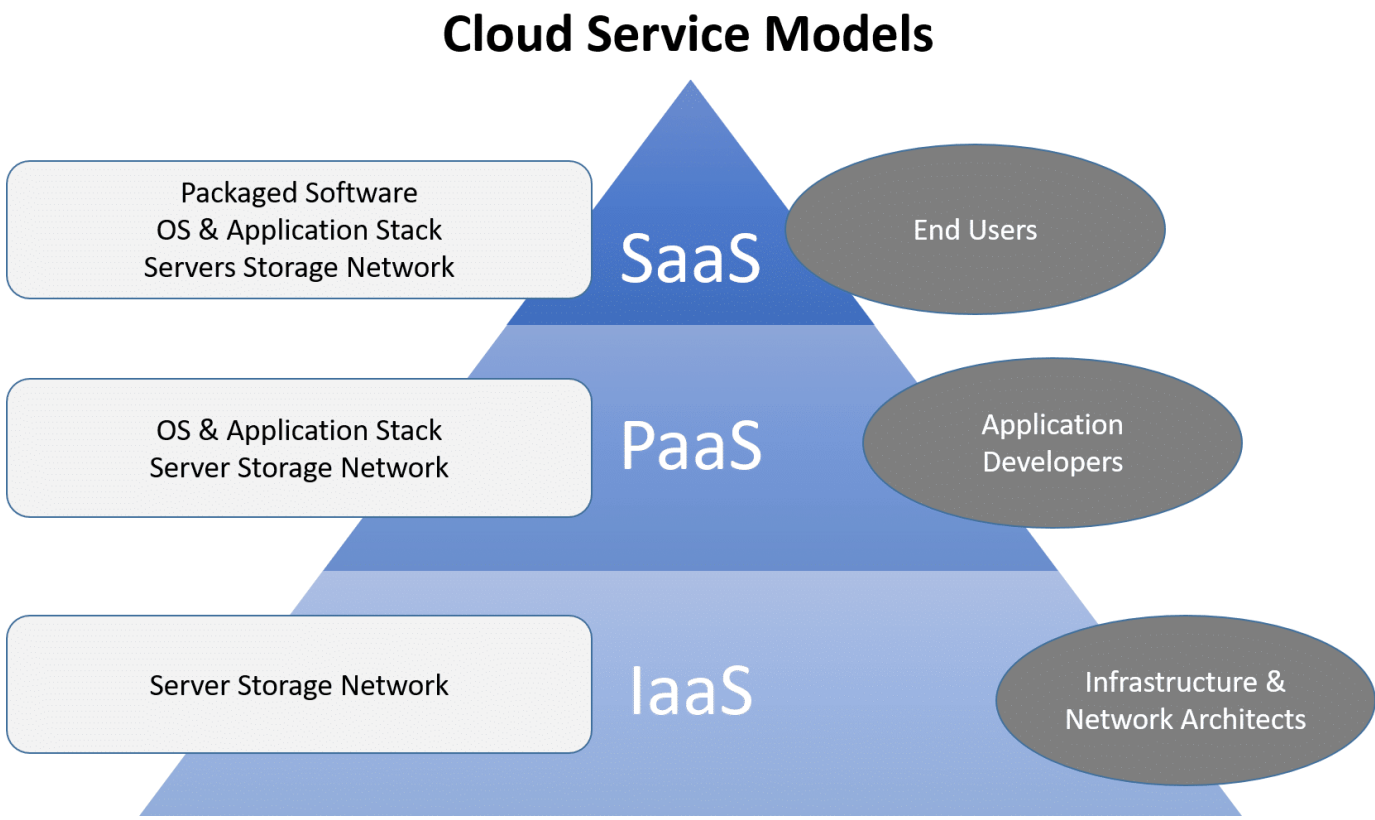
Docker

- Build Image
- Push image to docker hub
- View image in docker repo
- Image can be run using container



Cloud

- On-demand resource provisioning
- Provides resources when needed & user would release back after their usage
- Cloud services – IaaS, PaaS, SaaS
- Cloud delivery model – Private, Public and Hybrid Clouds



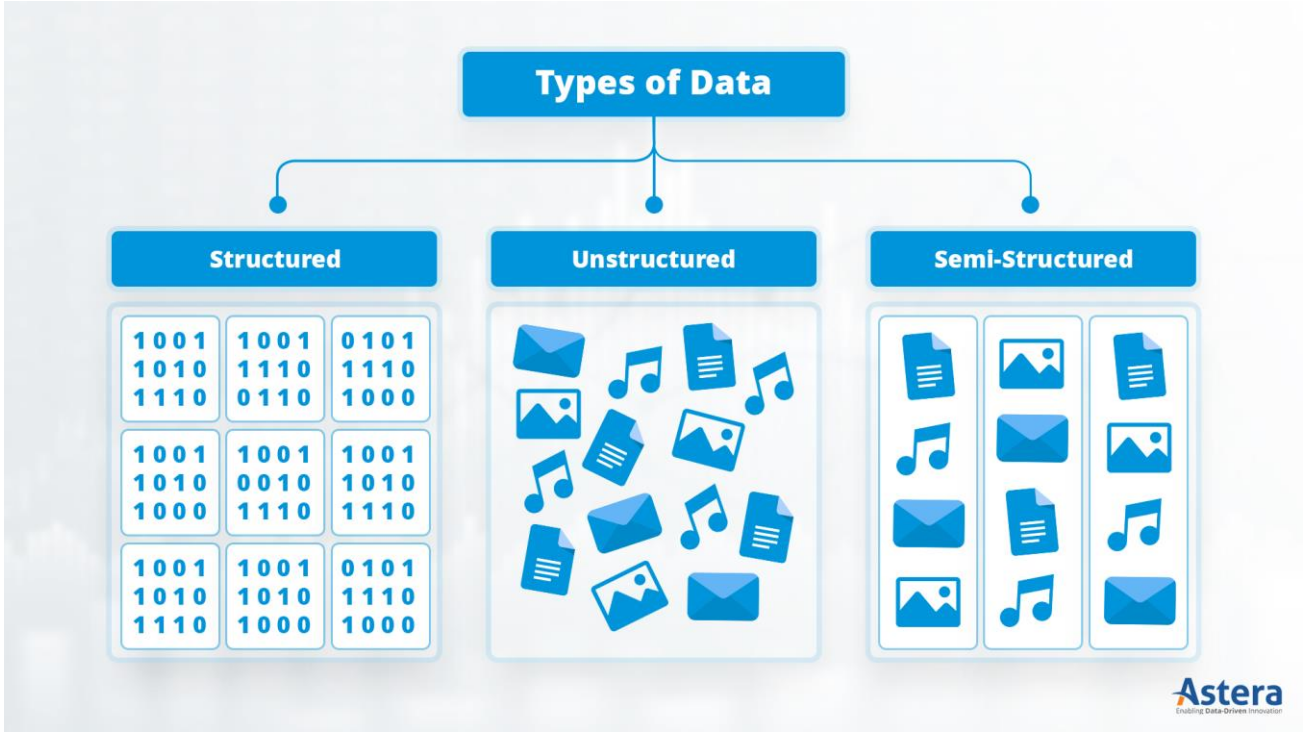
Week 2 Custom IDA Training

29 Aug 2023 – 1 Sept 2023

- Data Fundamentals (DBMS & Data Warehouse)
- Big Data Fundamentals (DataLakes)
- Azure SQL

Data

- meaningful information
- data aids in producing information (numbers) which is based on facts
- Structured Data – 2d data, rows & columns – SQL, Excel
- Semi-structured data (Tags) – JSON, XML, Parquet (compressed)
- Unstructured data – anything: audio file, video file, images



DBMS

- Contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access data (different environments like windows, android)
  - An environment that is bot convenient and efficient to use
- Database Applications
  - Banking transactions
  - Airlines: Reservation & Schedules
  - Universities: Registration & Grades
  - Sales: customer, products and purchase

RDBMS

- Data is stored in a table called relations: Primary Key – contains unique info(e.g., Employee code), Foreign Key – must be present in the parent table or else violation would occur;
- Relations can be normalized (remove redundancy)
- Each row in a relation contains a unique value (Primary Key)
- Attributes, column, tuple, table(relation)
  - Collection of columns – attributes
  - Collection of rows – tuples
- Each column in a relation contain values from same domain
  - e.g.: Employee table & Salary table – part of the same Database domain

Data Model

- Data Constraints
  - Default constraint
  - Check constraint
  - Primary key constraint
  - Foreign key constraint
- Organizes the data elements and standardize how the data elements relate to each other (Primary key – foreign key relationship)
- Defines how logical structure of a database is modelled (basically how the data is presented)

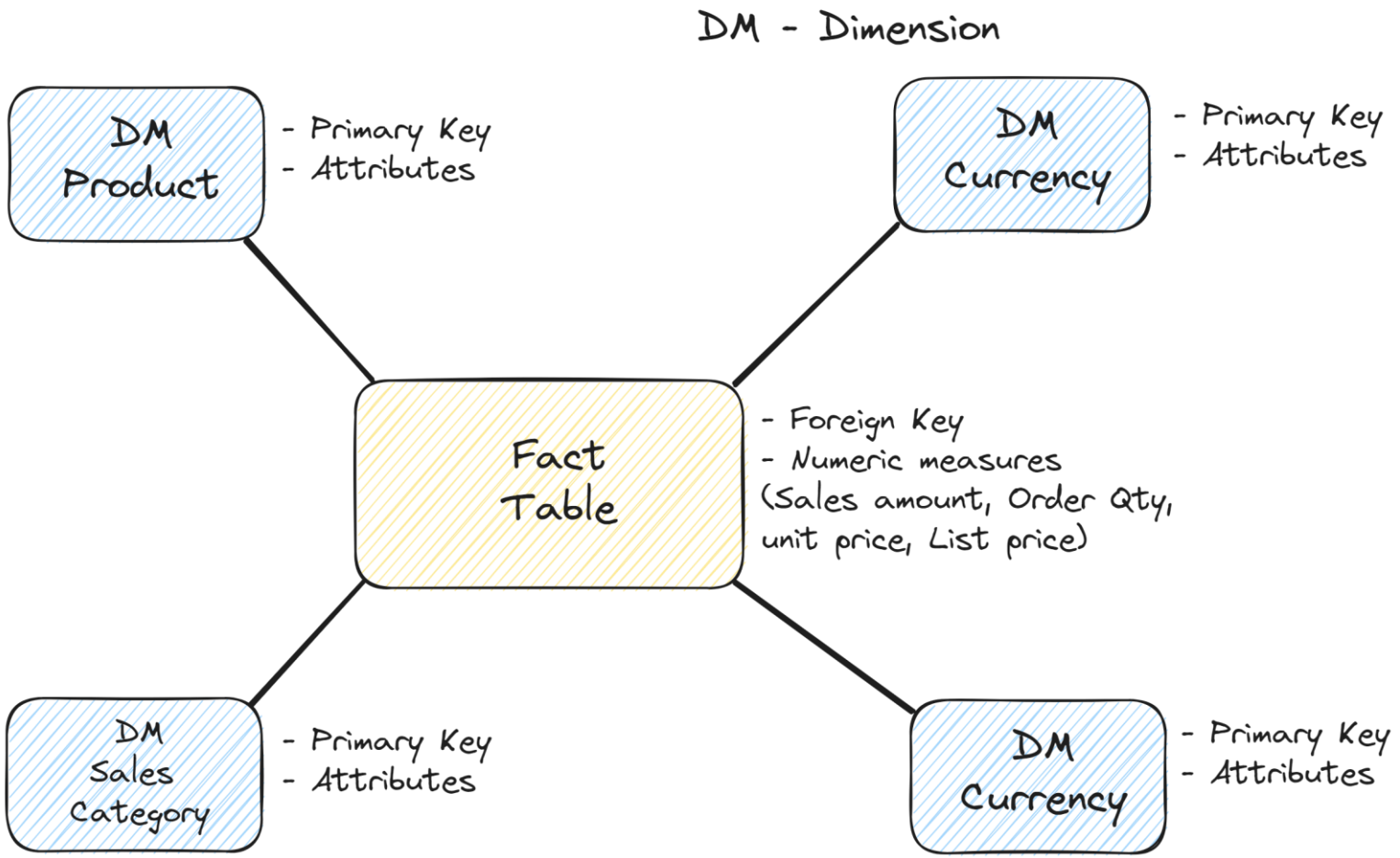
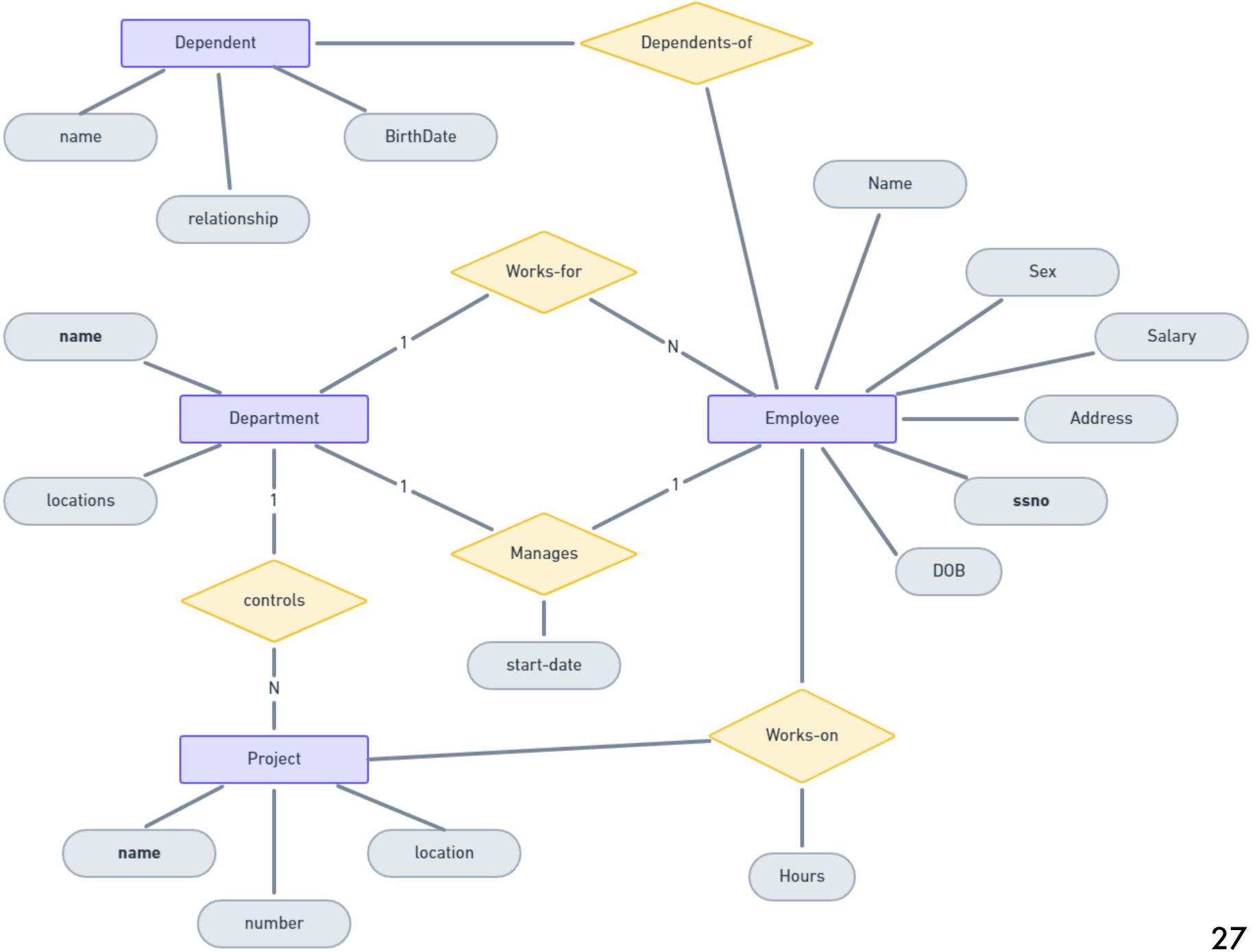


Fig - Data Model

ER DIAGRAM


Dependent
Dependents-of
name
BirthDate
relationship
Name
Works-for
Sex
name
Salary
1
N
Department
Employee
Address
locations
1
1
1
ssno
Manages
DOB
controls
start-date
N
Works-on
Project
name
location
Hours
number

Functional Dependency

- Concept that specifies the relationship between two sets of attributes where one attribute determines the value of another attribute.
- Denoted by X -> Y, where the attribute set on the left side of the arrow, X is called Determinant and Y is called the Dependent. Eg: Emp_Id -> Emp_name
- Trivial Functional Dependency
  - A -> B has trivial dependency if B is a subset of A.
  - The following dependencies are also trivial like A -> A, B -> B
- Non-Trivial Functional Dependency
  - A -> B has non trivial dependency if B is not a subset of A.

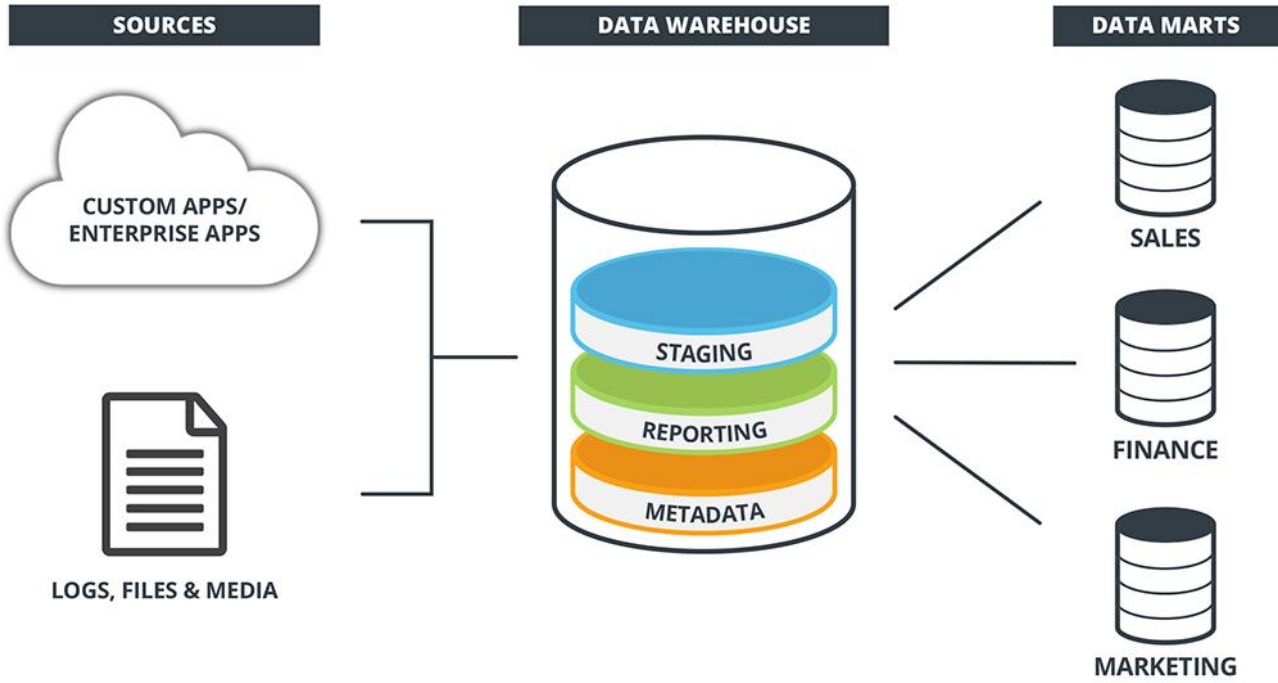
Data Warehousing

Central repository to store huge amount of data

- The Business Problem
  - Business data is spread across many systems.
  - Data can be inconsistent, duplicated and contradictory.
  - Fundamental questions can’t be easily answered.
- Data warehouse contains large volumes of historical data
- Is optimized for querying, as opposed to inserting or updating data
- Is incrementally loaded with new business data at regular intervals
- Provides the basis for enterprise BI solutions

Data Warehouse Architecture

- Data sources – from where the data originates/ from where we are extracting the data [ETL : Extract, Transformation, Load]
- 3 types
  - Central Data warehouse
  - Departmental Data Marts (limited to a data warehousing solution)
  - Hub-and-Spoke



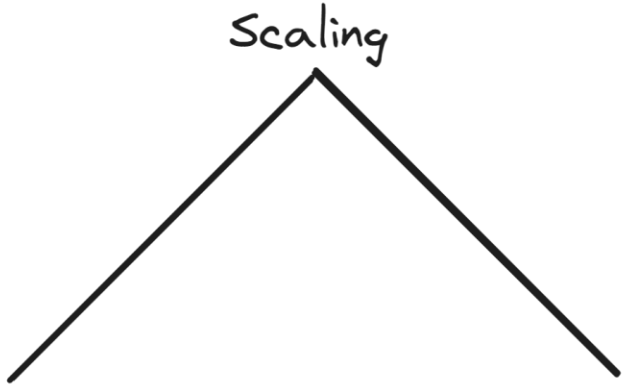
Transactions: OLTP & OLAP

- OLTP – Online Transaction Process: Database (Joins & Subsequences), Create, Read, Update, Delete
- OLAP – Online Analytical Process: [Dimensions & Facts] Datawarehouse (2-dimension -> historical) & Cubes (3-dimension)
- Schema – Star Schema & Snowflake Schema
  - Star schema is most used programming approach – All dimensions connect to the fact measures
  - Snowflake schema is a process where dimensions are connected with sub-dimensions and sub-sub-dimensions

BIG DATA

- Massive volume of both structured, semi-structured and unstructured data that is so large that its difficult to process using traditional database and software techniques
- Big data is high Volume, high Velocity, and high Variety information assets that require new forms of processing to enable enhanced decision making, insight, discovery, and process optimization
- Why Big Data?
  - Businesses have thrived on their ability to derive insights from information or data
  - It has helped them make better, smarter, real time, and fact-based decisions

Scaling



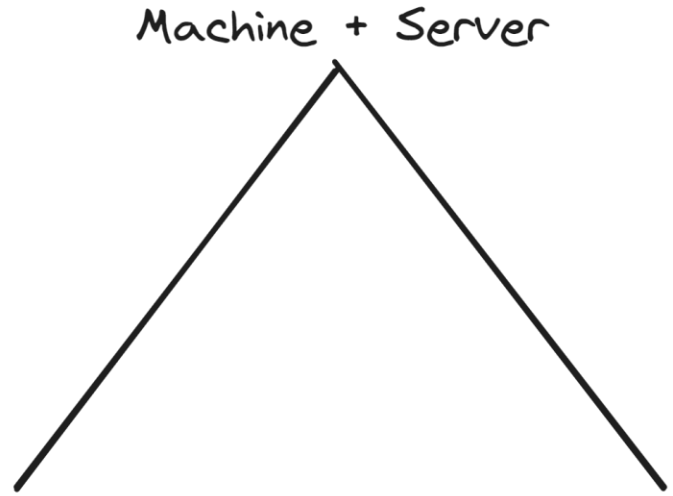
Vertical Scaling

- upgrading the same hardware
- eg. replacing 2 cores CPU with 4 cores CPU
- 4GB -> 8GB
- 20GB -> 60GB

[scale Up - Scale Down]

Horizontal Scaling

- Adding completely new hardware
- eg. adding another CPU with 2 cores
- [Scale out - Scale in]



Physical Requirement

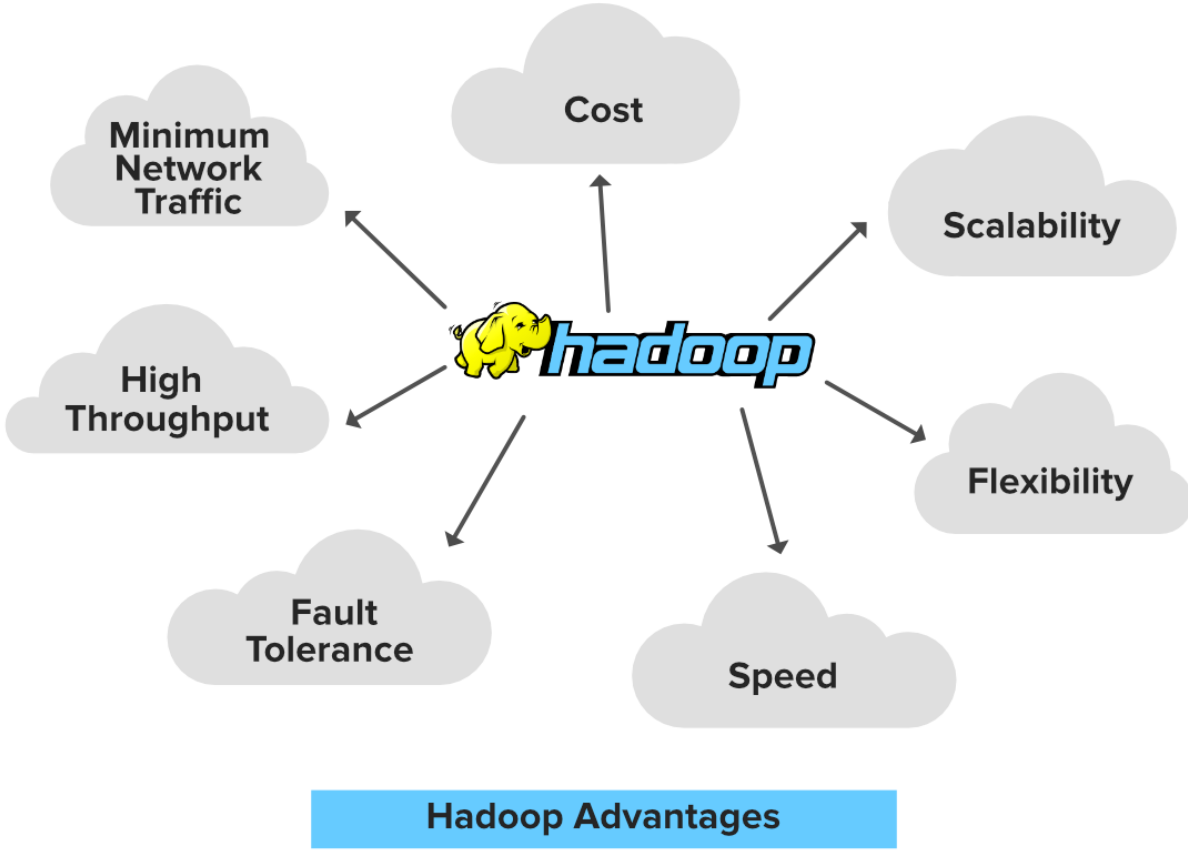
- Storage disk
- Processor
- RAM

Software Requirement

- OS: License
- S/W: License

Hadoop

- Framework that allows for the distributed processing of large data sets across dusters of computers using simple programming models
- Hadoop Architecture
  - R: Randomized
  - A: Access
  - I: Independent
  - D: Disk
- RAID 0 (Striping – diving your information into multiple strips), RAID 1 (Mirroring), RAID 5 (Parity), RAID 1+0 = 10 (Striping + Mirroring)



CLOUD COMPUTING

Cloud computing is the delivery of computing services over the internet, enabling faster innovation, flexible resources, and economics of scale

- Private Cloud
  - Single instance of the software runs on a server
  - Serves a single client organization, and is managed by a third party
- Public Cloud
  - Owned by cloud services or hosting provider
  - Provides resources and services to multiple organizations and users
  - Accessed via secure network connection (typically over the internet)
- Hybrid Cloud
  - Combines Public & Private clouds to allow applications to run in the most appropriate location

Cloud Services

IAAS

Build pay-as-you-go IT infrastructure by renting servers, VMs, storage, networks, and operating systems from a cloud provider

PAAS

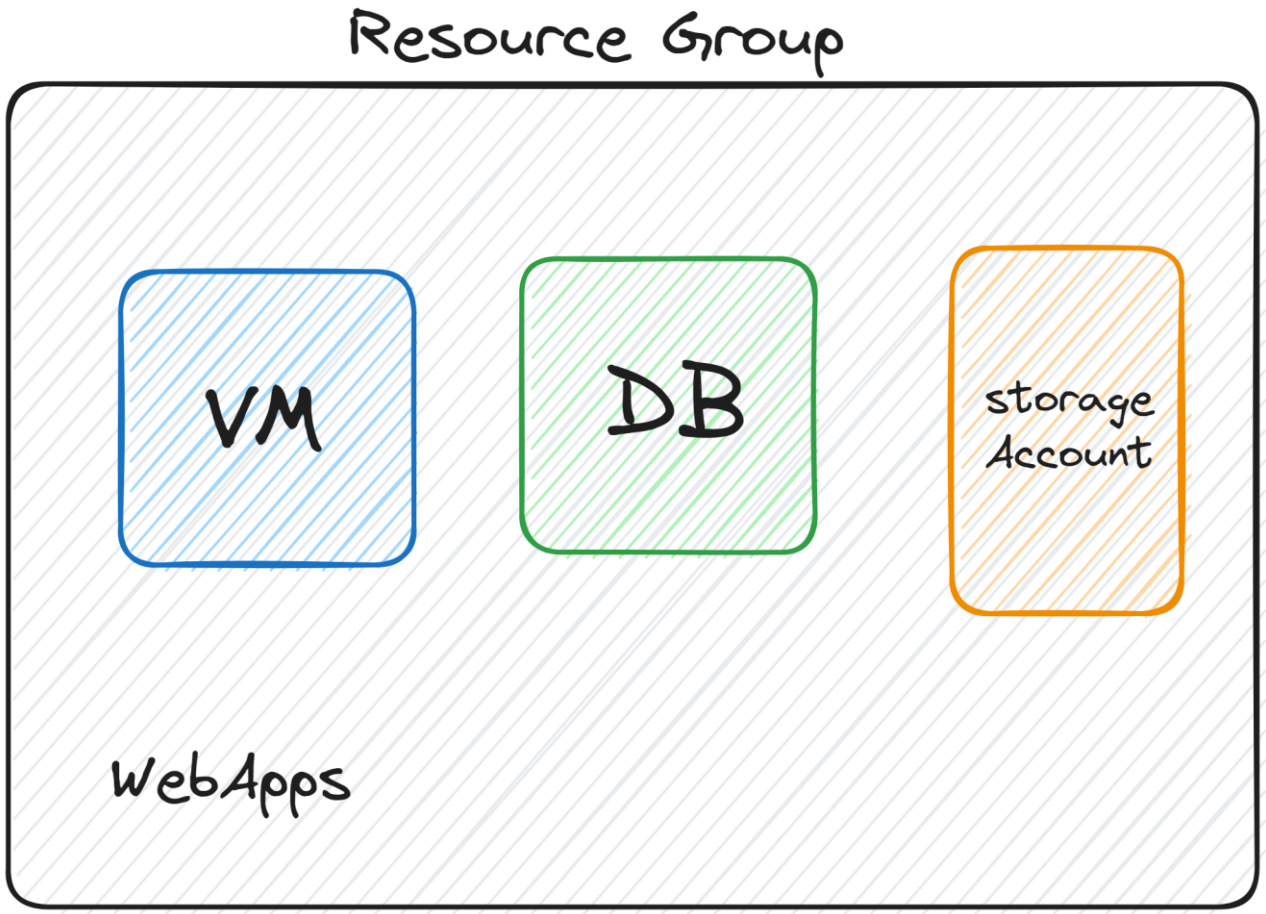
Provides environment for building, testing, and deploying software applications; without focusing on managing underlying infrastructure

SAAS

Users connect to and use cloud-based apps over the internet

Azure Resource Group

- A resource group is a container to manage and aggregate resources in a single unit
  - Resources can exist in only one resource group
  - Resources can exist in different regions
  - Resources can be moved to different resource groups
  - Applications can utilize multiple resource groups

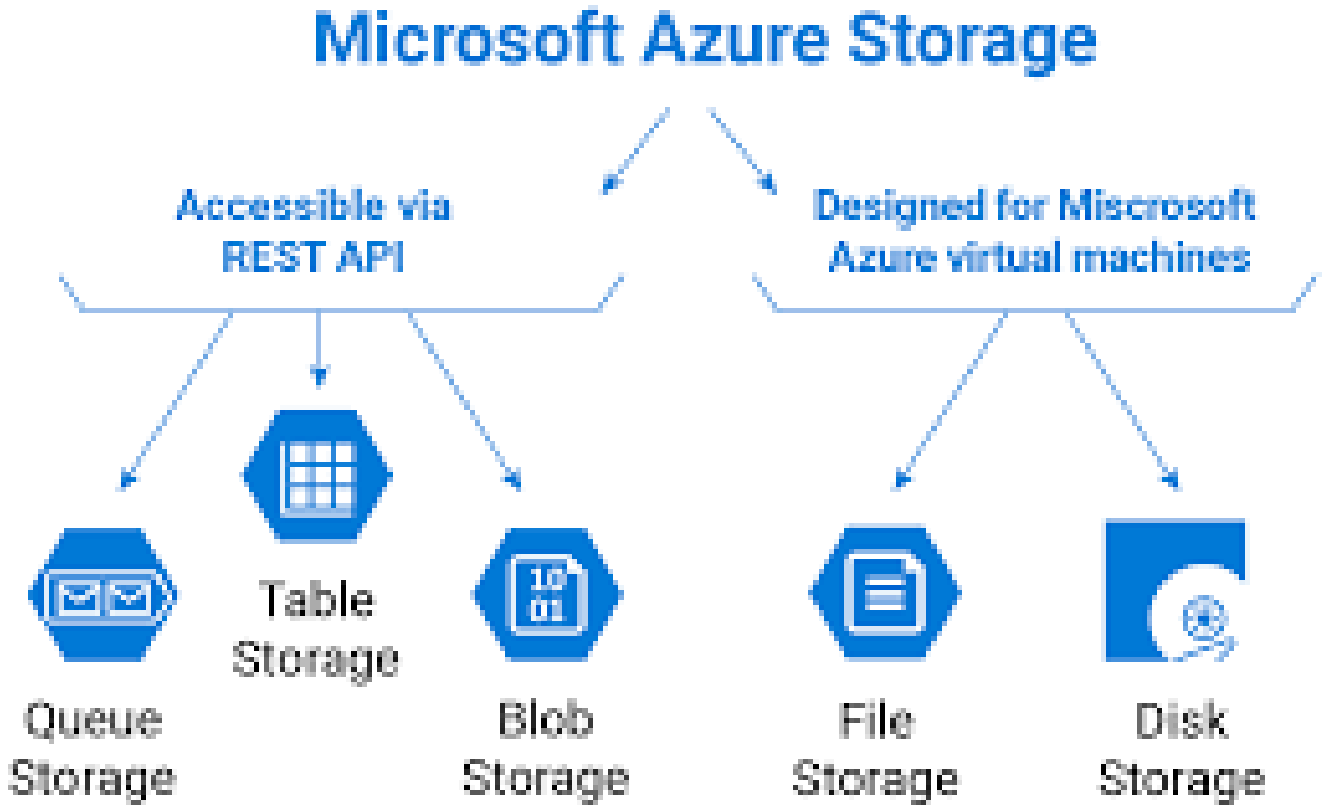


AZURE CONTAINER SERVICES

- Light-weight, virtualized environment that does not require OS management, & can respond to changes on demand
- Azure Container Instances – a PaaS offering that runs a container or pod of containers in Azure
- Azure Container Apps – a PaaS offering like container instances that can load balance and scale
- Azure Kubernetes Service – an orchestration service for containers with distributed architectures

Azure Storage services

- Azure Blob
- Azure Disk
- Azure Queue
- Azure Files
- Azure Tables

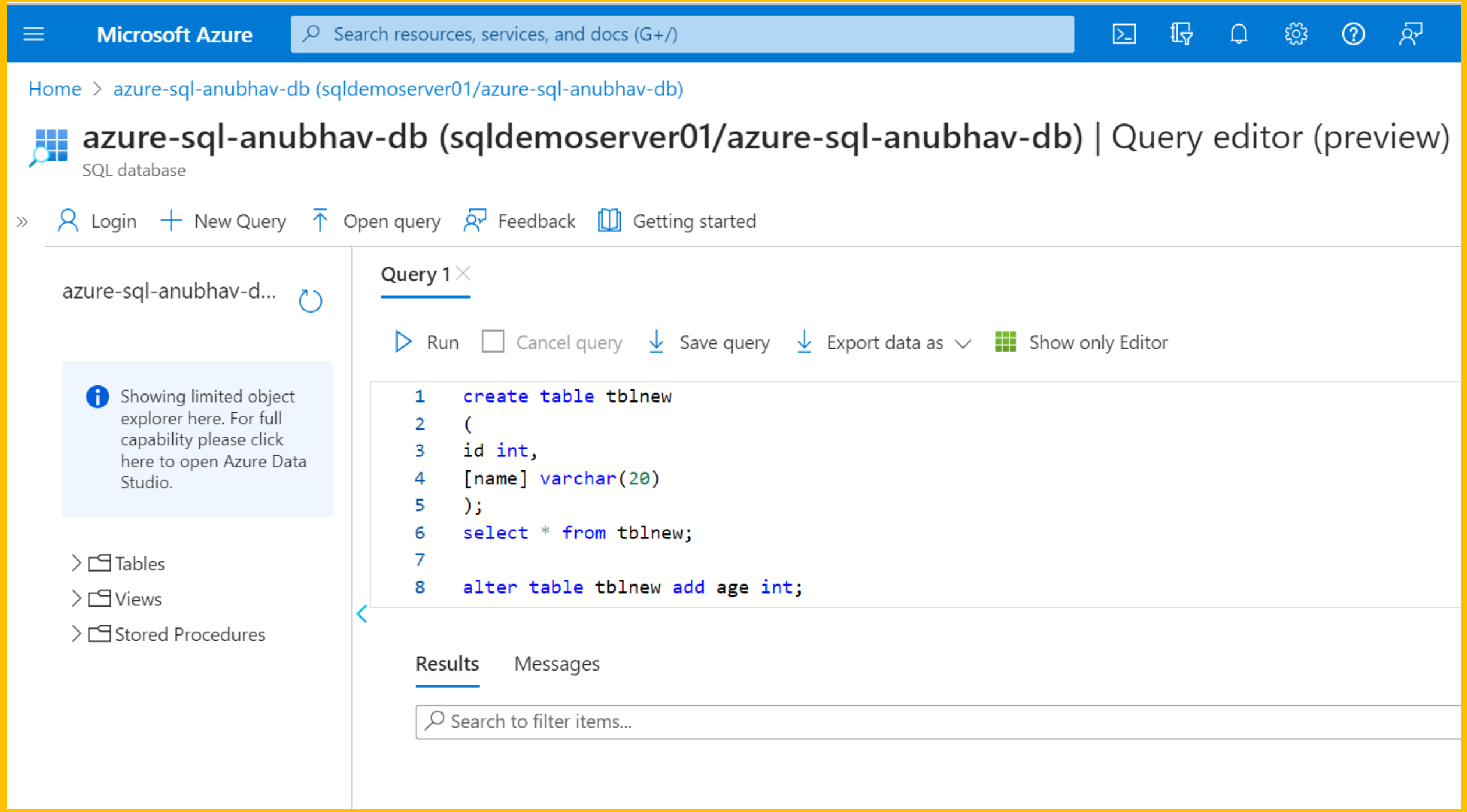


What to use for Data in Azure?

- Storage Account
  - Low cost, High throughput data store
  - Used to store No-SQL data
  - When you do not need to query the data directly, No ad hoc query support
  - Suits the storage of relatively static data
  - Acting as a HDInsight Hadoop data store
- Data Lake Store
  - Low cost, High throughput data store
  - Unlimited storage for No-SQL data
  - When you do not need to query the data directly, No ad hoc query support
  - Suits the storage of relatively static data
  - Suits acting as a Databricks, HDInsight and IOT data store
- Azure Databricks
  - Eases the deployment of Spark-based culture
  - Fastest processing of ML solutions, Enables collaboration

AZURE: SQL

DDL	DML	DCL
Data Definition Language (Physical Structure) • Create: Physical Design • Alter: Modify Physical Design • Drop: Deleting Physical Design	Data Manipulation Language (Logical Information) • Insert • Update • Delete • Filter • Truncate	Data Central Language (Permissions Information) • Grant • Revoke • Deny


Microsoft Azure
Search resources, services, and docs (G+/)
Home > azure-sql-anubhav-db (sqldemoserver01/azure-sql-anubhav-db)
azure-sql-anubhav-db (sqldemoserver01/azure-sql-anubhav-db) | Query editor (preview)
SQL database
Login
New Query
Open query
Feedback
Getting started
azure-sql-anubhav-d...
Showing limited object explorer here. For full capability please click here to open Azure Data Studio.
Tables
Views
Stored Procedures
Query 1
Run
Cancel query
Save query
Export data as
Show only Editor
1 create table tblnew
2 (
3 id int,
4 [name] varchar(20)
5);
6 select * from tblnew;
7
8 alter table tblnew add age int;
Results
Messages
Search to filter items...

CREATE TABLE: CONSTRAINTS

- Primary Key constraint – Unique values, no null allowed
- Foreign Key constraint (Reference based constraint)
- Unique constraint – Same as primary key constraint except that only single value allowed
- Default Constraint – Default
- Check constraint – e.g. Age >= 18

SQL: JOIN

Combine rows from multiple tables by specifying matching criteria

- Based on primary key – foreign key relationships
- Inner Join: Matching Data
- Outer Join
  - Left outer join: All values from left table + matching values from right table + those values which are not matched will be denoted as NULL
  - Right outer join: All values from right table + matching values from left table + unmatched values denoted as NULL
  - Full outer join: All information will return [Left + right outer join]
- Cross Join: Table1 * Table2 [Cartesian Product]
- Self Join: Connecting the same table with a different alias

SQL: SUBQUERIES

- Scalar subquery: Returns single value to outer query
  - Can be used anywhere single-valued expression is used
- Multi-valued subquery: Returns multiple value as a single column set to the outer query
  - Used with IN predicate
- Self-contained or Correlated Subqueries: Have no connection with the outer query other than passing results to it; Correlated subqueries refer to elements of tables used in outer query

SQL: SET OPERATORS

- INTERACTION BETWEEN SETS:
  - The results of two input queries may be further manipulated
  - Both sets must have the same number of comparable columns
- UNION Operator:
  - Returns set of distinct rows combined from both input sets
  - Duplicates are removed during query processing (affects performance)
- UNION ALL: Returns a result set with all rows from both input sets
- INTERSECT:
  - Returns set of distinct rows that appear in both input results
- EXCEPT: It returns distinct set of rows that appear in left set but not on the right

SQL: VIEWS

- First line of security
- They are based on Existing table data
- Will not be able to update multiple column
- Referred in the SELECT statement

EXECUTING STORED PROCEDURES

- Stored procedures are collections of T-SQL statements stored in a database
- Can return results, manipulate data, perform administrative actions on the server
- Can provide a trusted application programming interface;
- Used through the Batch Statement

BUILT-IN FUNCTIONS

Can be categorized by scope of input and type of output:

- Scalar- Operate on single row, return a single value
- Grouped- Take one or more values but return a single summarizing value
- Rowset- Return a virtual table that can be used in T-SQL statement
- Window- Operate on a window(set) of rows

SQL: INDEXES

- Clustered – stores data ins sorted format to enable fast searching
  - When a primary key is created a clustered index is created
  - Value<=Required Nearest to that
  - Navigate to the next page/pointer
- Non-Cluster – created to locate rows in a table without a clustered index (called a heap)
- Column store – Rows that store data in a column based format

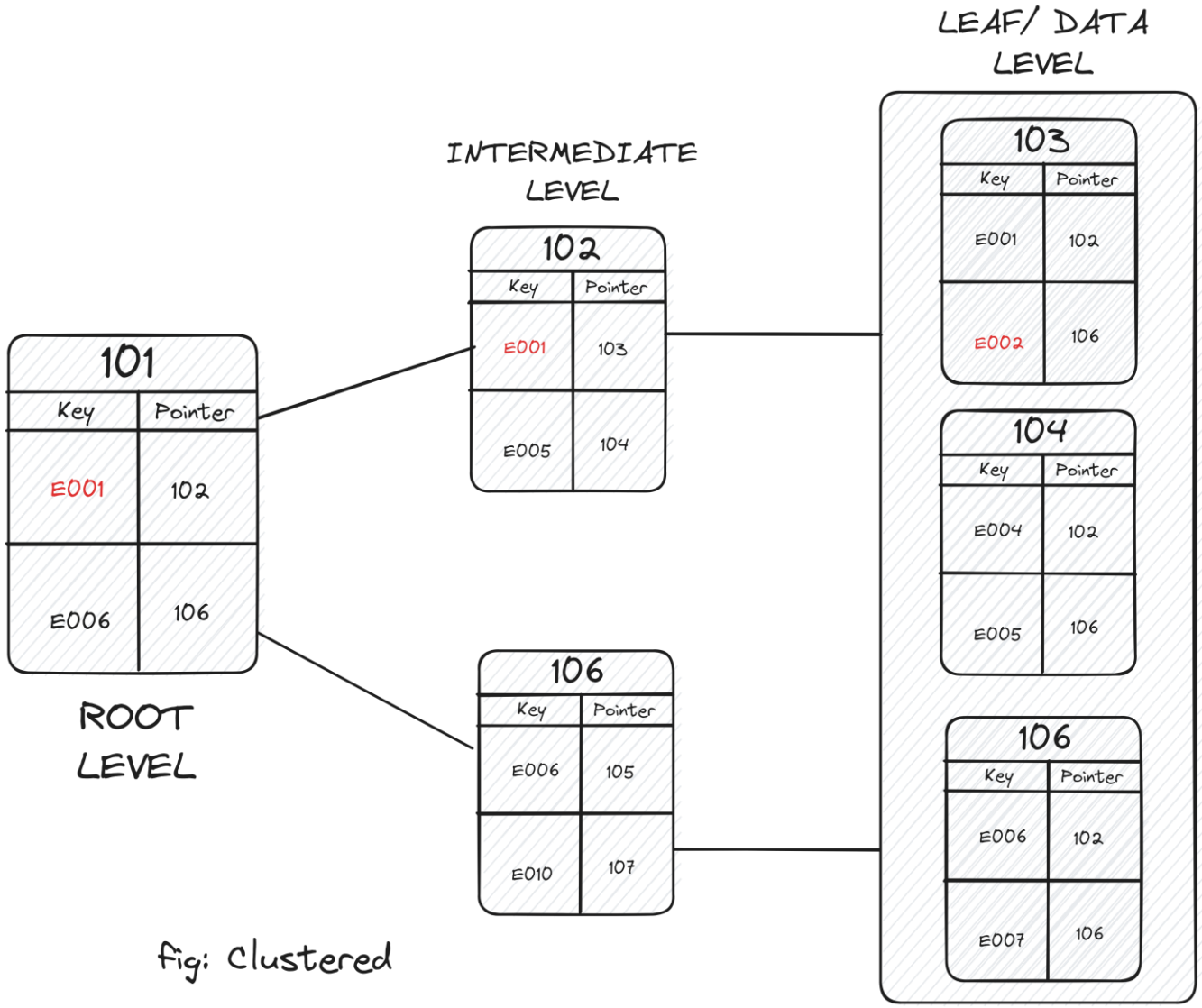


fig: Clustered

Week 3 Custom IDA Training

4 Sept 2023 – 8 Sept 2023

- Azure Storage Types
- Azure Data Factory

AZURE DATA STORAGE

- Storage – Simple Storage & ADLS(Azure Data Lake Storage)
- Benefits of using Azure to store data
  - Automated Backup: helps in mitigating the risk of losing data
  - Support for data analytics: performing analytics on data consumption eg. live feeds
  - Global replication: choose to replicate information in multiple locations
  - Storage tiers: modes - Hot, Cool, Cold, Archive
  - Encryption capabilities: carries transparent data encryption to secure data
  - Virtual disks
  - Multiple data types: can store any type of data- structured, semi-structured, unstructured

COMPARISION: On-premise vs On-Cloud

On-Premise	On-Cloud (Opex)
Cost-effectiveness • Purchase, install, maintain resources • Upgrade: capex • Power, colling: maintain Reliability • Backup, recovery • Purchase new machine Storage – Type • Required new servers & administrative tools	Cost-effectiveness • Pay as you go • Scalable without any upfront [auto-scaling] Reliability • Provides these features with no upfront cost Storage – Type • Provides distributed access & tiered storage

STORAGE ACCOUNTS

Storage account – container that groups a set of Azure storage services. Only data services can be included in a storage account such as Azure, blobs, azure Queues, & Azure tables

How many do you need? – the number of storage accounts you need is typically determined by your data diversity, & tolerance for management overhead

Number of storage accounts you need is based on:

- Data diversity: organizations often generate data that differs in where it is consumed and how sensitive it is
- Cost sensitivity: settings you choose for the account do influence the cost of services, and the number of accounts you create
- Management overhead: each storage account requires some time and attention from an administrator to create & maintain

Microsoft Azure | Search resources, services, and docs (G+/) | Shellunext_1693422079... UNEXT (NPUNEXT.ONMICROSOF...

Home > Storage accounts > anubhavazurestorage

Storage accounts

Unext (npunext.onmicrosoft.com)

Create | Restore | ...

Filter for any field...

Name

anubhavazurestorage

Page 1 of 1

anub
Storage a

Search

Data Lake
Resource sh
Advisor rec
Endpoints
Locks

Monitoring

Insights
Alerts
Metrics
Workbooks
Diagnostic
Logs

Monitoring (cla

Metrics (cla
Diagnostic
Usage (clas

Automation

Tasks (previ
Export temp

azure-storage-alert-rule-1

Alert details

Severity
3 - Informational

Fired time
9/4/2023, 12:00 PM

Affected resource
anubhavazurestor...

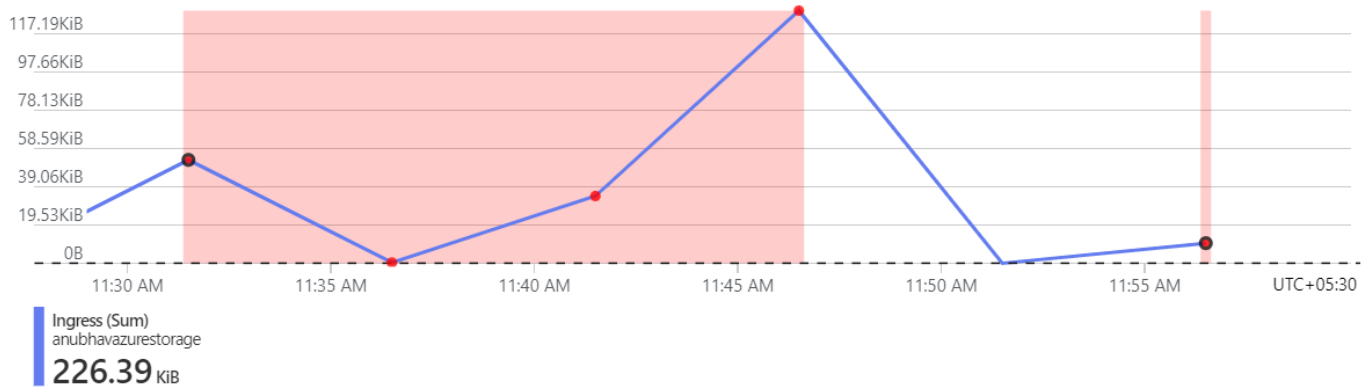
Hierarchy
npunext-16735049... > anubhav-storag...

User response
New

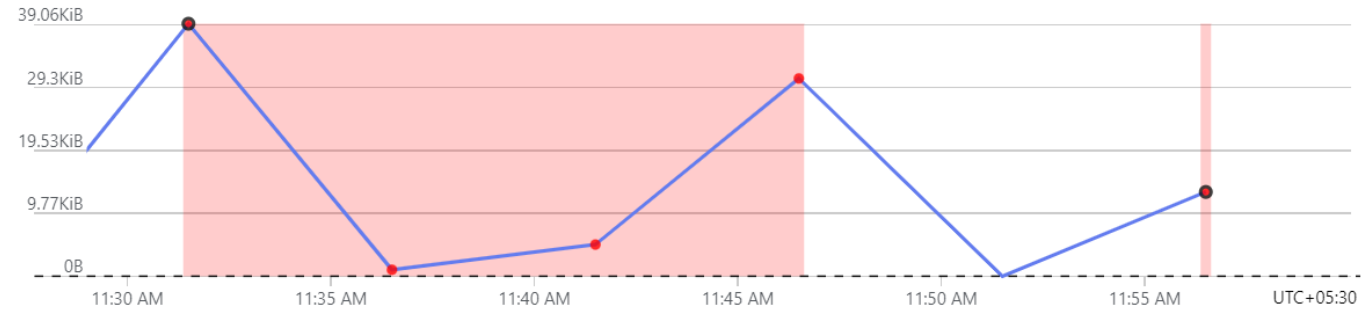
Alert condition
Fired

Change user response

Why did this alert fire?



Ingress (Sum)
anubhavazurestorage
226.39 KiB



Egress (Sum)
anubhavazurestorage
89 KiB

AZURE DATA LAKE STORAGE

Azure data lake storage – gen 2:

- Hadoop access
- Security: ACL (Access Control List)
- Performance
- Redundancy

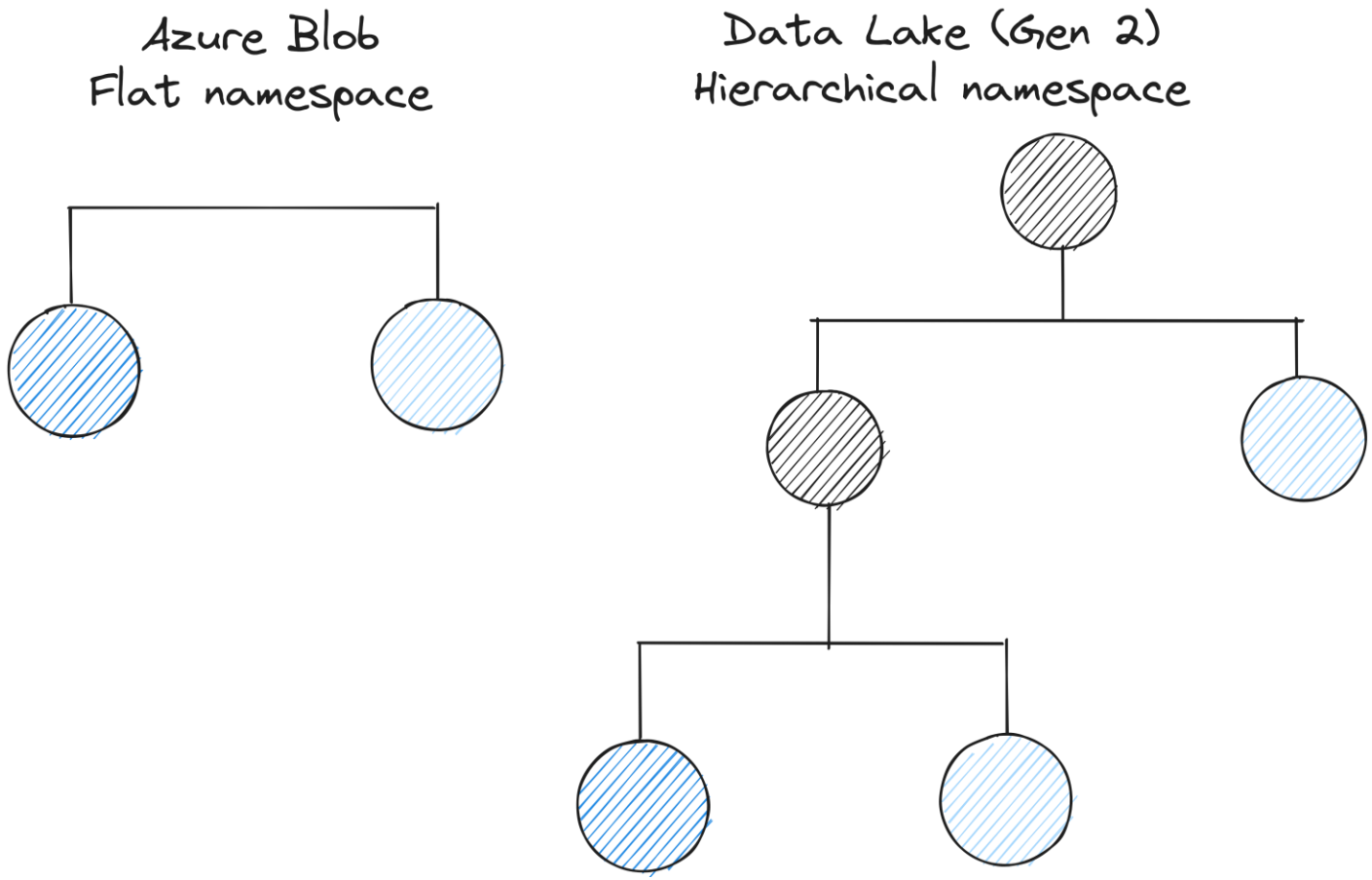
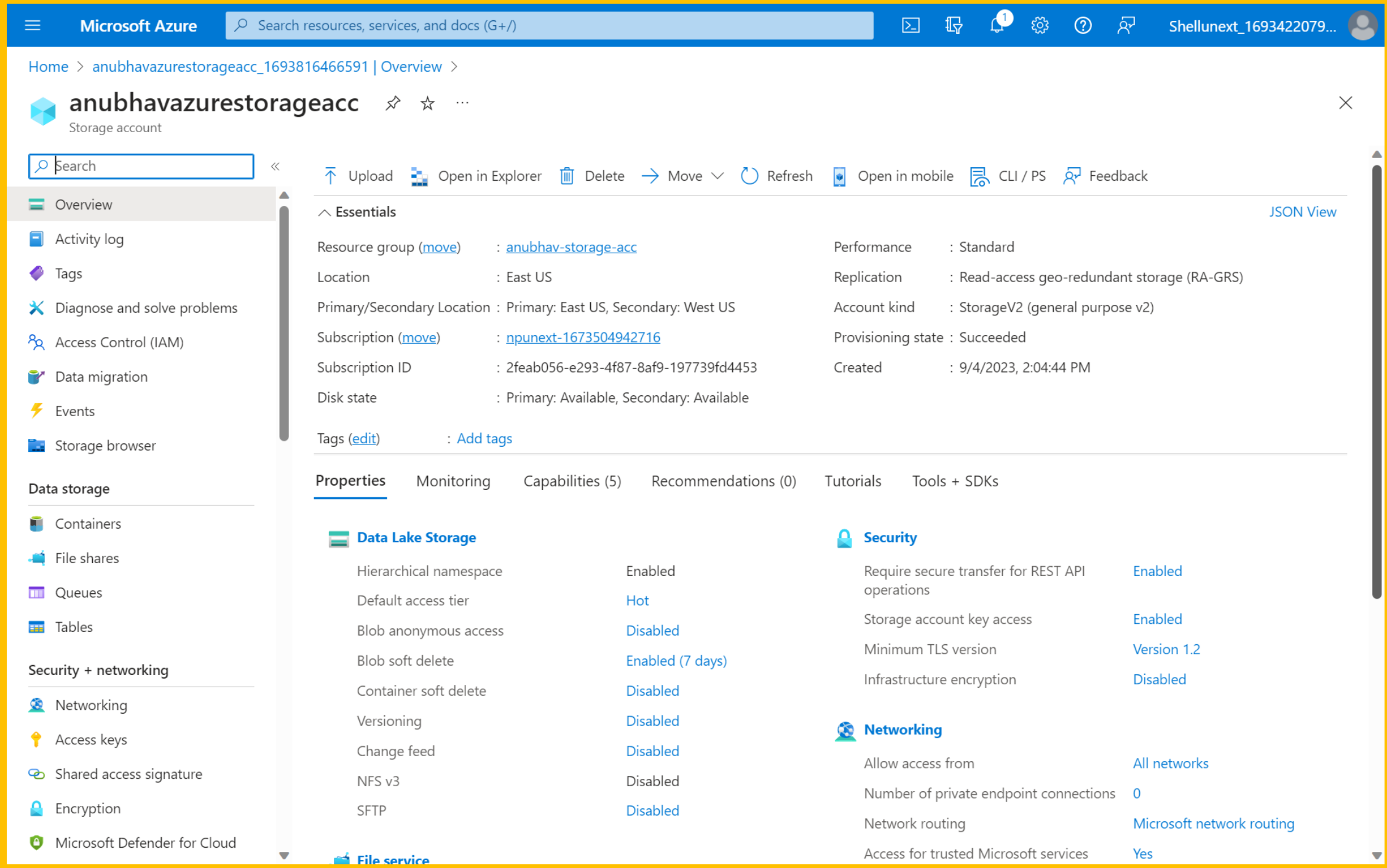
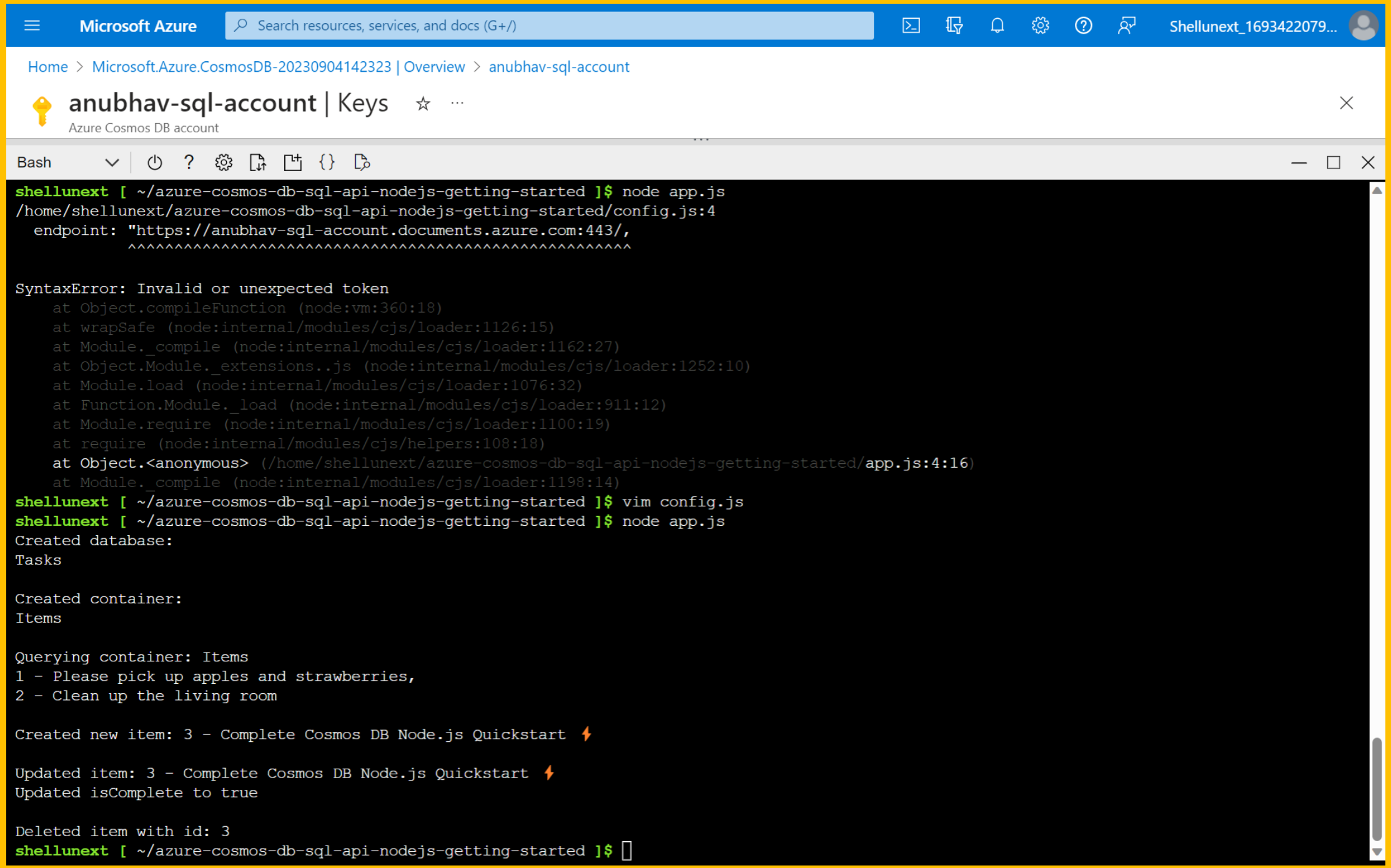


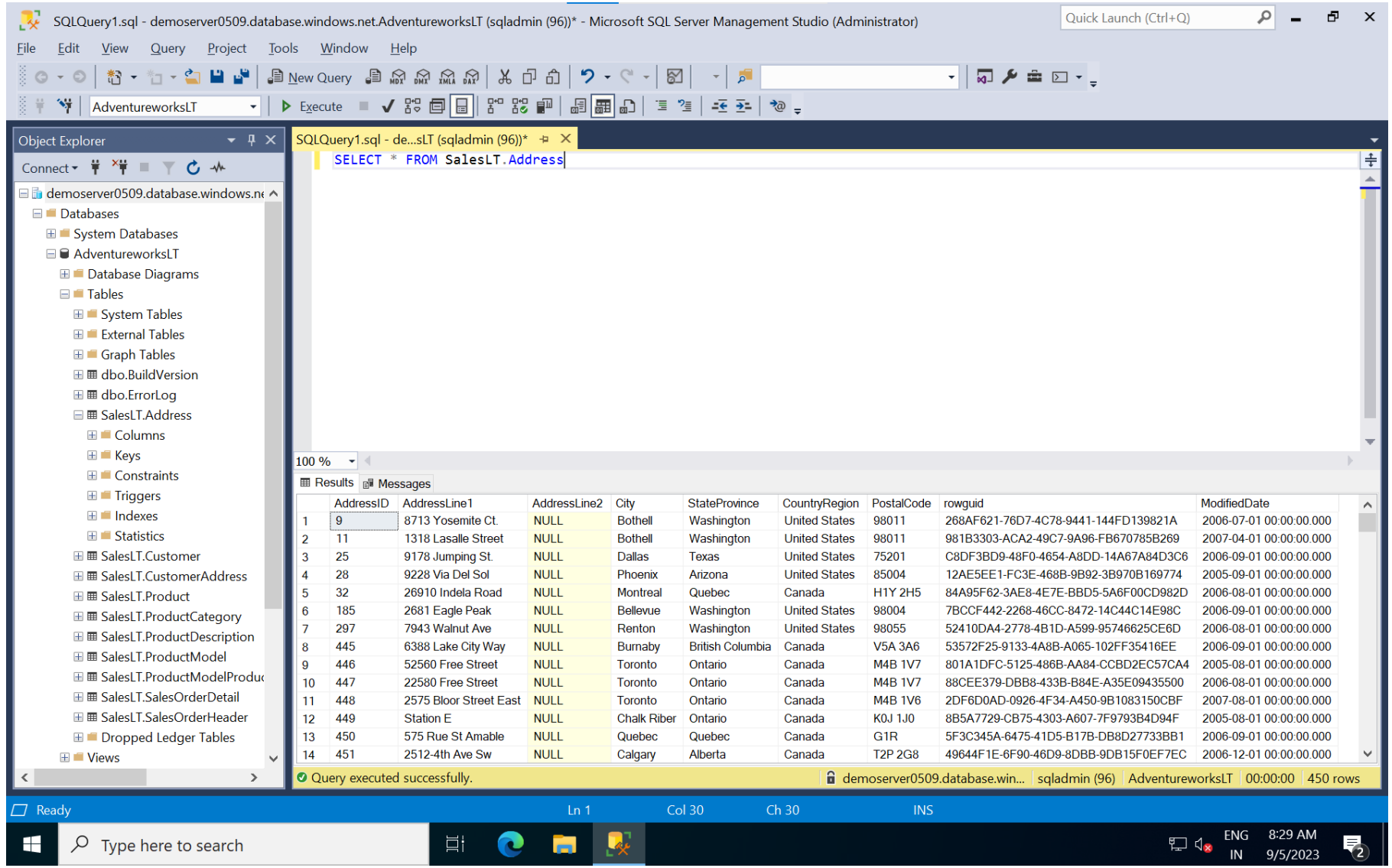
Fig: Compare Azure Blob Storage & Data lake Store Gen 2

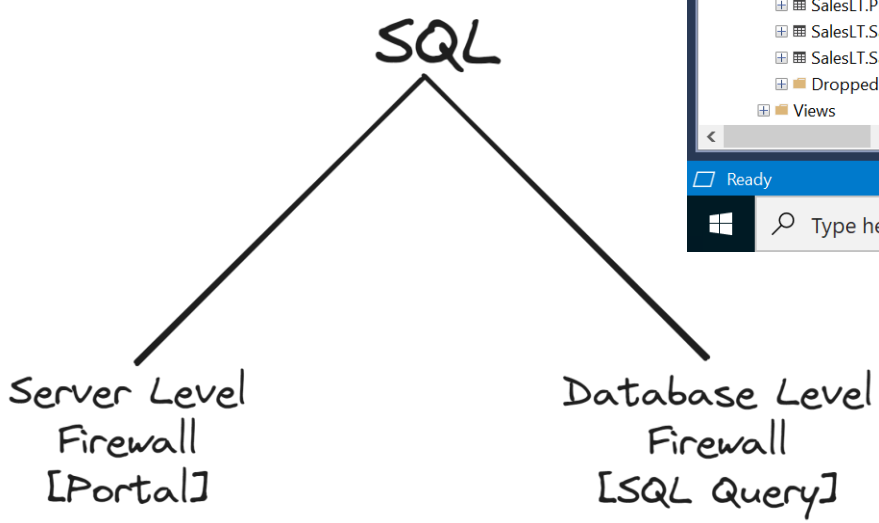
BIG DATA USE CASES

- Modern Data Warehouse
  - ADF is an Orchestrator and controls data movement
  - Azure SQL DW: Azure Synapse, Azure databricks, Azure analysis services
- Advanced data analytics:
  - Cosmos DB, Real time apps
- Real Time data analytics:
  - Apache Kafka for HOInsight to read live streaming signs from sensors & IOT


Microsoft Azure
Search resources, services, and docs (G+/)
Shellunext_1693422079...
Home > anubhavazurestorageacc_1693816466591 | Overview >
anubhavazurestorageacc
Storage account
Search
Overview
Activity log
Tags
Diagnose and solve problems
Access Control (IAM)
Data migration
Events
Storage browser
Data storage
Containers
File shares
Queues
Tables
Security + networking
Networking
Access keys
Shared access signature
Encryption
Microsoft Defender for Cloud
Upload
Open in Explorer
Delete
Move
Refresh
Open in mobile
CLI / PS
Feedback
Essentials
JSON View
Resource group (move) : anubhav-storage-acc
Location : East US
Primary/Secondary Location : Primary: East US, Secondary: West US
Subscription (move) : npunext-1673504942716
Subscription ID : 2feab056-e293-4f87-8af9-197739fd4453
Disk state : Primary: Available, Secondary: Available
Performance : Standard
Replication : Read-access geo-redundant storage (RA-GRS)
Account kind : StorageV2 (general purpose v2)
Provisioning state : Succeeded
Created : 9/4/2023, 2:04:44 PM
Tags (edit) : Add tags
Properties
Monitoring
Capabilities (5)
Recommendations (0)
Tutorials
Tools + SDKs
Data Lake Storage
Hierarchical namespace Enabled
Default access tier Hot
Blob anonymous access Disabled
Blob soft delete Enabled (7 days)
Container soft delete Disabled
Versioning Disabled
Change feed Disabled
NFS v3 Disabled
SFTP Disabled
Security
Require secure transfer for REST API operations Enabled
Storage account key access Enabled
Minimum TLS version Version 1.2
Infrastructure encryption Disabled
Networking
Allow access from All networks
Number of private endpoint connections 0
Network routing Microsoft network routing
Access for trusted Microsoft services Yes
File service

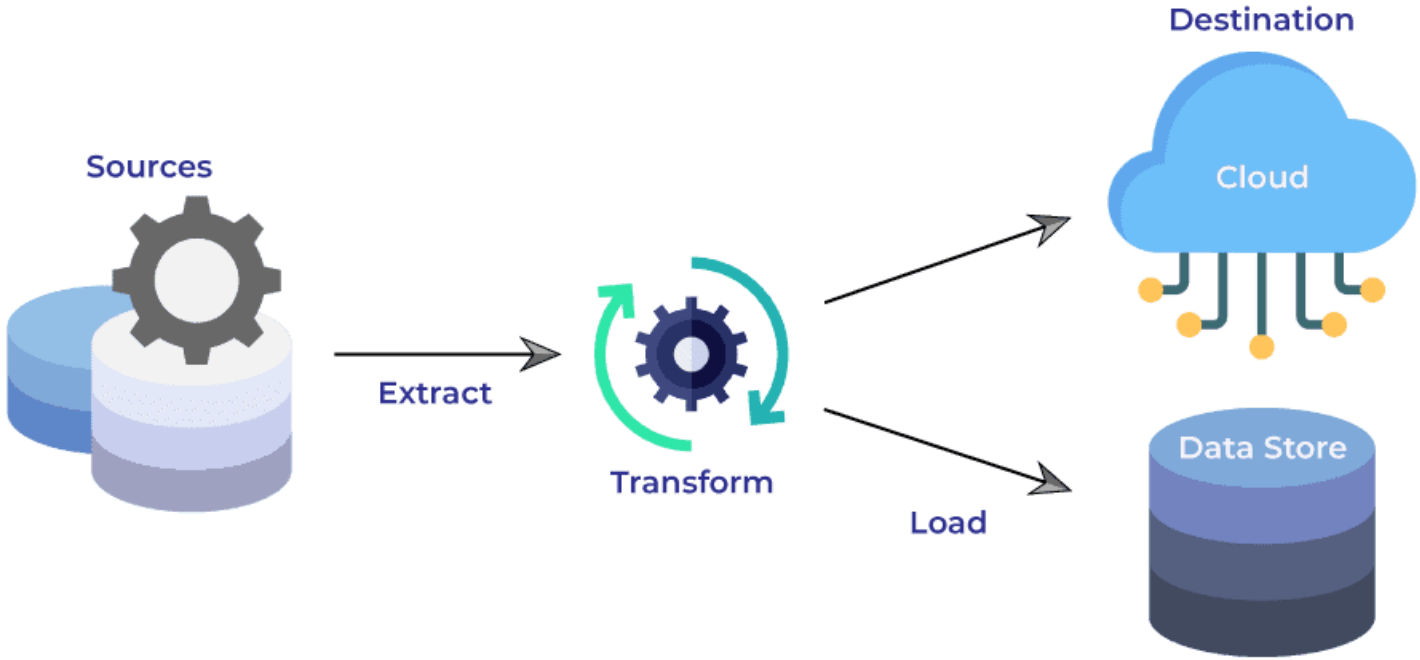

Microsoft Azure
Search resources, services, and docs (G+/)
Shellunext_1693422079...
Home > Microsoft.Azure.CosmosDB-20230904142323 | Overview > anubhav-sql-account
anubhav-sql-account | Keys
Azure Cosmos DB account
Bash
shellunext [~/azure-cosmos-db-sql-api-nodejs-getting-started]$ node app.js
/home/shellunext/azure-cosmos-db-sql-api-nodejs-getting-started/config.js:4
endpoint: "https://anubhav-sql-account.documents.azure.com:443/",
^^
SyntaxError: Invalid or unexpected token
at Object.compileFunction (node:vm:360:18)
at wrapSafe (node:internal/modules/cjs/loader:1126:15)
at Module._compile (node:internal/modules/cjs/loader:1162:27)
at Object.Module._extensions..js (node:internal/modules/cjs/loader:1252:10)
at Module.load (node:internal/modules/cjs/loader:1076:32)
at Function.Module._load (node:internal/modules/cjs/loader:911:12)
at Module.require (node:internal/modules/cjs/loader:1100:19)
at require (node:internal/modules/cjs/helpers:108:18)
at Object.<anonymous> (/home/shellunext/azure-cosmos-db-sql-api-nodejs-getting-started/app.js:4:16)
at Module._compile (node:internal/modules/cjs/loader:1198:14)
shellunext [~/azure-cosmos-db-sql-api-nodejs-getting-started]$ vim config.js
shellunext [~/azure-cosmos-db-sql-api-nodejs-getting-started]$ node app.js
Created database:
Tasks
Created container:
Items
Querying container: Items
1 - Please pick up apples and strawberries,
2 - Clean up the living room
Created new item: 3 - Complete Cosmos DB Node.js Quickstart
Updated item: 3 - Complete Cosmos DB Node.js Quickstart
Updated isComplete to true
Deleted item with id: 3
shellunext [~/azure-cosmos-db-sql-api-nodejs-getting-started]$


SQLQuery1.sql - demoserver0509.database.windows.net.AdventureworksLT (sqladmin (96))* - Microsoft SQL Server Management Studio (Administrator)
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
AdventureworksLT
Object Explorer
Connect
demoserver0509.database.windows.ne
Databases
System Databases
AdventureworksLT
Database Diagrams
Tables
System Tables
External Tables
Graph Tables
dbo.BuildVersion
dbo.ErrorLog
SalesLT.Address
Columns
Keys
Constraints
Triggers
Indexes
Statistics
SalesLT.Customer
SalesLT.CustomerAddress
SalesLT.Product
SalesLT.ProductCategory
SalesLT.ProductDescription
SalesLT.ProductModel
SalesLT.ProductModelProduc
SalesLT.SalesOrderDetail
SalesLT.SalesOrderHeader
Dropped Ledger Tables
Views
SQLQuery1.sql - de...sLT (sqladmin (96))*
SELECT * FROM SalesLT.Address
100 %
Results
Messages
AddressID AddressLine1 AddressLine2 City StateProvince CountryRegion PostalCode rowguid ModifiedDate
1 9 8713 Yosemite Ct. NULL Bothell Washington United States 98011 268AF621-76D7-4C78-9441-144FD139821A 2006-07-01 00:00:00.000
2 11 1318 Lasalle Street NULL Bothell Washington United States 98011 981B3303-ACA2-49C7-9A96-FB670785B269 2007-04-01 00:00:00.000
3 25 9178 Jumping St. NULL Dallas Texas United States 75201 C8DF3BD9-48F0-4654-A8DD-14A67A84D3C6 2006-09-01 00:00:00.000
4 28 9228 Via Del Sol NULL Phoenix Arizona United States 85004 12AE5EE1-FC3E-468B-9B92-3B970B169774 2005-09-01 00:00:00.000
5 32 26910 Indela Road NULL Montreal Quebec Canada H1Y 2H5 84A95F62-3AE8-4E7E-BBD5-5A6F00CD982D 2006-08-01 00:00:00.000
6 185 2681 Eagle Peak NULL Bellevue Washington United States 98004 7BCCF442-2268-46CC-8472-14C44C14E98C 2006-09-01 00:00:00.000
7 297 7943 Walnut Ave NULL Renton Washington United States 98055 52410DA4-2778-4B1D-A599-95746625CE6D 2006-08-01 00:00:00.000
8 445 6388 Lake City Way NULL Burnaby British Columbia Canada V5A 3A6 53572F25-9133-4A8B-A065-102FF35416EE 2006-09-01 00:00:00.000
9 446 52560 Free Street NULL Toronto Ontario Canada M4B 1V7 801A1DFC-5125-486B-AA84-CCBD2EC57CA4 2005-08-01 00:00:00.000
10 447 22580 Free Street NULL Toronto Ontario Canada M4B 1V7 88CEE379-DBB8-433B-B84E-A35E09435500 2006-08-01 00:00:00.000
11 448 2575 Bloor Street East NULL Toronto Ontario Canada M4B 1V6 2DF6D0AD-0926-4F34-A450-9B1083150CBF 2007-08-01 00:00:00.000
12 449 Station E NULL Chalk Riber Ontario Canada K0J 1J0 8B5A7729-CB75-4303-A607-7F9793B4D94F 2005-08-01 00:00:00.000
13 450 575 Rue St Amable NULL Quebec Quebec Canada G1R 5F3C345A-6475-41D5-B17B-DB8D27733BB1 2006-09-01 00:00:00.000
14 451 2512-4th Ave Sw NULL Calgary Alberta Canada T2P 2G8 49644F1E-6F90-46D9-8DBB-9DB15F0EF7EC 2006-12-01 00:00:00.000
Query executed successfully.
demoserver0509.database.win...
sqladmin (96)
AdventureworksLT
00:00:00
450 rows
Ready
Ln 1
Col 30
Ch 30
INS
Type here to search
ENG
IN
8:29 AM
9/5/2023


SQL
Server Level
Firewall
[Portal]
Database Level
Firewall
[SQL Query]

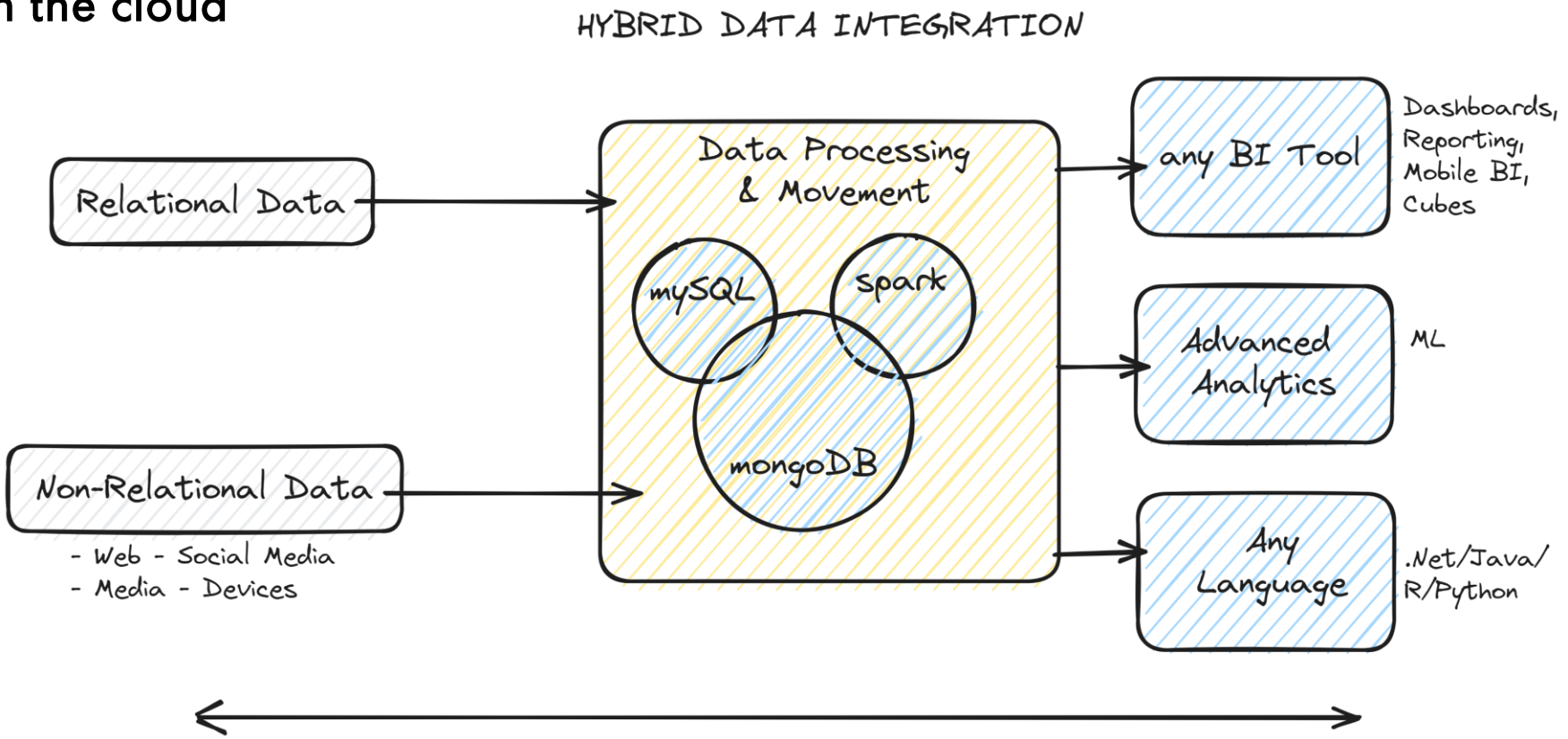
Orchestrating data movement with ADF

1. E – Extract [CSV, Txt, JSON]
2. L – Load [Sink, Destination, Data Warehouse]
3. T – Transformation [Cleaning, de-duplicate, Custom calculations, Convert] – delays



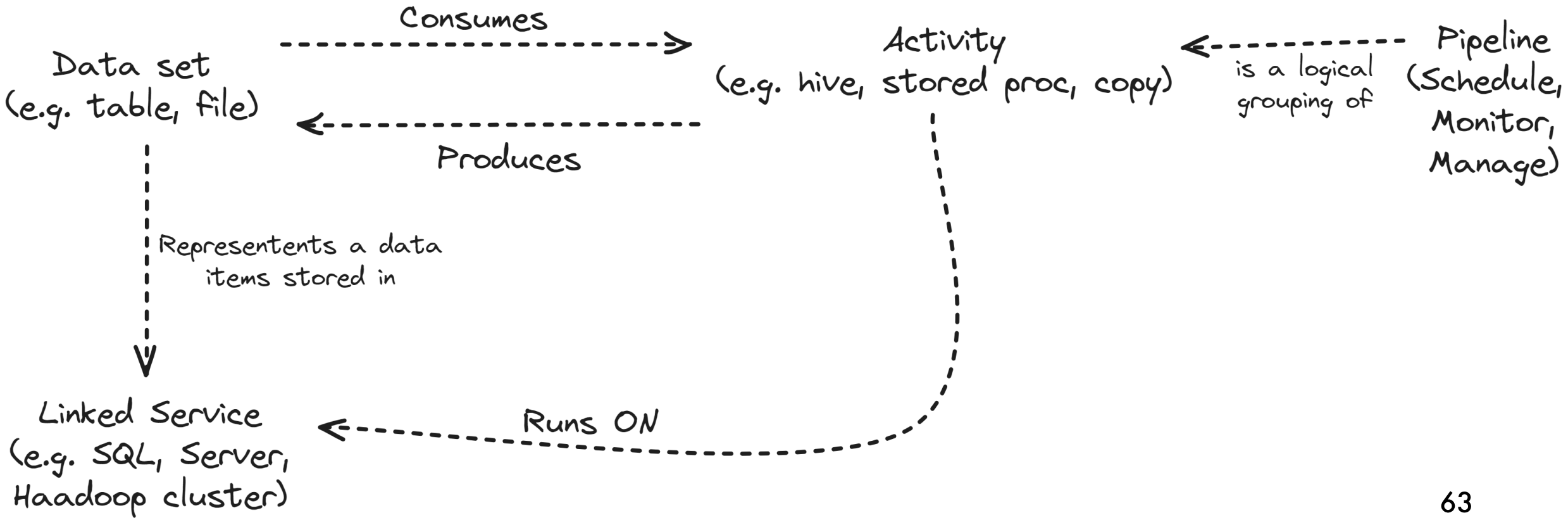
AZURE DATA FACTORY

Creates, orchestrates, and automates the movement, transformation and/or analysis of data through in the cloud



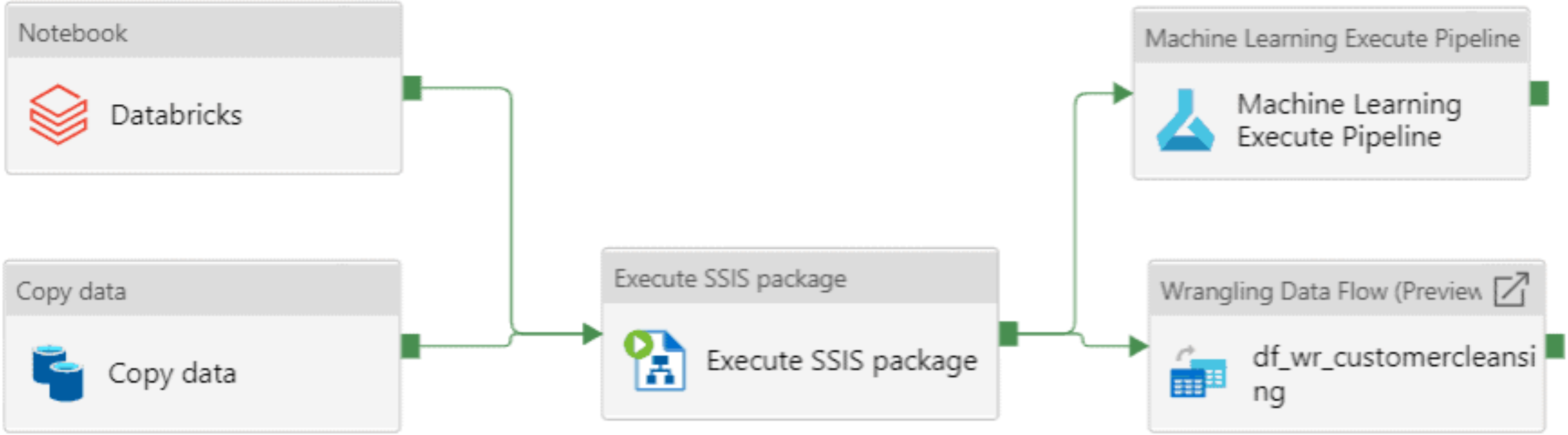
AZURE DATA FACTORY COMPONENT

Azure Data Factory process: Connect & collect -> Transform & enrich -> Publish -> Monitor



ADF Pipelines

- Pipeline is a grouping of logically related activities
- Pipeline can be scheduled so the activities within it get executed
- Pipeline can be managed & monitored



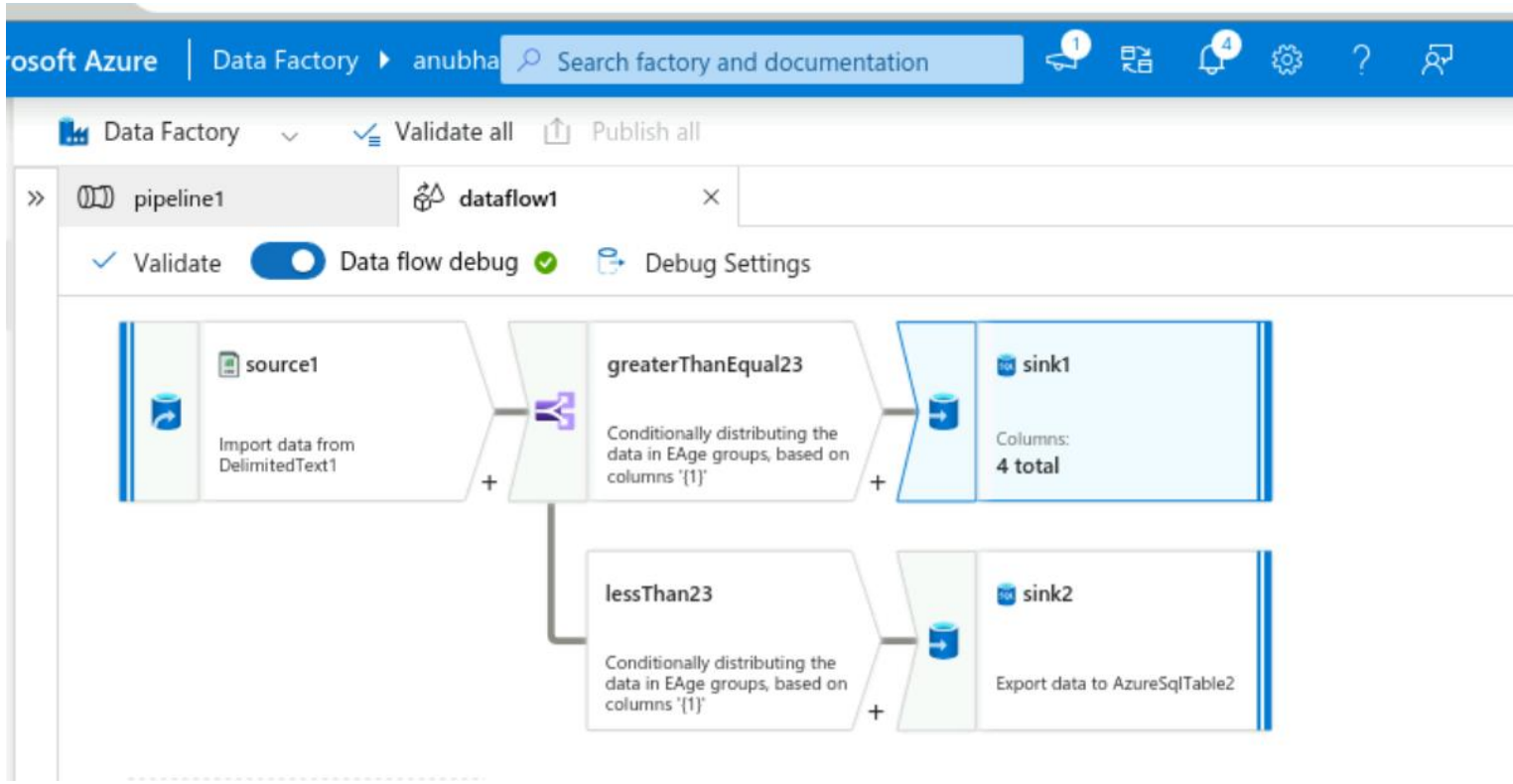
INGESTING & TRANSFORMING DATA

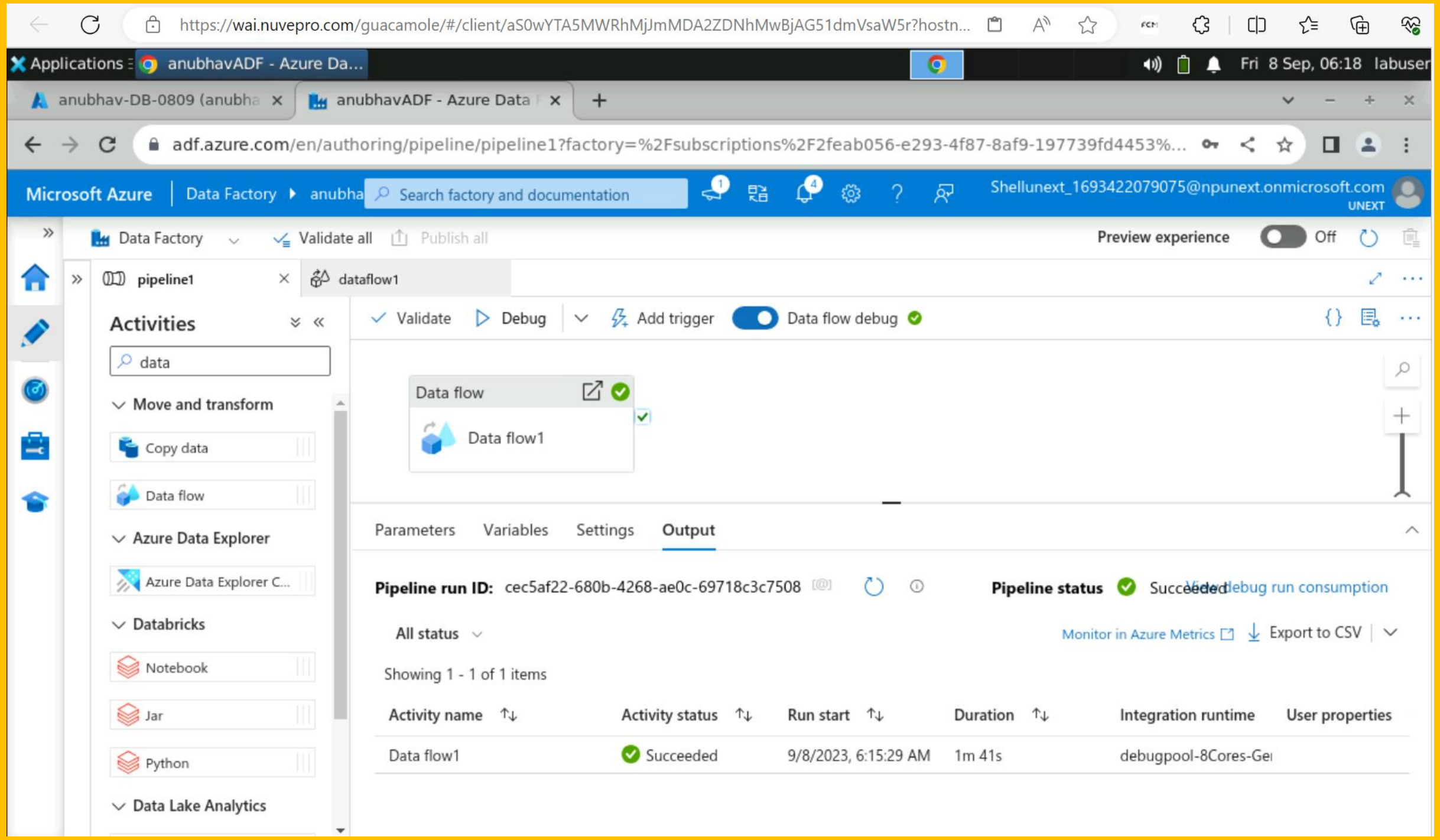
Ingesting data with the copy activity-

- Reads data from a source data store
- Performs serialization/deserialization, compression/decompression, column mapping, and so on. It performs these operations based on the configuration of the input dataset, output dataset, and Copy activity
- Writes data to the sink/destination data store

Data Flow in ADF

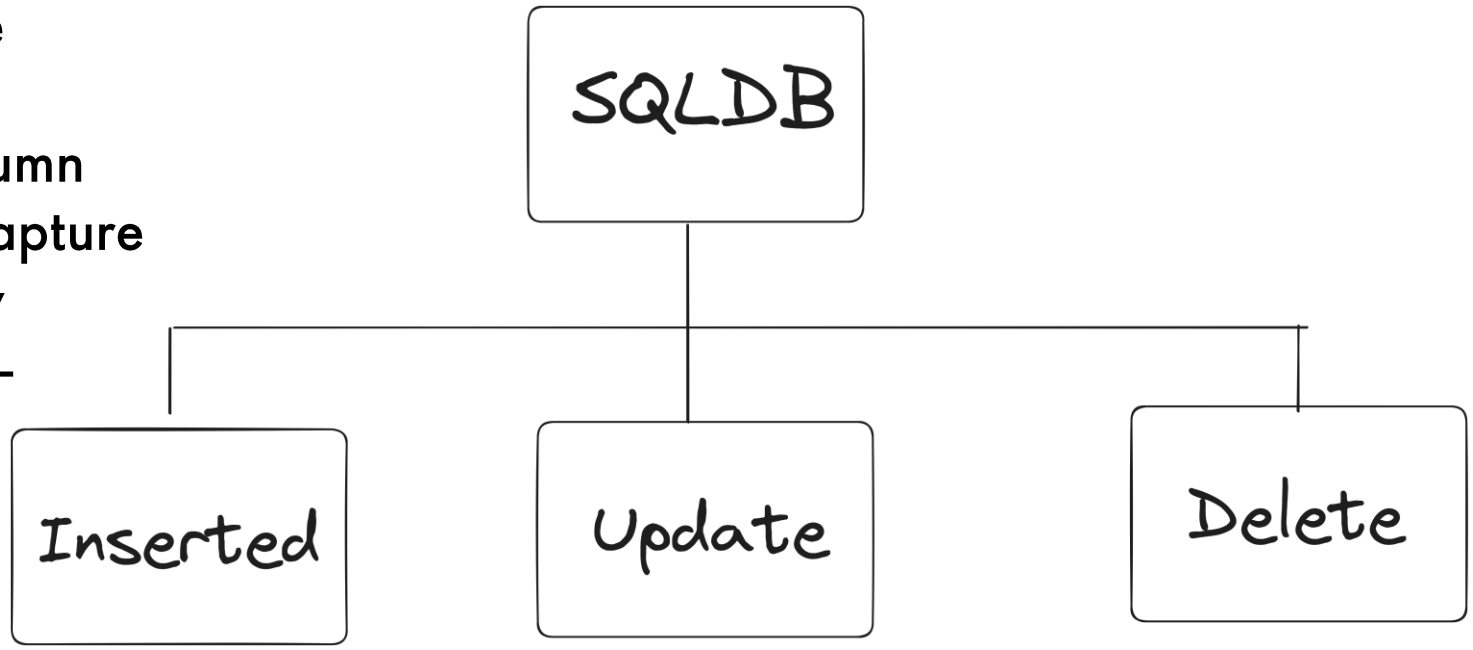
- Source Activity
- Sink Transformation
- Union Transformation
- Surrogate Key transformation
- Conditional Split Transformation
- Derived Column Transformation
- Mapping
- Validate

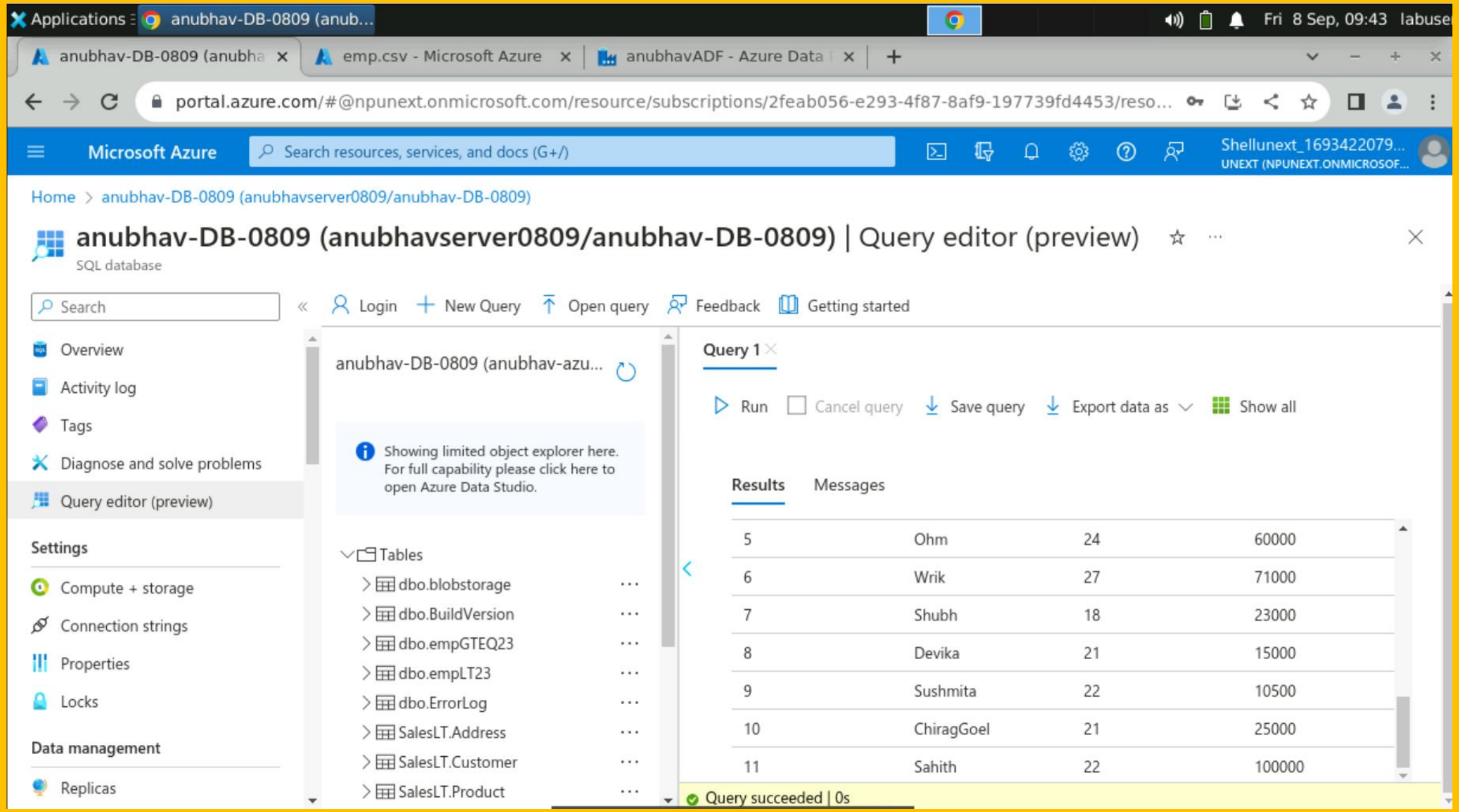



https://wai.nuvepro.com/guacamole/#/client/aS0wYTA5MWRhMjJmMDA2ZDNhMwBjAG51dmVsaW5r?hostn...
Applications
anubhavADF - Azure Da...
Fri 8 Sep, 06:18
labuser
anubhav-DB-0809 (anubha
anubhavADF - Azure Data
adf.azure.com/en/authoring/pipeline/pipeline1?factory=%2Fsubscriptions%2F2feab056-e293-4f87-8af9-197739fd4453%...
Microsoft Azure
Data Factory
anubha
Search factory and documentation
Shellunext_1693422079075@npunext.onmicrosoft.com
UNEXT
Data Factory
Validate all
Publish all
Preview experience
Off
pipeline1
dataflow1
Activities
data
Move and transform
Copy data
Data flow
Azure Data Explorer
Azure Data Explorer C...
Databricks
Notebook
Jar
Python
Data Lake Analytics
Validate
Debug
Add trigger
Data flow debug
Data flow
Data flow1
Parameters
Variables
Settings
Output
Pipeline run ID: cec5af22-680b-4268-ae0c-69718c3c7508
Pipeline status
Succeeded
All status
Monitor in Azure Metrics
Export to CSV
Showing 1 - 1 of 1 items
Activity name
Activity status
Run start
Duration
Integration runtime
User properties
Data flow1
Succeeded
9/8/2023, 6:15:29 AM
1m 41s
debugpool-8Cores-Ge

Extracting Modified Data

- Options for Extracting Modified Data
- Extracting Rows based on a Datetime column
- Demonstration: Using a Datetime column
- Extracting Data with Change Data Capture
- The CDC Control Task and Data Flow Components (Change Data Capture – Incremental update)

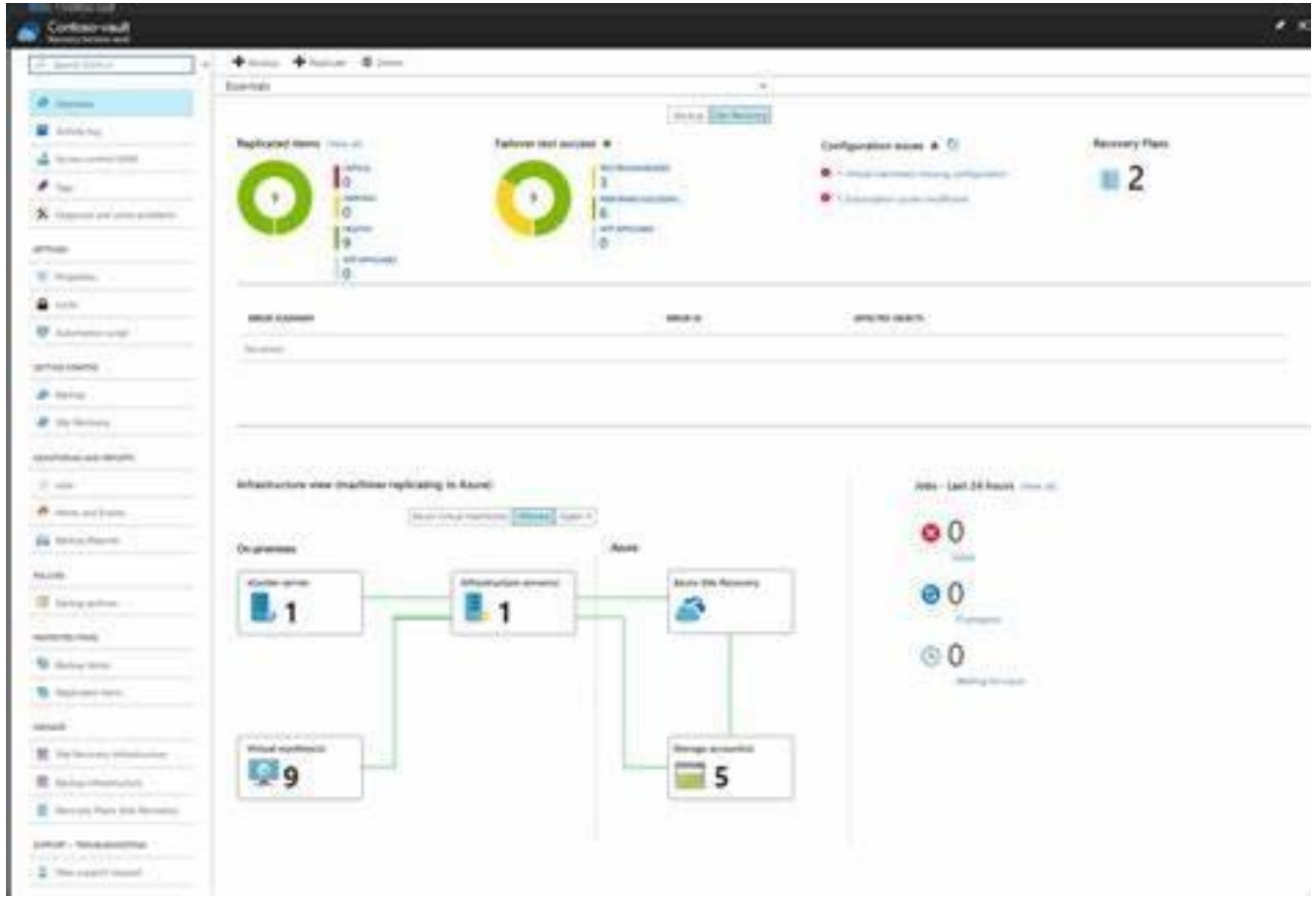



Applications
anubhav-DB-0809 (anub...
Fri 8 Sep, 09:43
anubhav-DB-0809 (anubha
emp.csv - Microsoft Azure
anubhavADF - Azure Data
portal.azure.com/#@npunext.onmicrosoft.com/resource/subscriptions/2feab056-e293-4f87-8af9-197739fd4453/reso...
Microsoft Azure
Search resources, services, and docs (G+/)
Shellunext_1693422079...
UNEXT (NPUNEXT.ONMICROSOF...
Home > anubhav-DB-0809 (anubhavserver0809/anubhav-DB-0809)
anubhav-DB-0809 (anubhavserver0809/anubhav-DB-0809) | Query editor (preview)
SQL database
Search
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Settings
Compute + storage
Connection strings
Properties
Locks
Data management
Replicas
Login
New Query
Open query
Feedback
Getting started
anubhav-DB-0809 (anubhav-azu...
Showing limited object explorer here. For full capability please click here to open Azure Data Studio.
Tables
dbo.blobstorage
dbo.BuildVersion
dbo.empGTEQ23
dbo.empLT23
dbo.ErrorLog
SalesLT.Address
SalesLT.Customer
SalesLT.Product
Query 1
Run
Cancel query
Save query
Export data as
Show all
Results
Messages
5 Ohm 24 60000
6 Wrik 27 71000
7 Shubh 18 23000
8 Devika 21 15000
9 Sushmita 22 10500
10 ChiragGoel 21 25000
11 Sahith 22 100000
Query succeeded | 0s

Monitoring & Troubleshooting

General Azure Monitoring Capabilities

- Azure Monitor
- Monitoring the network
- Diagnose & solve problems



AZURE MONITOR

Provides a holistic monitoring approach by collecting, analyzing, and acting on telemetry from both cloud & on-premises environments

- Metric Data: Provides quantifiable information about a system over time that enables you to observe the behavior of the system
- Log data: Logs can be queries & even analyzed using Azure Monitor logs. In addition, this information is typically presented in the overview page of an azure Resource in the Azure portal
- Alerts: Alerts notify you of critical conditions and potentially take corrective automated actions based on triggers from metrics or logs

MONITORING THE NETWORK

Has the capability to monitor & measure network activity

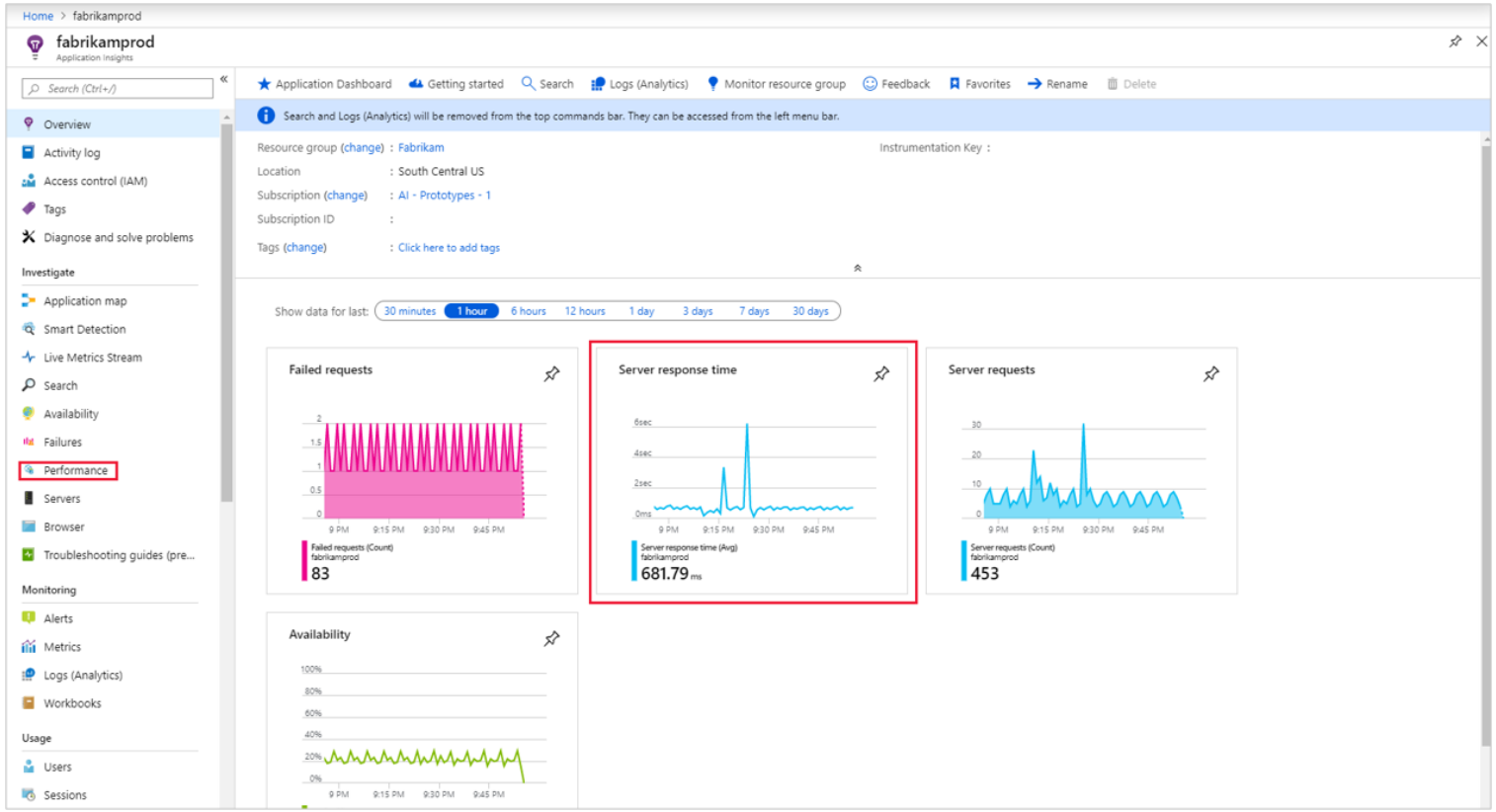
- Network performance monitor: Measure the performance & reachability of the networks that you have configured
- Application Gateway Analytics: Contains rich, out-of-the-box views you can get insights into;

TROUBLESHOOTING ISSUES

- Unable to connect to data platform
  - Check firewall config
  - Test connection by accessing it from a location external to your network
  - Check maintenance schedules
- Authentication failures
  - First check is to ensure that the username and password is correct
  - Check the storage account keys and ensure that they match in the connection string
- Cosmos DB Mongo DB API errors
  - Mongo client drivers establishes more than one connection
  - On the server side, connections which are idle for more than 30 minutes are automatically closed down
  - Check for timeouts

Performance Issues

- Data Lake storage
- Cosmos DB
- SQL data warehouse
- SQL Database
- Colocation of resources



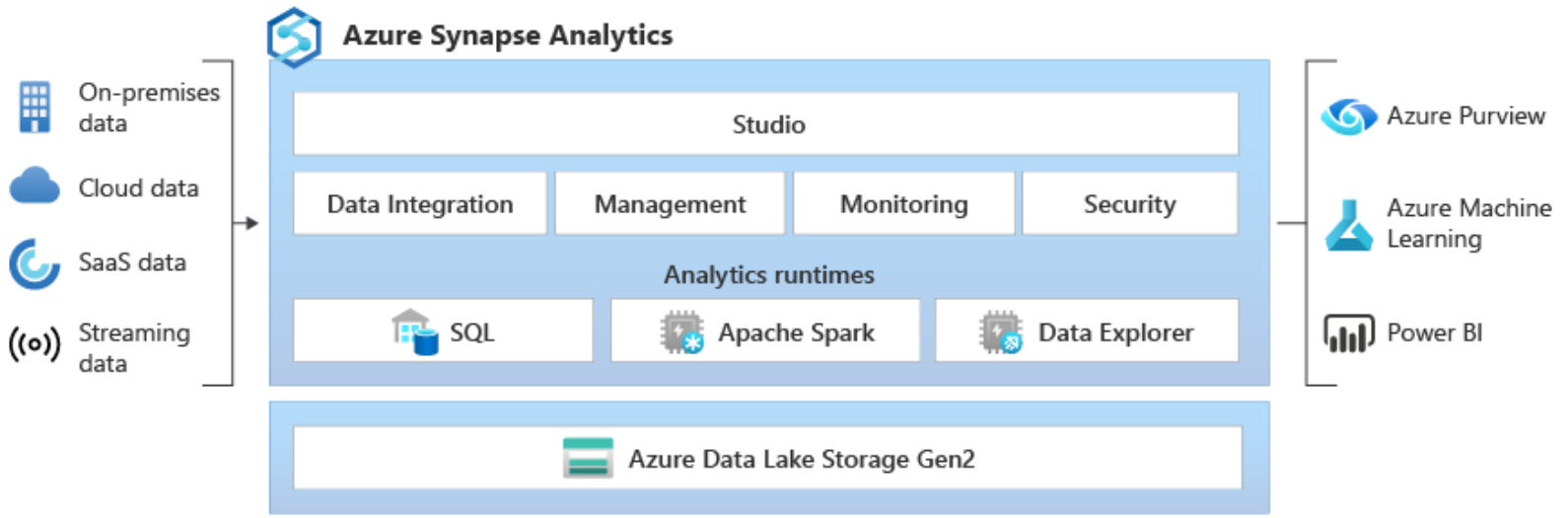
Week 4 Custom IDA Training

12 Sept 2023 – 15 Sept 2023

- Azure Synapse Analytics
- Data Visualization using Power BI
- Python

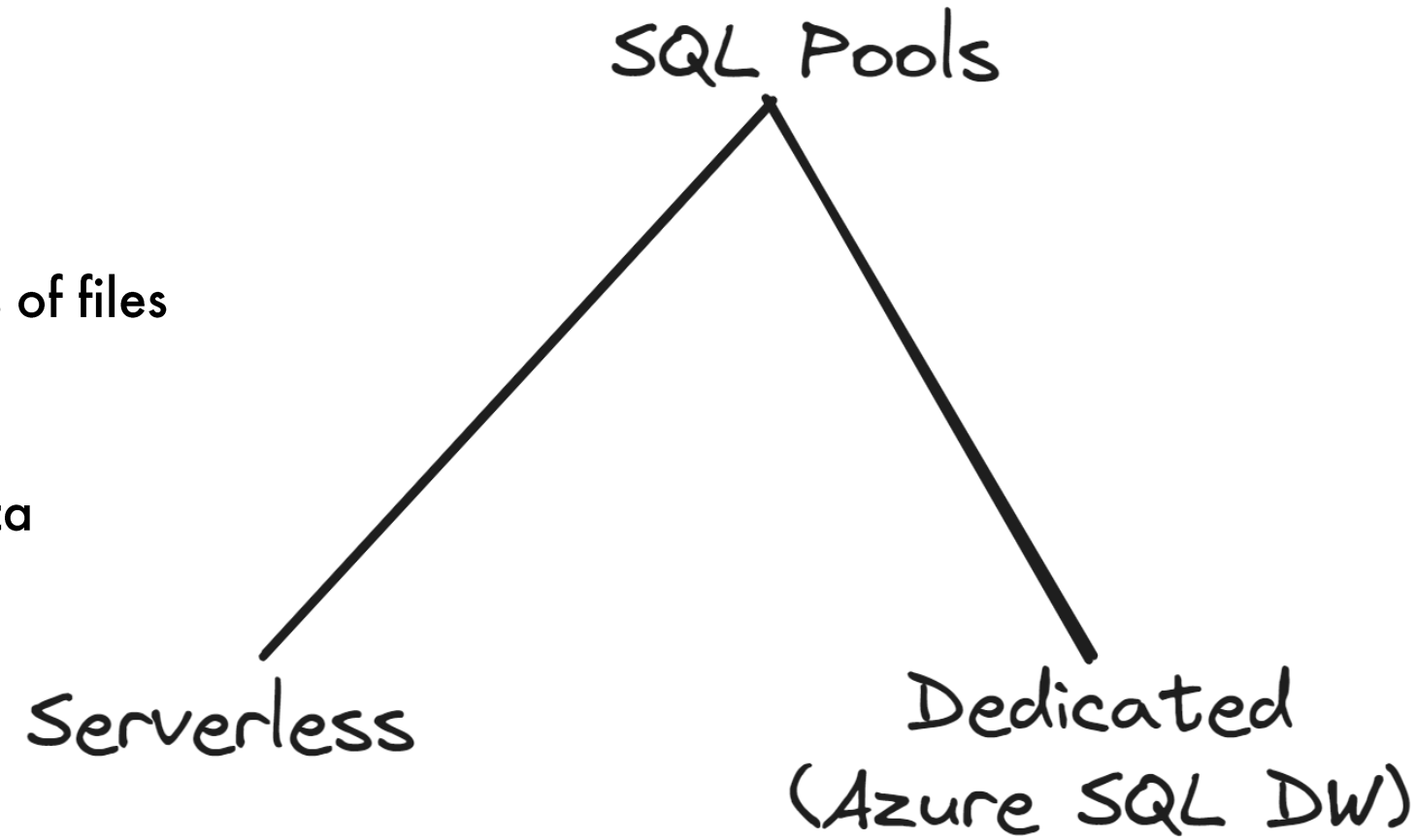
AZURE SYNAPSE ANALYTICS

- Large-scale data warehousing
- Advanced analytics
- Data exploration & discovery
- Real time analytics
- Data integration
- Integrated Analytics



SQL POOLS

- Serverless
  - Data exploration and analysis of files in data lake
- Dedicated
  - Host large scale relational data warehouses



SYNAPSE COMPONENTS

- Synapse Spark
  - Requires synapse pool which runs in background
- Synapse Studio
  - IDE for all the services like Synapse Pipeline, Synapse SQL, Synapse Link, etc.
- Synapse Pipeline
  - They are limited to the synapse studio (it is similar to ADF)
- Synapse Link
  - Creates a communication to Azure Cosmos database
- Compute Service
  - Virtual machines create them automatically when working with Sparks

BUSINESS INTELLIGENCE

- SQL Server Integration Service (SSIS) – Extract, transform, load data from the source
- SQL Server Analysis Service (SSAS) – Facts, Dimensions
- SQL Server Reporting Services (SSRS) – Mobile Reports, Desktop Reports
- Self Service BI
  - Power pivot (Data model)
  - Power Query (query in)
  - Powerview * (Visualization)
  - Power Map (Spatial data)

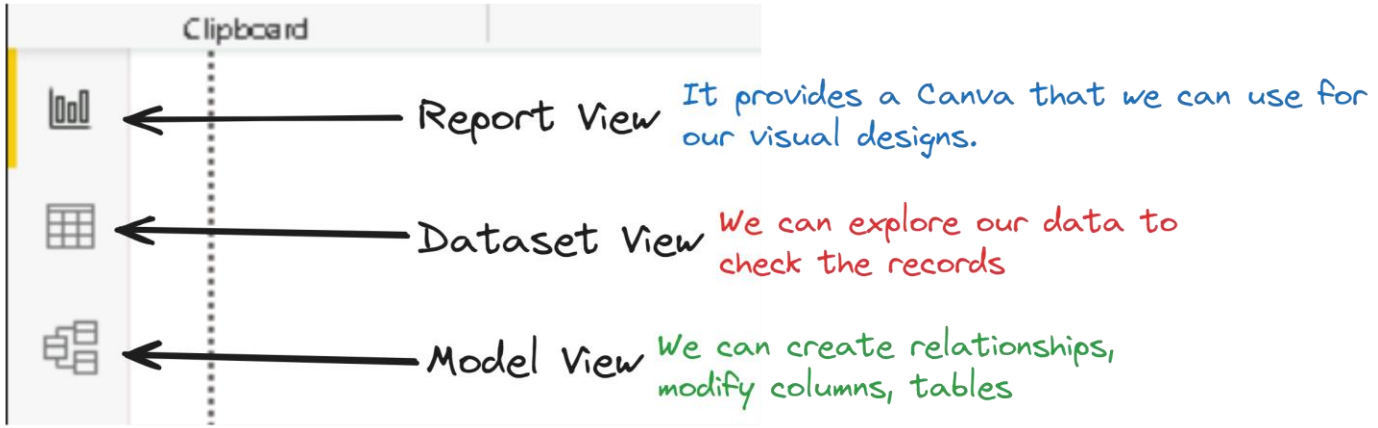
DATA ANALYSIS

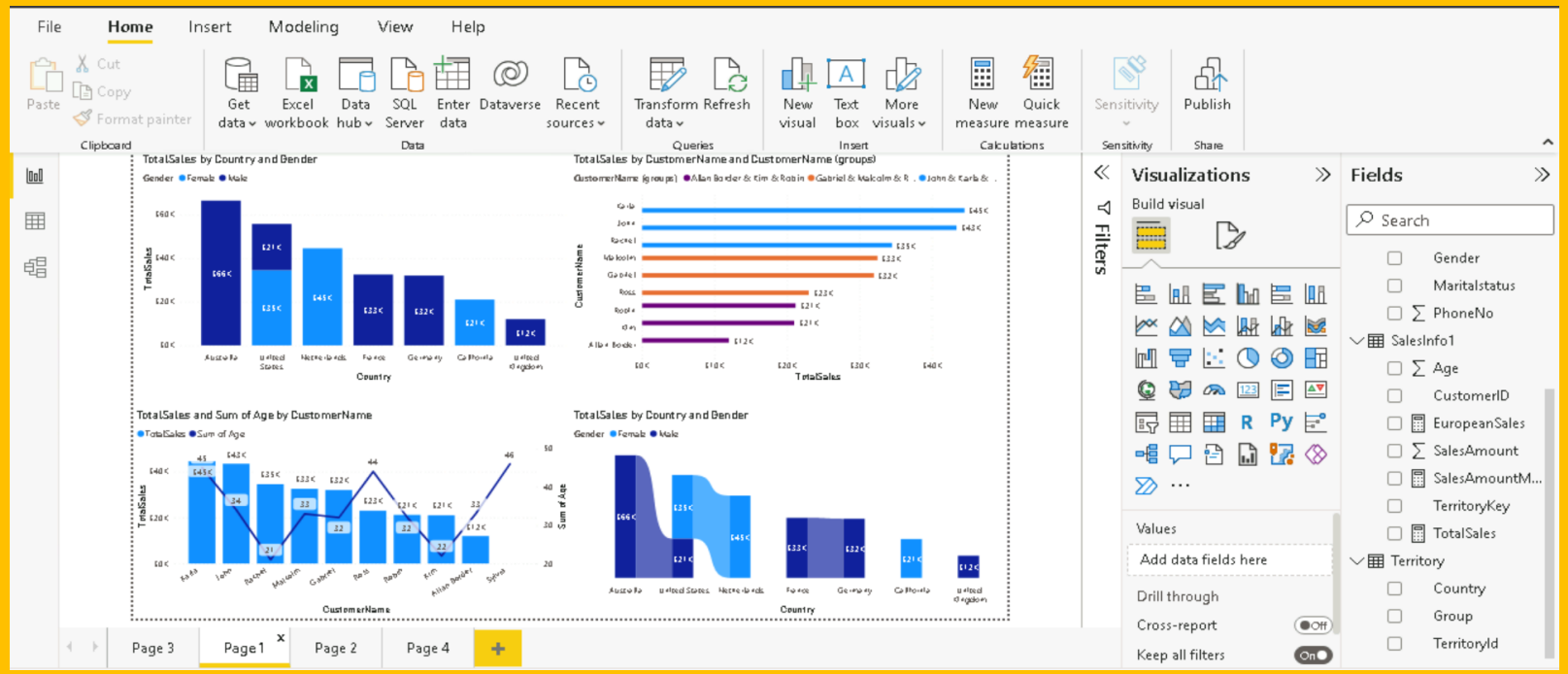
Data Analysis is telling a story with data

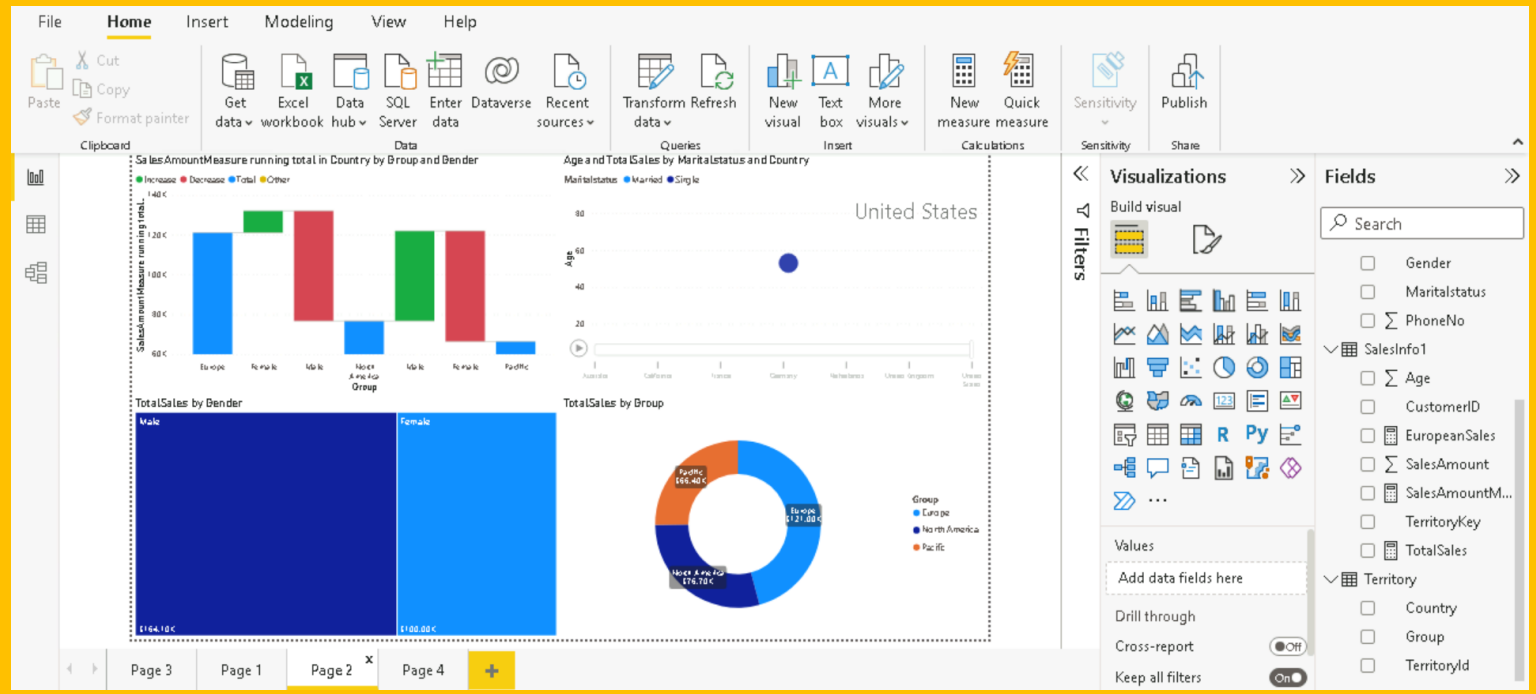
- Descriptive
- Diagnostic
- Predictive
- Prescriptive
- Cognitive

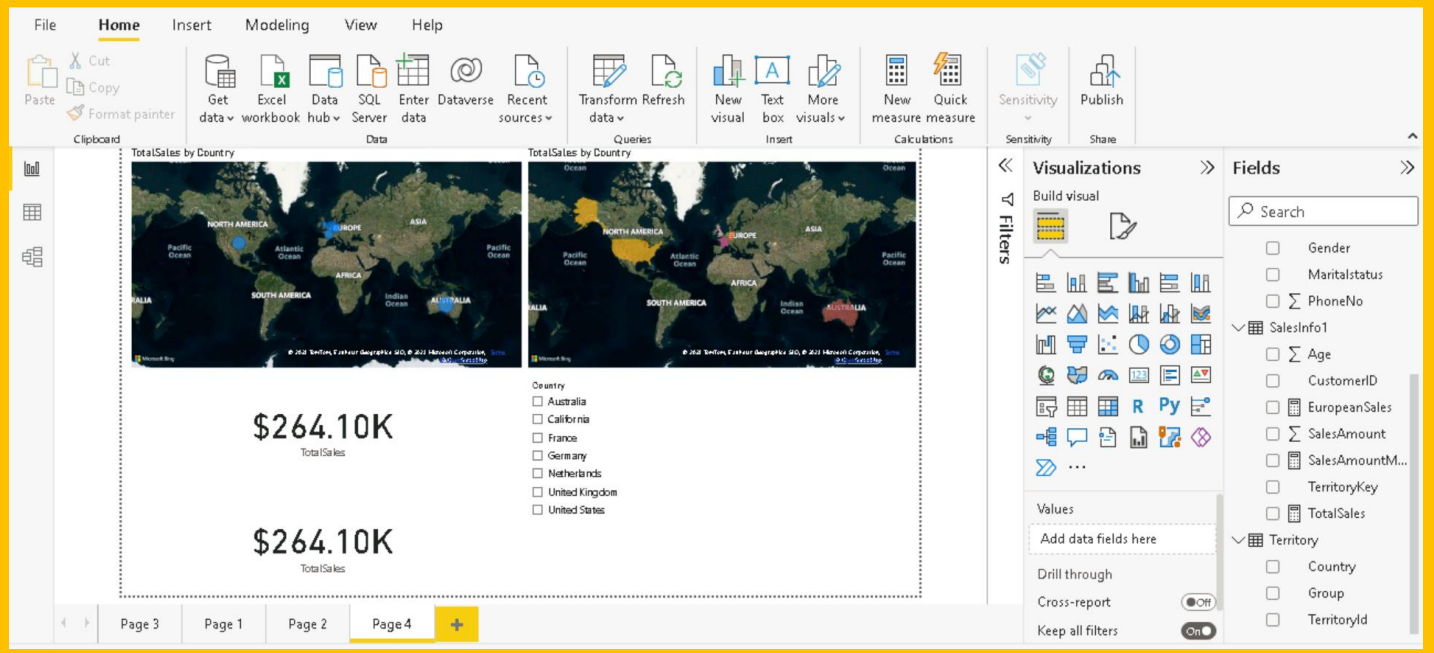
POWER BI

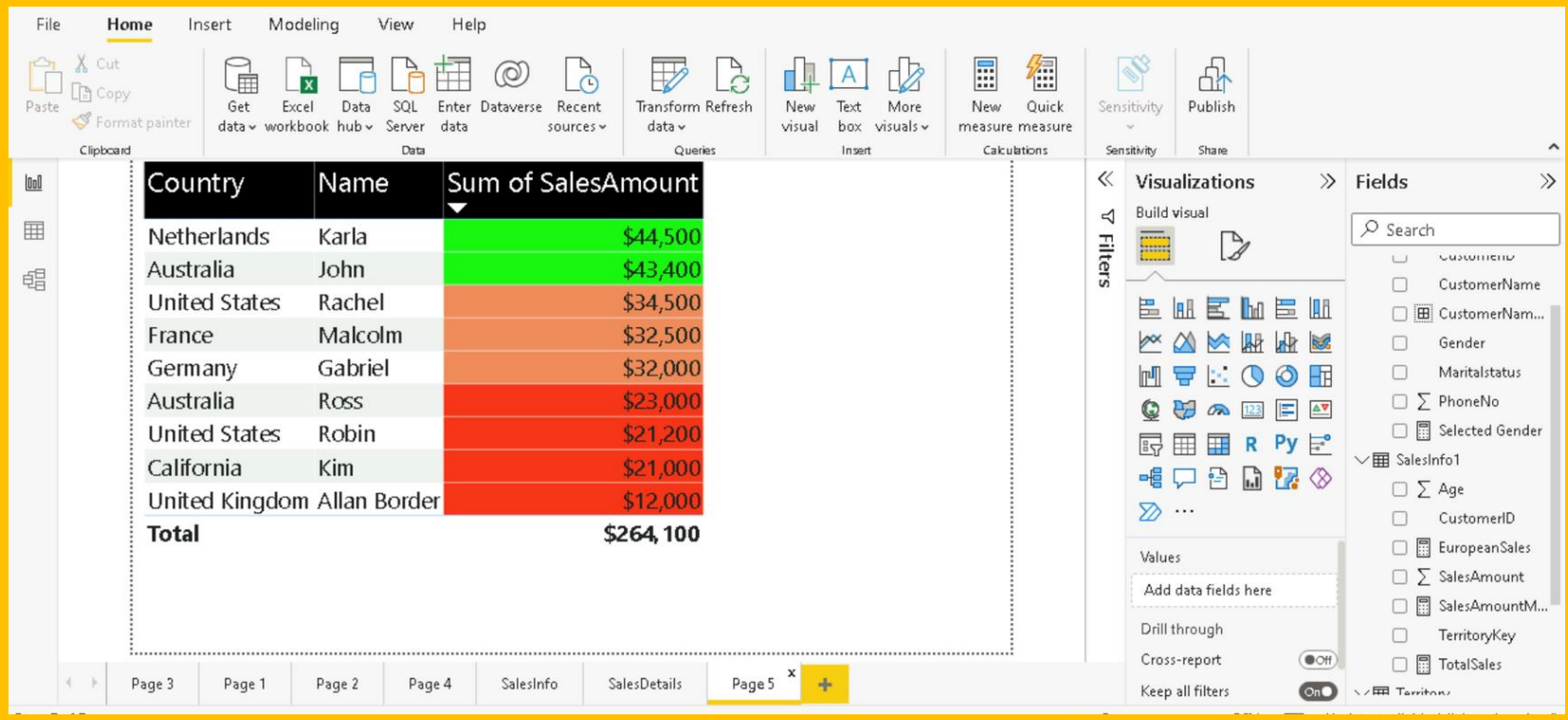
- **Load Data:** Create a dataset - Tables
- **Transform Data:** Navigate to Power Query Editor - Shaping the data (clean; de-duplication)
- **Import data (Local data set):** creating local copy - Dataset - stores the information
- **Direct Query (Data source):** Create Schemas - Connect directly to my Data Source - compare schemas in Power BI with Data Source schema

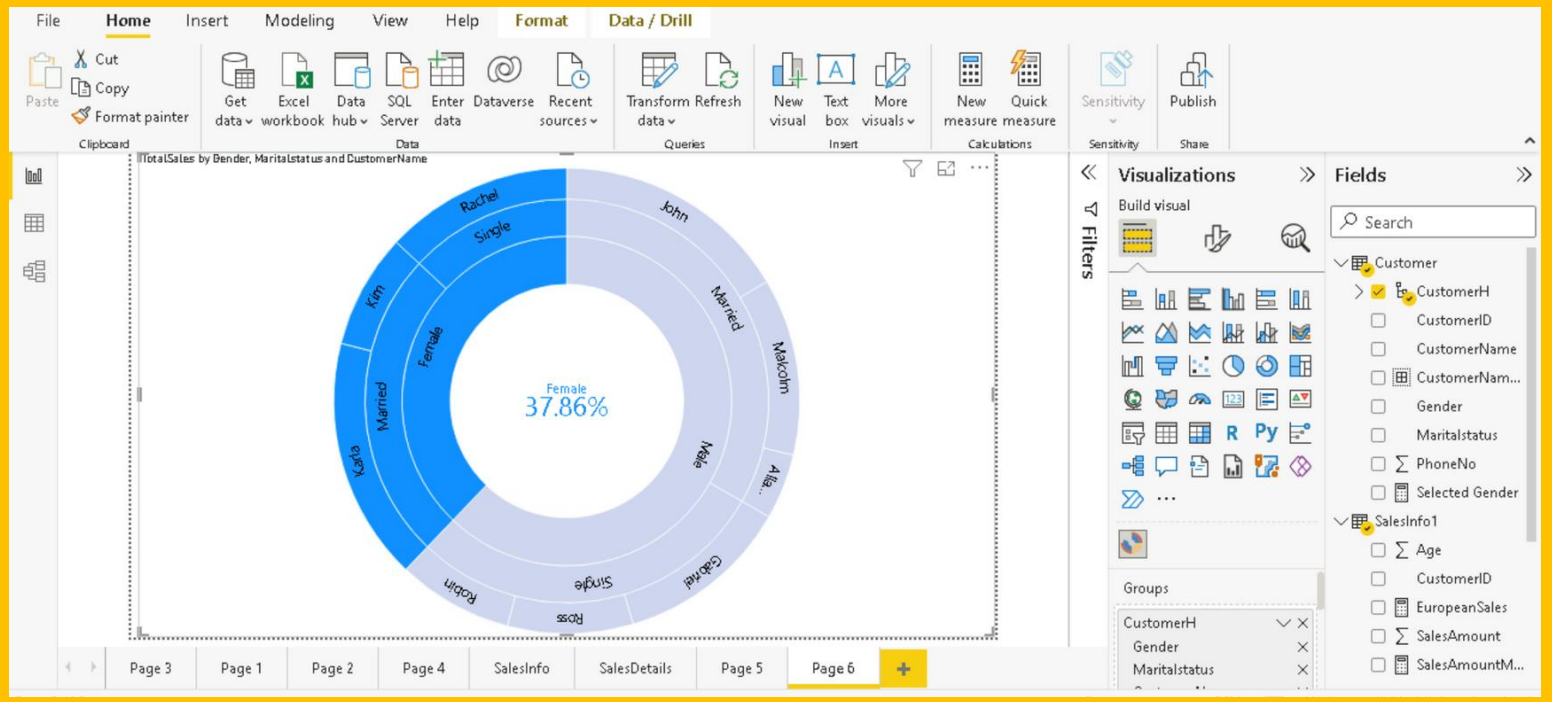


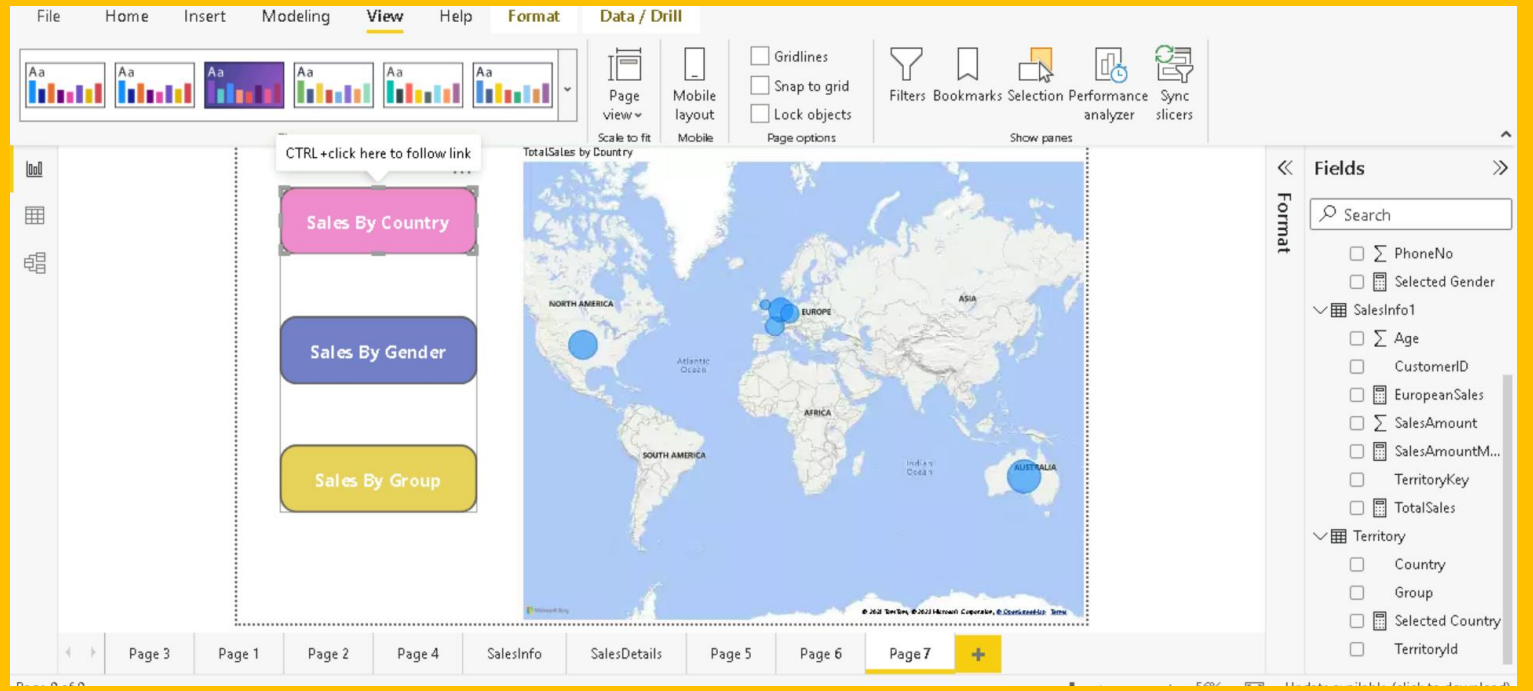

File
Home
Insert
Modeling
View
Help
Paste
Cut
Copy
Format painter
Clipboard
Get data
Excel workbook
Data hub
SQL Server
Enter data
Dataverse
Recent sources
Data
Transform data
Refresh
Queries
New visual
Text box
More visuals
Insert
New measure
Quick measure
Calculations
Sensitivity
Publish
Share
TotalSales by Country and Bender
Gender Female Male
TotalSales
Country
TotalSales by CustomerName and CustomerName (groups)
CustomerName (groups)
CustomerName
TotalSales
TotalSales and Sum of Age by CustomerName
TotalSales Sum of Age
CustomerName
TotalSales by Country and Bender
Gender Female Male
Country
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Off
Keep all filters
On
Fields
Search
Gender
Maritalstatus
PhoneNo
SalesInfo1
Age
CustomerID
EuropeanSales
SalesAmount
SalesAmountM...
TerritoryKey
TotalSales
Territory
Country
Group
TerritoryId
Filters
Page 3
Page 1
Page 2
Page 4


File
Home
Insert
Modeling
View
Help
Paste
Cut
Copy
Format painter
Clipboard
Get data
Excel workbook
Data hub
SQL Server
Enter data
Dataverse
Recent sources
Data
Transform data
Refresh
Queries
New visual
Text box
More visuals
Insert
New measure
Quick measure
Calculations
Sensitivity
Publish
Share
SalesAmountMeasure running total in Country by Group and Bender
Increase Decrease Total Other
SalesAmountMeasure running total
Group
Age and TotalSales by Maritalstatus and Country
Maritalstatus Married Single
United States
Age
TotalSales by Bender
Male
Female
TotalSales by Group
Group
Europe
North America
Pacific
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Off
Keep all filters
On
Fields
Search
Gender
Maritalstatus
PhoneNo
SalesInfo1
Age
CustomerID
EuropeanSales
SalesAmount
SalesAmountM...
TerritoryKey
TotalSales
Territory
Country
Group
TerritoryId
Filters
Page 3
Page 1
Page 2
Page 4


File
Home
Insert
Modeling
View
Help
Paste
Cut
Copy
Format painter
Clipboard
Get data
Excel workbook
Data hub
SQL Server
Enter data
Dataverse
Recent sources
Data
Transform data
Refresh
Queries
New visual
Text box
More visuals
Insert
New measure
Quick measure
Calculations
Sensitivity
Publish
Share
TotalSales by Country
TotalSales by Country
$264.10K
TotalSales
$264.10K
TotalSales
Country
Australia
California
France
Germany
Netherlands
United Kingdom
United States
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Fields
Search
Gender
Maritalstatus
PhoneNo
SalesInfo1
Age
CustomerID
EuropeanSales
SalesAmount
SalesAmountM...
TerritoryKey
TotalSales
Territory
Country
Group
TerritoryId
Filters
Page 3
Page 1
Page 2
Page 4


File
Home
Insert
Modeling
View
Help
Paste
Cut
Copy
Format painter
Clipboard
Get data
Excel workbook
Data hub
SQL Server
Enter data
Dataverse
Recent sources
Data
Transform data
Refresh
Queries
New visual
Text box
More visuals
Insert
New measure
Quick measure
Calculations
Sensitivity
Publish
Share
Country
Name
Sum of SalesAmount
Netherlands Karla $44,500
Australia John $43,400
United States Rachel $34,500
France Malcolm $32,500
Germany Gabriel $32,000
Australia Ross $23,000
United States Robin $21,200
California Kim $21,000
United Kingdom Allan Border $12,000
Total $264,100
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Fields
Search
CustomerName
CustomerNam...
Gender
Maritalstatus
PhoneNo
Selected Gender
SalesInfo1
Age
CustomerID
EuropeanSales
SalesAmount
SalesAmountM...
TerritoryKey
TotalSales
Filters
Page 3
Page 1
Page 2
Page 4
SalesInfo
SalesDetails
Page 5

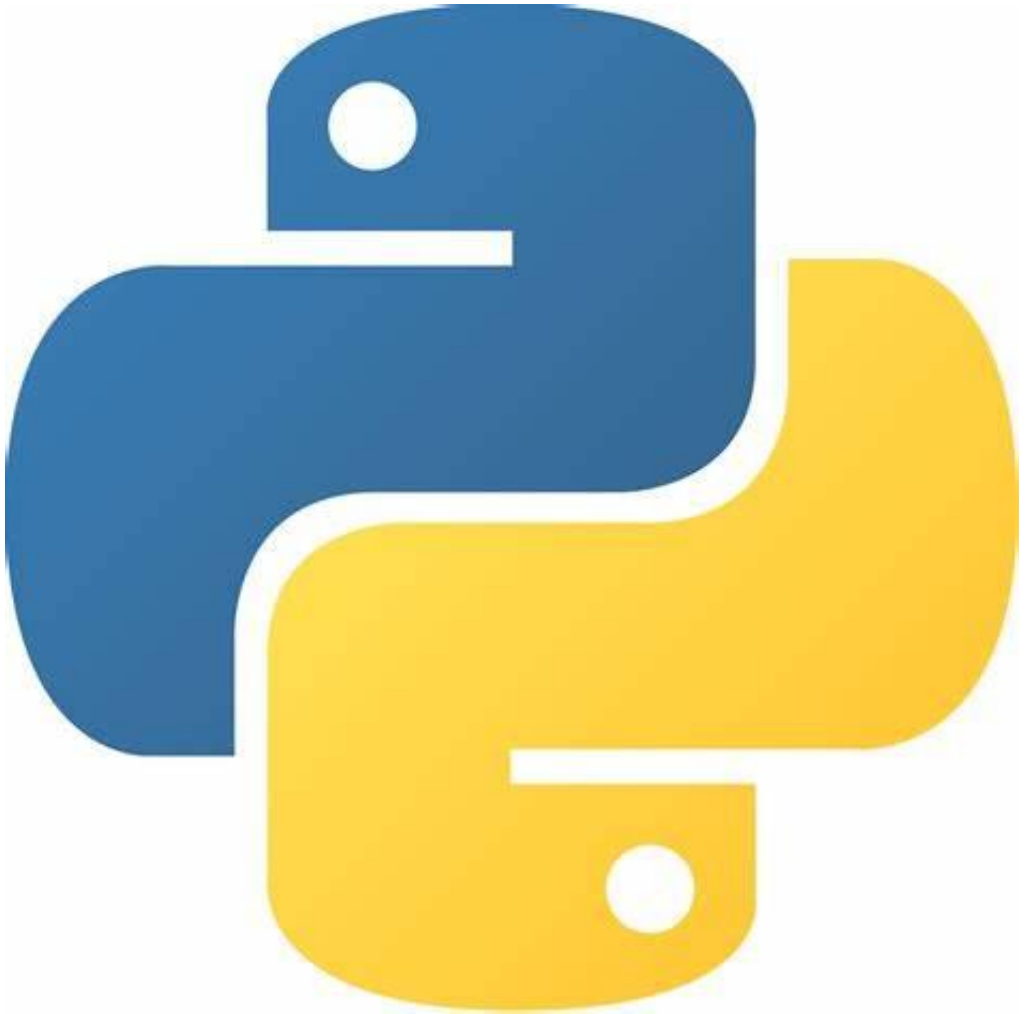

File
Home
Insert
Modeling
View
Help
Format
Data / Drill
Cut
Copy
Format painter
Paste
Clipboard
Get data
Excel workbook
Data hub
SQL Server
Enter data
Dataverse
Recent sources
Data
Transform data
Refresh
Queries
New visual
Text box
More visuals
Insert
New measure
Quick measure
Calculations
Sensitivity
Publish
Share
TotalSales by Gender, Maritalstatus and CustomerName
Rachel
Single
Kim
Female
Married
Karla
John
Married
Malcolm
Male
Allis...
Gabriel
Single
Robin
Ross
Female
37.86%
Visualizations
Build visual
Filters
Fields
Search
Customer
CustomerH
CustomerID
CustomerName
CustomerNam...
Gender
Maritalstatus
PhoneNo
Selected Gender
SalesInfo1
Age
CustomerID
EuropeanSales
SalesAmount
SalesAmountM...
Groups
CustomerH
Gender
Maritalstatus
Page 3
Page 1
Page 2
Page 4
SalesInfo
SalesDetails
Page 5
Page 6


File
Home
Insert
Modeling
View
Help
Format
Data / Drill
Page view
Scale to fit
Mobile layout
Mobile
Gridlines
Snap to grid
Lock objects
Page options
Filters
Bookmarks
Selection
Performance analyzer
Sync slicers
Show panes
CTRL +click here to follow link
Sales By Country
Sales By Gender
Sales By Group
TotalSales by Country
NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
AUSTRALIA
Fields
Search
PhoneNo
Selected Gender
SalesInfo1
Age
CustomerID
EuropeanSales
SalesAmount
SalesAmountM...
TerritoryKey
TotalSales
Territory
Country
Group
Selected Country
TerritoryId
Format
Page 3
Page 1
Page 2
Page 4
SalesInfo
SalesDetails
Page 5
Page 6
Page 7

PYTHON

Usage of python programming

- Web development (Server side)
- Software development
- Mathematics
- System Scripting



WHAT CAN PYTHON DO?

- Python can be used on a server to create web applications
- Python can be used alongside software to create workflows
- Python can connect to database systems. It can also read and modify files
- Python can be used to handle big data & perform complex mathematics
- Python can be used for rapid prototyping, or for production-ready software development

WHY PYTHON?

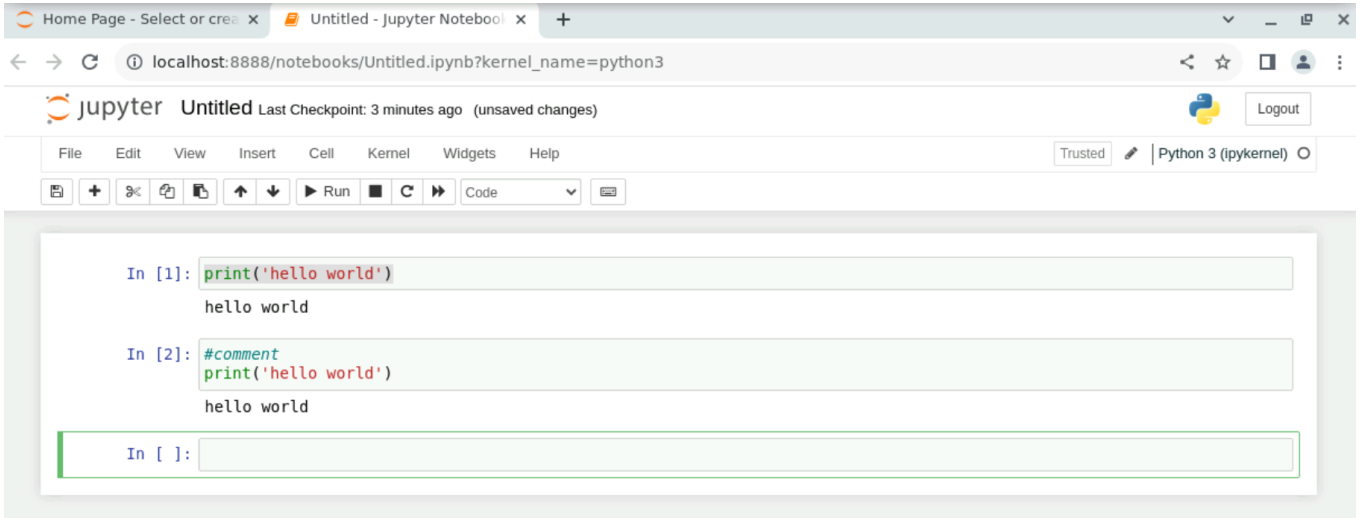
- Platform-independent: works on different platform
- Python has a syntax similar to English language

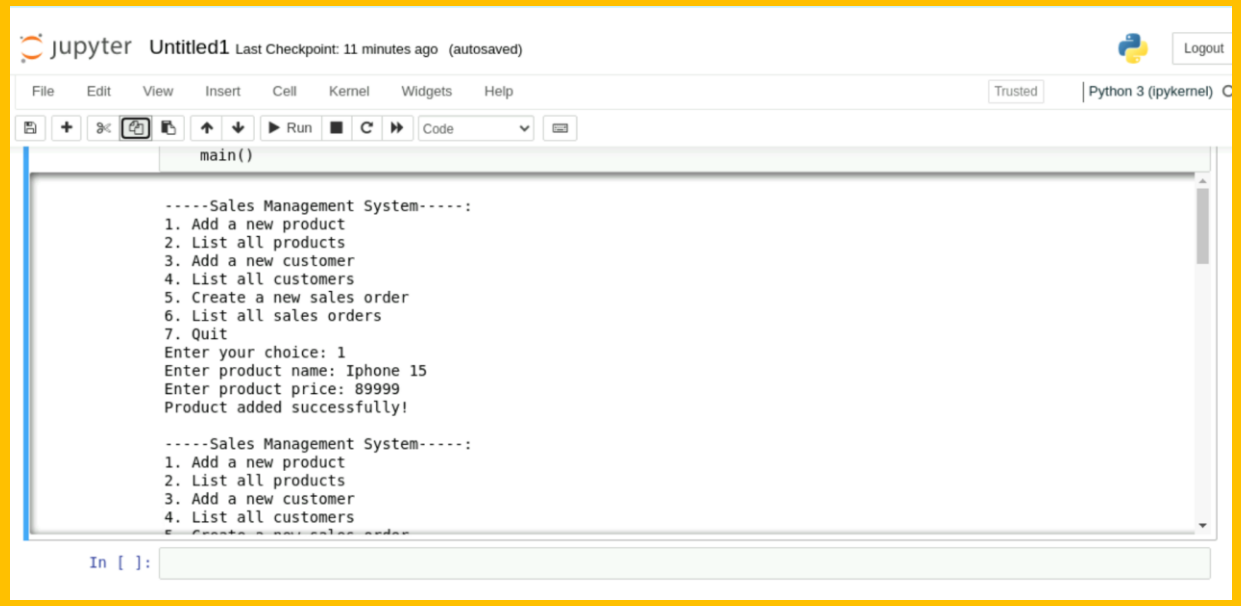
English	Python
• Vocabulary • Grammar	• Keywords • Syntax

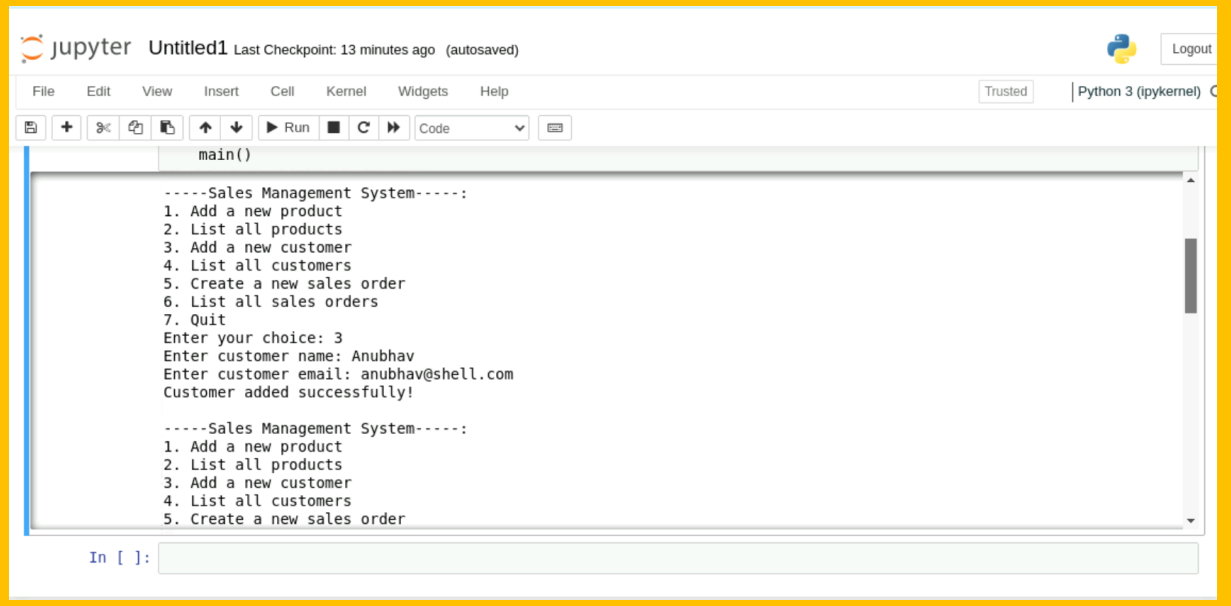
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick
- Python can be treated in a procedural way or object-oriented or functional way

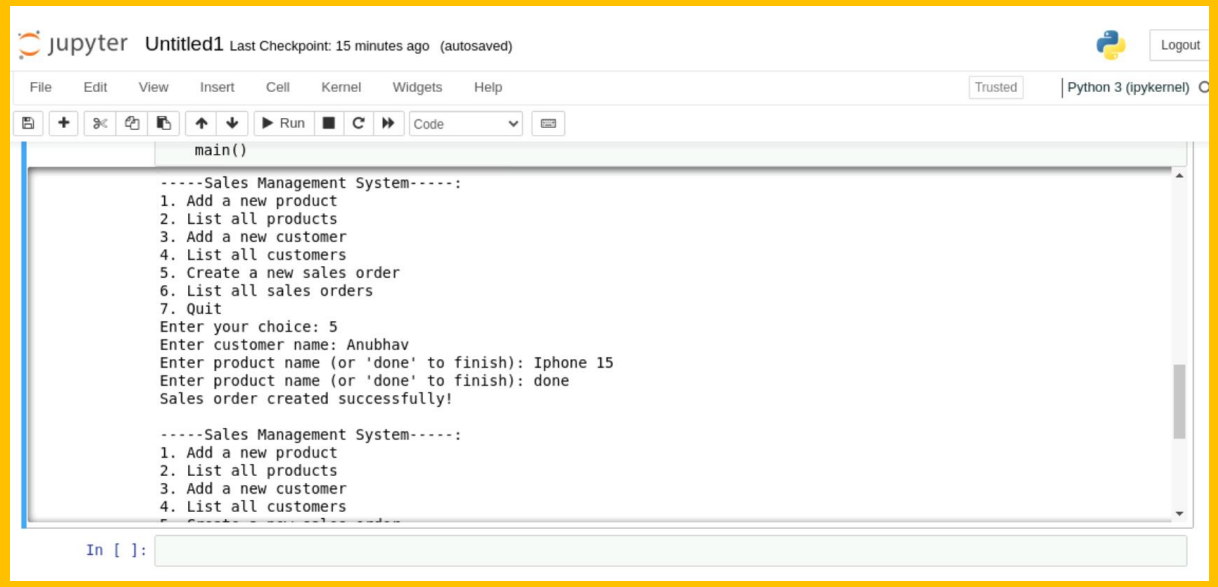
PYTHON INDENTATION

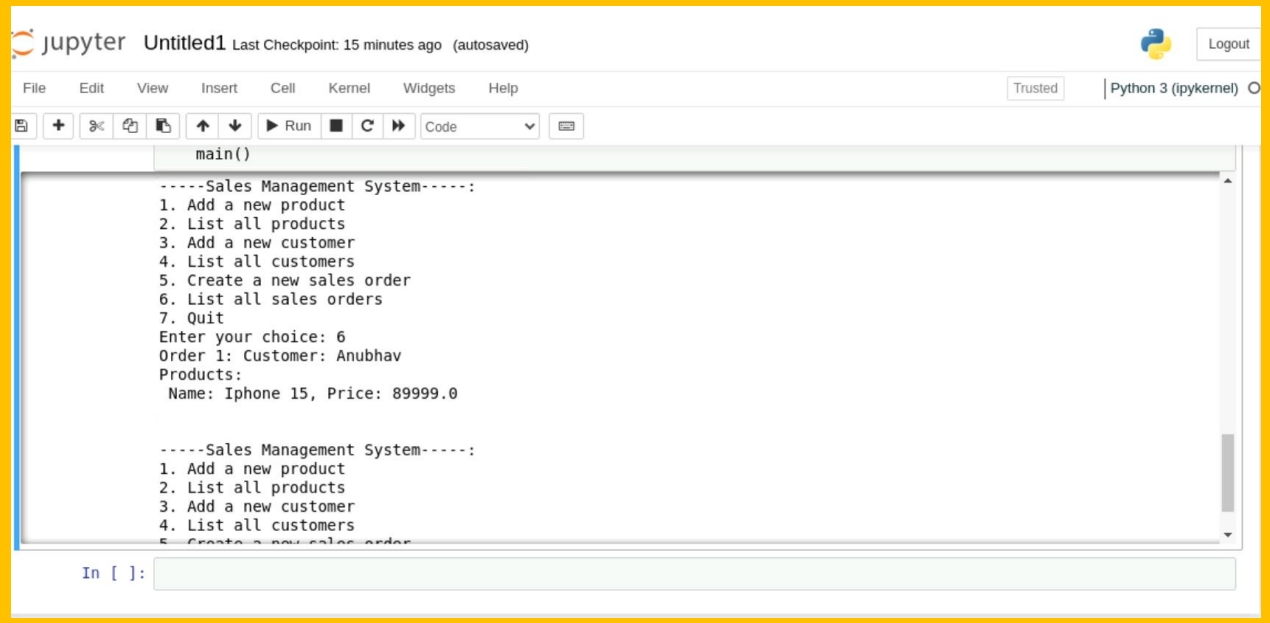
- Indentation refers to the spaces at the beginning of a code line
- While in other languages, the indentation in code is used for readability only, the indentation in Python is very important
- Python uses indentation in a block of code




jupyter Untitled1 Last Checkpoint: 11 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
main()
-----Sales Management System-----:
1. Add a new product
2. List all products
3. Add a new customer
4. List all customers
5. Create a new sales order
6. List all sales orders
7. Quit
Enter your choice: 1
Enter product name: Iphone 15
Enter product price: 89999
Product added successfully!
-----Sales Management System-----:
1. Add a new product
2. List all products
3. Add a new customer
4. List all customers
In [ ]:


jupyter Untitled1 Last Checkpoint: 13 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
main()
-----Sales Management System-----:
1. Add a new product
2. List all products
3. Add a new customer
4. List all customers
5. Create a new sales order
6. List all sales orders
7. Quit
Enter your choice: 3
Enter customer name: Anubhav
Enter customer email: anubhav@shell.com
Customer added successfully!
-----Sales Management System-----:
1. Add a new product
2. List all products
3. Add a new customer
4. List all customers
5. Create a new sales order
In [ ]:


jupyter Untitled1 Last Checkpoint: 15 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
main()
-----Sales Management System-----:
1. Add a new product
2. List all products
3. Add a new customer
4. List all customers
5. Create a new sales order
6. List all sales orders
7. Quit
Enter your choice: 5
Enter customer name: Anubhav
Enter product name (or 'done' to finish): Iphone 15
Enter product name (or 'done' to finish): done
Sales order created successfully!
-----Sales Management System-----:
1. Add a new product
2. List all products
3. Add a new customer
4. List all customers
In [ ]:


jupyter Untitled1 Last Checkpoint: 15 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
main()
-----Sales Management System-----:
1. Add a new product
2. List all products
3. Add a new customer
4. List all customers
5. Create a new sales order
6. List all sales orders
7. Quit
Enter your choice: 6
Order 1: Customer: Anubhav
Products:
Name: Iphone 15, Price: 89999.0
-----Sales Management System-----:
1. Add a new product
2. List all products
3. Add a new customer
4. List all customers
In [ ]:

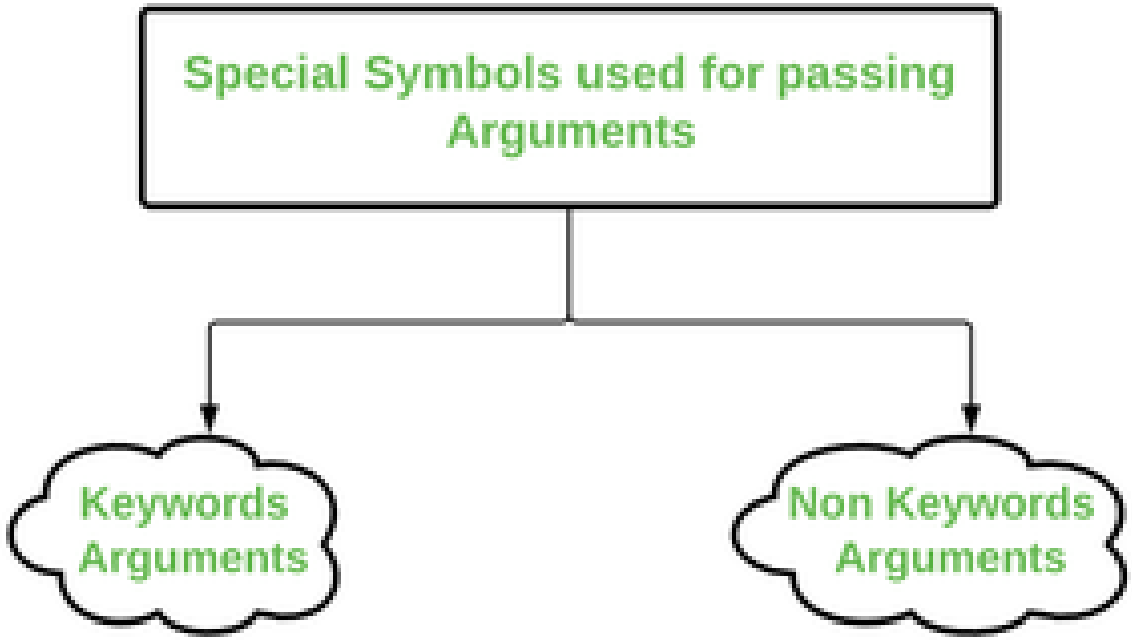
Week 5 Custom IDA Training

19 Sept 2023 – 22 Sept 2023

- Python (Continued)
- PySpark

*args & **kwargs in Python

- *args (Non-keyword arguments) : The special syntax *args in function definitions in Python is used to pass a variable number of arguments to a function. It is used to pass a non-keyworded, variable-length argument list.
- **kwargs (Keyword arguments) : The special syntax **kwargs in function definitions in Python is used to pass a keyworded, variable-length argument list. The double star allows us to pass through keyword arguments (and any number of them).



KEYWORD VS POSITIONAL ARGUMENTS

Keyword-only Argument	Position-only Argument
• Parameter names are used to pass the arg during a function call • Order the parameter can be changed while passing the info • Function_name(param="value")	• We need to maintain the order of passing args; the args should be define while passing parameter • Order must not be changed • Function_name (value1, value2)

PYTHON - CLASSES AND OBJECTS

- Python is an object oriented programming language.
- Almost everything in Python is an object, with its properties and methods.
- A Class is like an object constructor, or a "blueprint" for creating objects.
- All classes have a function called __init__(), which is always executed when the class is being initiated.
- The self parameter is a reference to the current instance of the class, and is used to access variables that belongs to the class.

In [2]:
```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __str__(self):
        return f"{self.name}({self.age})"

p1 = Person("john", 25)

print(p1)
```

```
john(25)
```

In [4]:
```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

p1 = Person("john", 25)

print(p1)
```

```
<__main__.Person object at 0x7fa14f7e49d0>
```

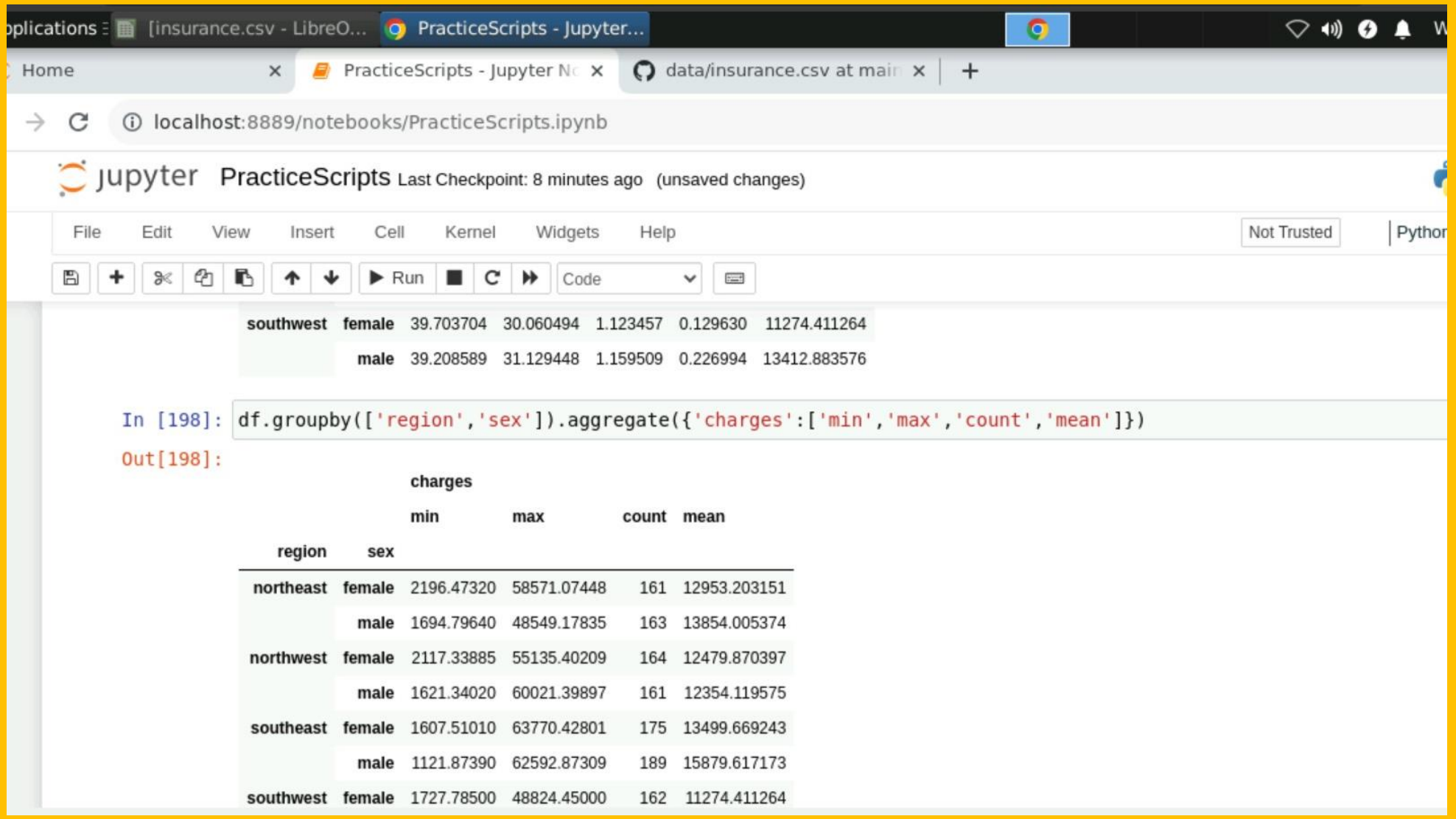
OBJECT-ORIENTED PROGRAMMING

- Inheritance: Inheritance is the capability of one class to derive or inherit the properties from another class.
- Polymorphism: Polymorphism simply means having many forms.
- Encapsulation: Wrapping data and the methods that work on data within one unit.
- Data Abstraction: It hides unnecessary code details from the user. Data abstraction in Python can be achieved by creating abstract classes.

LISTS IN PYTHON

- Lists are used to store multiple items in a single variable
- Lists are one of 4 built-in datatypes in python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage
- Lists are created using square brackets:
  - thisList = [“apple”, “banana”, “cherry”]
  - print(thisList)

Working with Dataframes



southwest	female	39.703704	30.060494	1.123457	0.129630	11274.411264
	male	39.208589	31.129448	1.159509	0.226994	13412.883576

In [198]:
```
df.groupby(['region','sex']).aggregate({'charges':['min','max','count','mean']})
```

Out[198]:

		charges			
		min	max	count	mean
region	sex				
northeast	female	2196.47320	58571.07448	161	12953.203151
	male	1694.79640	48549.17835	163	13854.005374
northwest	female	2117.33885	55135.40209	164	12479.870397
	male	1621.34020	60021.39897	161	12354.119575
southeast	female	1607.51010	63770.42801	175	13499.669243
	male	1121.87390	62592.87309	189	15879.617173
southwest	female	1727.78500	48824.45000	162	11274.411264

Spark vs RDD

Spark	RDD
• Unified engine for large scale data analytics • Multi-language support – Python, R, Scala, Java, C#, SQL • Can be used over multiple clusters • Follows master-slave architecture • Batch processing, Real-time processing, Data analytics, Machine Learning, Graph processing, In-memory processing	• Immutable – New RDD is created if update is carried out. • Type Inferred – Data type is inferred from the data itself. • Cacheable – can be made persistent • Lazy Evaluation – execute only when required to (when actions are called) • Fault Tolerance - recoverable

Word Count in a File using Spark

```
>>> words = srcfile.flatMap(lambda line: line.split(" "))
>>> mappedwords = words.map(lambda word: (word, 1))
>>> countz = mappedwords.reduceByKey(lambda a,b:a+b)
>>> words.collect()
['PySpark', 'Day', '1', 'Working', 'in', 'the', 'VM.', 'Sun', 'is', 'bright', 'today?', 'Too', 'bright', 'maybe']
>>> mappedwords.collect()
[('PySpark', 1), ('Day', 1), ('1', 1), ('Working', 1), ('in', 1), ('the', 1), ('VM.', 1), ('Sun', 1), ('is', 1), ('bright', 1), ('today?', 1), ('Too', 1), (
'bright', 1), ('maybe', 1)]
>>> countz.collect()
[('PySpark', 1), ('Day', 1), ('Sun', 1), ('today?', 1), ('1', 1), ('Working', 1), ('in', 1), ('the', 1), ('VM.', 1), ('is', 1), ('bright', 2), ('Too', 1), (
'maybe', 1)]
```

Applications | PySparkShell - Details f... | labuser@ip-172-31-14-8... | datasets - File Manager | Thu 21 Sep, 05:34

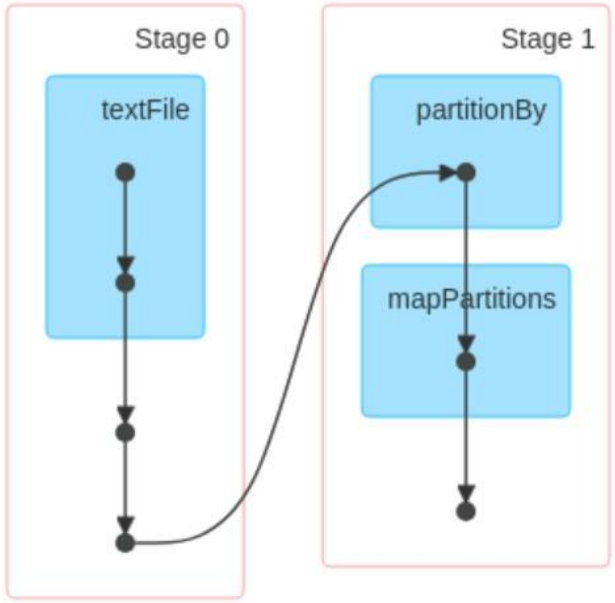
Examples | Apache Spark | PySparkShell - Details for

localhost:4040/jobs/job/?id=0

Details for Job 0

Status: SUCCEEDED
Submitted: 2023/09/21 04:53:34
Duration: 5 s
Completed Stages: 2

▸ Event Timeline
▾ DAG Visualization



▾ Completed Stages (2)

CREATING DATAFRAMES

```
SparkSession available as 'spark'.
>>> emp = [(1,"Smith",1,"2018","10","M",3000.00),
...     (2,"Rose",1,"2010","20","M",4000.00),
...     (3,"Williams",1,"2010","10","M",1000.00),
...     (4,"Jones",2,"2005","10","F",2000.00),
...     (5,"Brown",2,"2010","40","",300.00),
...       (6,"Brown",2,"2010","50","",2000.00)
...   ]
>>> EmpSchema = StructType([
...     StructField('Emp_id', IntegerType(), True),
...     StructField('Empname', StringType(), True),
...     StructField('MGR', IntegerType(), True),
...    StructField('YOJ', StringType(), True),
...      StructField('dept_id', StringType(), True),
...     StructField('gender', StringType(), True),
...      StructField('salary', DoubleType(), True)
... ])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'StructType' is not defined
>>> from pyspark.sql.types import StructType,StructField, StringType,I
>>> EmpSchema = StructType([
...     StructField('Emp_id', IntegerType(), True),
...     StructField('Empname', StringType(), True),
...     StructField('MGR', IntegerType(), True),
...    StructField('YOJ', StringType(), True),
...      StructField('dept_id', StringType(), True),
...     StructField('gender', StringType(), True),
...      StructField('salary', DoubleType(), True)
... ])
>>> empDF = spark.createDataFrame(data=emp, schema = EmpSchema)
>>> empDF.show()
```

```
In [6]: salesdf.rdd.getNumPartitions()
Out[6]: 2

In [7]: 1 salesdf.count()
Out[7]: 51290

In [8]: salesdf_repart = salesdf.repartition(5)

In [9]: salesdf_repart.rdd.getNumPartitions()
[Stage 5:==============================>                          (1 + 1) / 2]
Out[9]: 5
```

```
>>> empDF.orderBy(["dept_id","salary"], ascending = [True,False]).show()
+------+--------+---+----+-------+------+------+
|Emp_id| Empname|MGR| YOJ|dept_id|gender|salary|
+------+--------+---+----+-------+------+------+
|     1|   Smith|  1|2018|     10|     M|3000.0|
|     4|   Jones|  2|2005|     10|     F|2000.0|
|     3|Williams|  1|2010|     10|     M|1000.0|
|     2|    Rose|  1|2010|     20|     M|4000.0|
|     5|   Brown|  2|2010|     40|      | 300.0|
|     6|   Brown|  2|2010|     50|      |2000.0|
+------+--------+---+----+-------+------+------+
```

Week 6 Custom IDA Training

25 Sept 2023 – 29 Sept 2023

- PySpark (Continued)
- Databricks
- Azure Containers
- Azure DevOps

PYSPARK DATAFRAME

A DataFrame is equivalent to a relational table in spark SQL and can be created through various functions in SparkSessions.

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/09/25 03:53:24 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
where applicable
```

In [4]:
```
df=spark.createDataFrame(users)
```

In [5]:
```
df.show()
```

```
+-----------+-------+------------+------------+--------------------+----------+------+---+----------+------------+---------
---------+--------------------+
|amount_paid|courses| current_city|customer_from|               email|first_name|gender| id|is_customer|   last_name|    last_
updated_ts|       phone_numbers|
+-----------+-------+------------+------------+--------------------+----------+------+---+----------+------------+---------
---------+--------------------+
|    1000.55| [1, 2]|       Dallas|  2021-01-15|cvandenoord0@etsy...|    Corrie|  male|  1|      true|Van den Oord|2021-02-1
0 01:15:00|{+1 234 567 8901,...|
|      900.0|    [3]|      Houston|  2021-02-14|nbrewitt1@dailyma...|  Nikolaus|  male|  2|      true|     Brewitt|2021-02-1
8 03:33:00|{+1 234 567 8923,...|
|     850.55| [2, 4]|             |  2021-01-21|openney2@vistapri...|    Orelie|female|  3|      true|      Penney|2021-03-1
5 15:16:55|{+1 714 512 9752,...|
|       null|     []|San Fransisco|        null|  amaddocks3@home.pl|     Ashby|  male|  4|     false|    Maddocks|2021-04-1
0 17:45:30|       {null, null}|
|       null|     []|         null|        null|krome4@shutterfly...|      Kurt|female|  5|     false|        Rome|2021-04-0
2 00:55:18|{+1 817 934 7142,...|
+-----------+-------+------------+------------+--------------------+----------+------+---+----------+------------+---------
---------+--------------------+
```

In [7]:
```
df.select("id","current_city","customer_from").show()
```

In [23]:
```
#sort the data in asc order by customer _from with null values should come at end
df.select("id","current_city","customer_from").orderBy("customer_from",ascending=False).show()
```

```
+---+------------+-------------+
| id| current_city|customer_from|
+---+------------+-------------+
|  2|      Houston|   2021-02-14|
|  3|             |   2021-01-21|
|  1|       Dallas|   2021-01-15|
|  4|San Fransisco|         null|
|  5|         null|         null|
+---+------------+-------------+
```

In [26]:
```
df.select("id","current_city","customer_from").orderBy(df["customer_from"].asc_nulls_last()).show()
```

```
+---+------------+-------------+
| id| current_city|customer_from|
+---+------------+-------------+
|  1|       Dallas|   2021-01-15|
|  3|             |   2021-01-21|
|  2|      Houston|   2021-02-14|
|  4|San Fransisco|         null|
|  5|         null|         null|
+---+------------+-------------+
```

In [27]:
```
#2nd dataset
zipdf=spark.read.option("header",True).option("inferschema",True).csv("/home/labuser/Downloads/zipcode.csv")
```

In [28]:
```
zipdf.show()
```

PYSPARK PARTITIONBY()

PySpark partitionBy() is a way to split a large dataset into smaller datasets based on one or more partition keys. When you create a DataFrame from a file/table, based on certain parameters PySpark creates the DataFrame with certain number of partitions in memory.

```
In [26]: #partionby makes 8 different folders
         names_df.write.mode("overwrite").partitionBy("year","sex").option("path","/home/labuser/Downloads/partitionbygendert").sa

In [27]: spark.sql("select * from babynames_gender ").show()
```

```
+----------+------+-----+----+---+
|First Name|County|Count|Year|Sex|
+----------+------+-----+----+---+
|       AVA|Albany|   23|2014|  F|
|      EMMA|Albany|   20|2014|  F|
|    SOPHIA|Albany|   19|2014|  F|
|    OLIVIA|Albany|   19|2014|  F|
|       MIA|Albany|   15|2014|  F|
| CHARLOTTE|Albany|   15|2014|  F|
|    AMELIA|Albany|   14|2014|  F|
|  ISABELLA|Albany|   13|2014|  F|
|   ABIGAIL|Albany|   12|2014|  F|
|     CHLOE|Albany|   12|2014|  F|
|      ELLA|Albany|   12|2014|  F|
|      ANNA|Albany|   12|2014|  F|
|     EMILY|Albany|   11|2014|  F|
|     GRACE|Albany|   11|2014|  F|
|    EVELYN|Albany|   11|2014|  F|
|   NATALIE|Albany|   11|2014|  F|
|   MADISON|Albany|   10|2014|  F|
|  SCARLETT|Albany|   10|2014|  F|
|      LILY|Albany|    9|2014|  F|
| ELIZABETH|Albany|    8|2014|  F|
+----------+------+-----+----+---+
only showing top 20 rows
```

PYSPARK WRITESTREAM

Interface for saving the content of the streaming DataFrame out into external storage.

2023-09-26T10:40:00.000+0000	logout	user3	null
2023-09-26T10:45:00.000+0000	page_view	user1	page789
2023-09-26T10:50:00.000+0000	login	user2	null
2023-09-26T10:55:00.000+0000	logout	user1	null
2023-09-26T11:00:00.000+0000	page_view	user2	page101
2023-09-26T11:05:00.000+0000	login	user1	null
2023-09-26T11:10:00.000+0000	page_view	user3	page222

In [0]:
```
outputstream="dbfs:/mnt/saunext/inputfiles/outputstream"
```

In [0]:
```
df.writeStream\
.option("checkpointlocation",f"{outputstream}/naval/checkpoint")\
.option("path",f"{outputstream}/OhmOmar/output")\
.table("2609db.jsonsample")
```

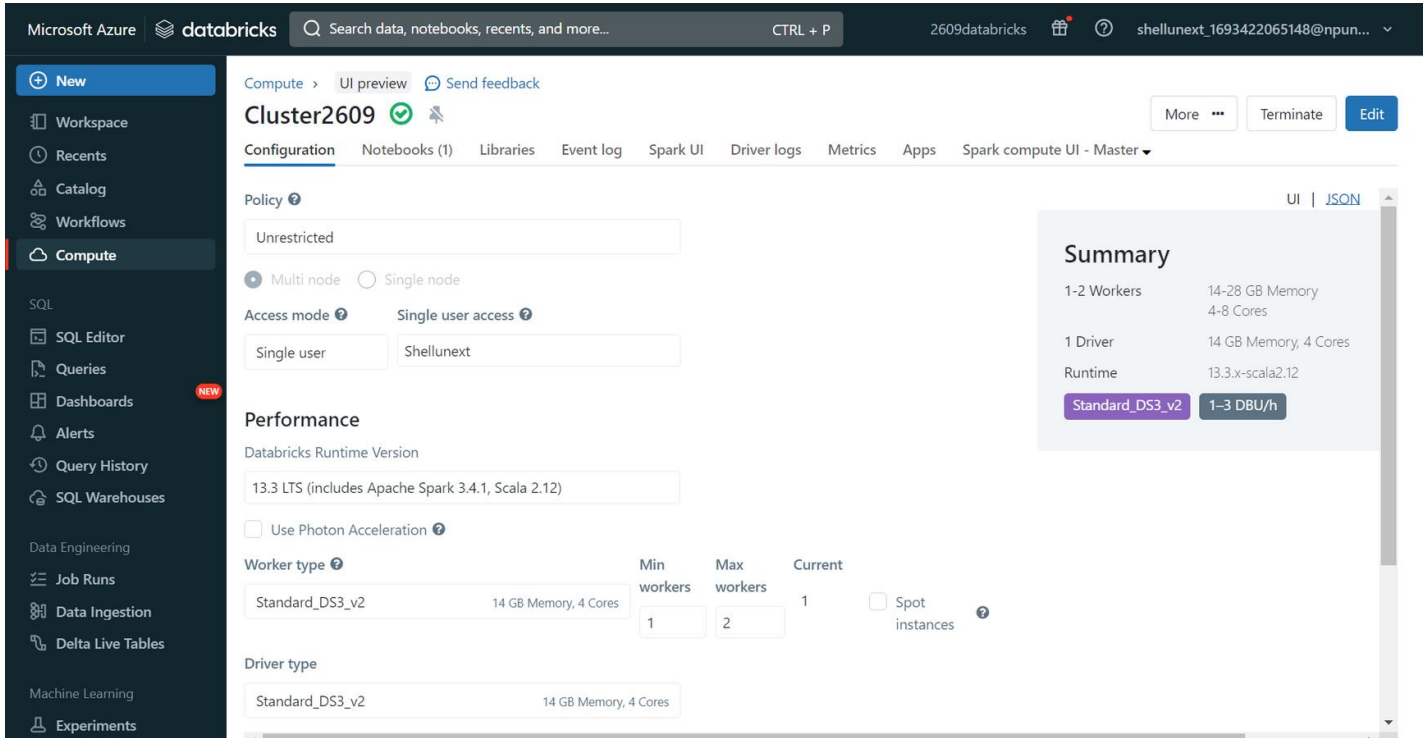
Out[0]:
```
<pyspark.sql.streaming.query.StreamingQuery at 0x7efd48db4be0>
```

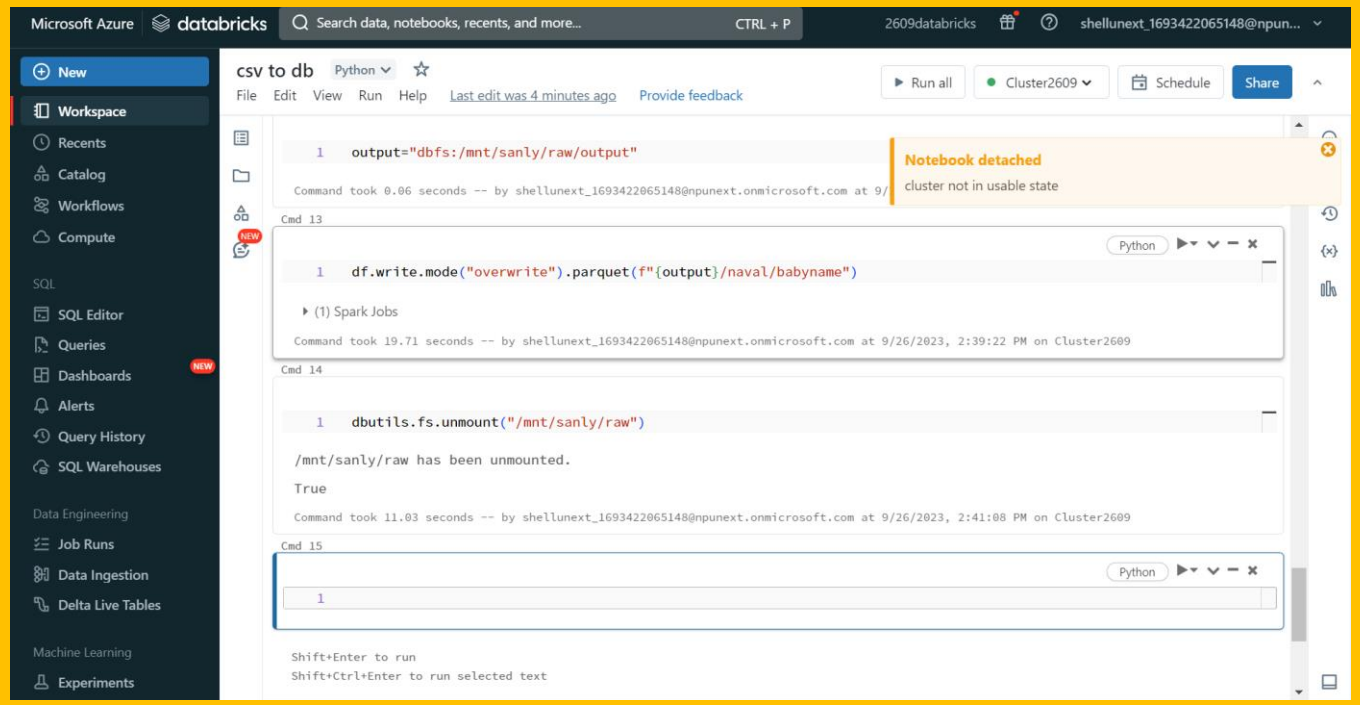
In [0]:
```
for stream in spark.streams.active:
    stream.stop()
```

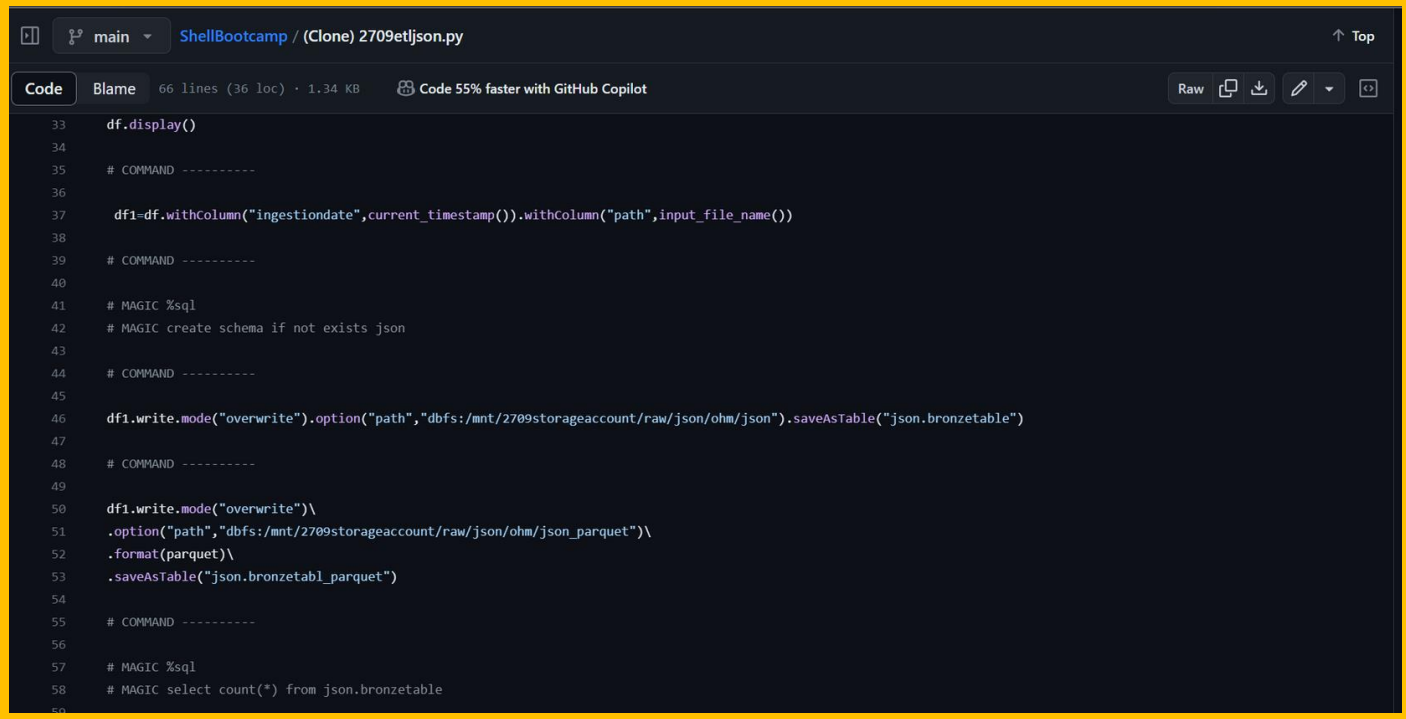
In [0]:
```
(spark
.readStream
.schema(users_sch)
.json("dbfs:/mnt/saunext/inputfiles/inputstream/")
```

DATABRICKS

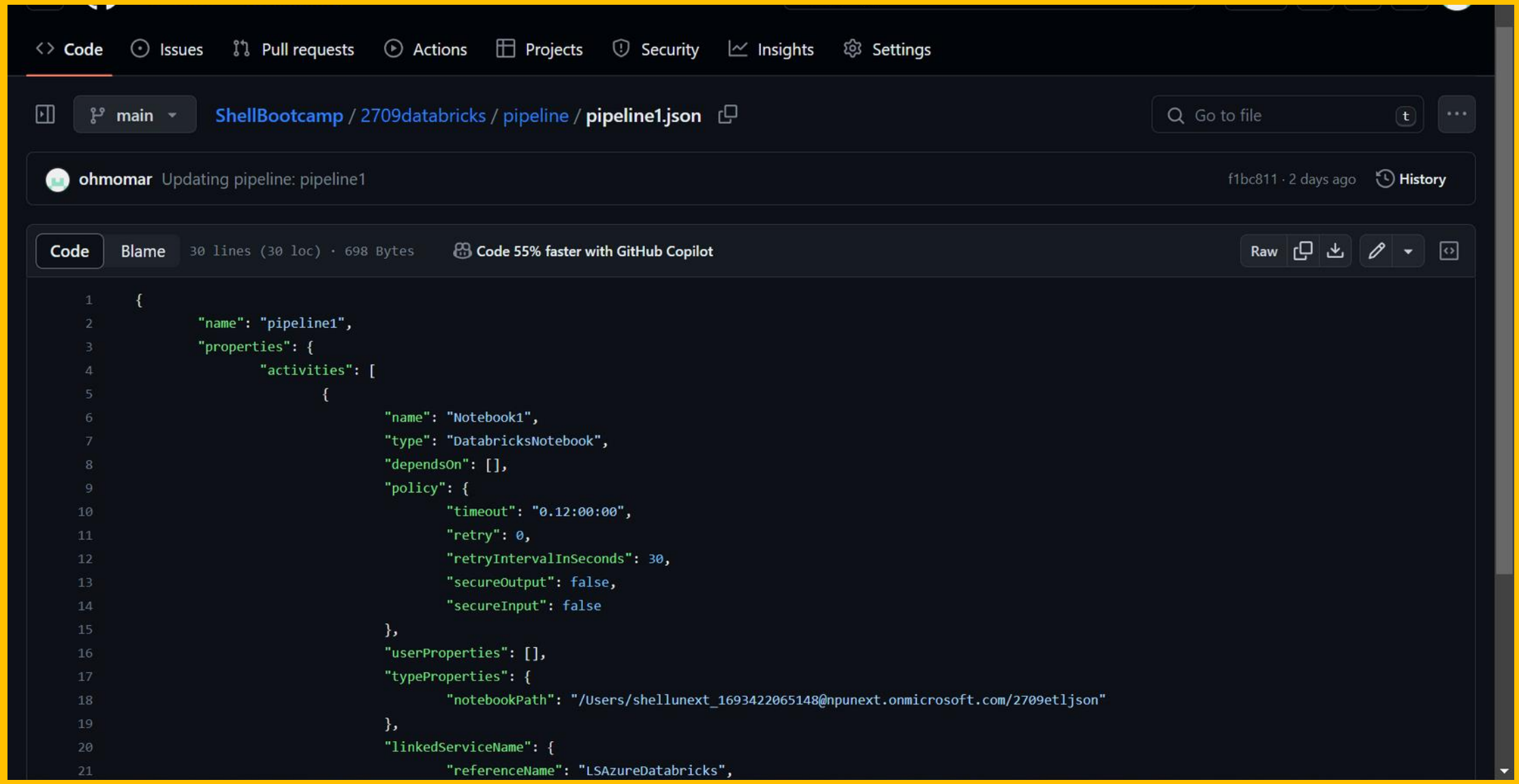
Databricks combines data warehouse and data lakes into a lakehouse architecture. It is used to process, store, clean, share, analyze, model, and monetize their datasets with solutions from BI to machine learning.




Microsoft Azure
databricks
Search data, notebooks, recents, and more... CTRL + P
2609databricks
shellunext_1693422065148@npun...
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Experiments
csv to db
Python
File Edit View Run Help
Last edit was 4 minutes ago
Provide feedback
Run all
Cluster2609
Schedule
Share
1 output="dbfs:/mnt/sanly/raw/output"
Command took 0.06 seconds -- by shellunext_1693422065148@npunext.onmicrosoft.com at 9/
Notebook detached
cluster not in usable state
Cmd 13
Python
1 df.write.mode("overwrite").parquet(f"{output}/naval/babyname")
(1) Spark Jobs
Command took 19.71 seconds -- by shellunext_1693422065148@npunext.onmicrosoft.com at 9/26/2023, 2:39:22 PM on Cluster2609
Cmd 14
1 dbutils.fs.unmount("/mnt/sanly/raw")
/mnt/sanly/raw has been unmounted.
True
Command took 11.03 seconds -- by shellunext_1693422065148@npunext.onmicrosoft.com at 9/26/2023, 2:41:08 PM on Cluster2609
Cmd 15
Python
1
Shift+Enter to run
Shift+Ctrl+Enter to run selected text

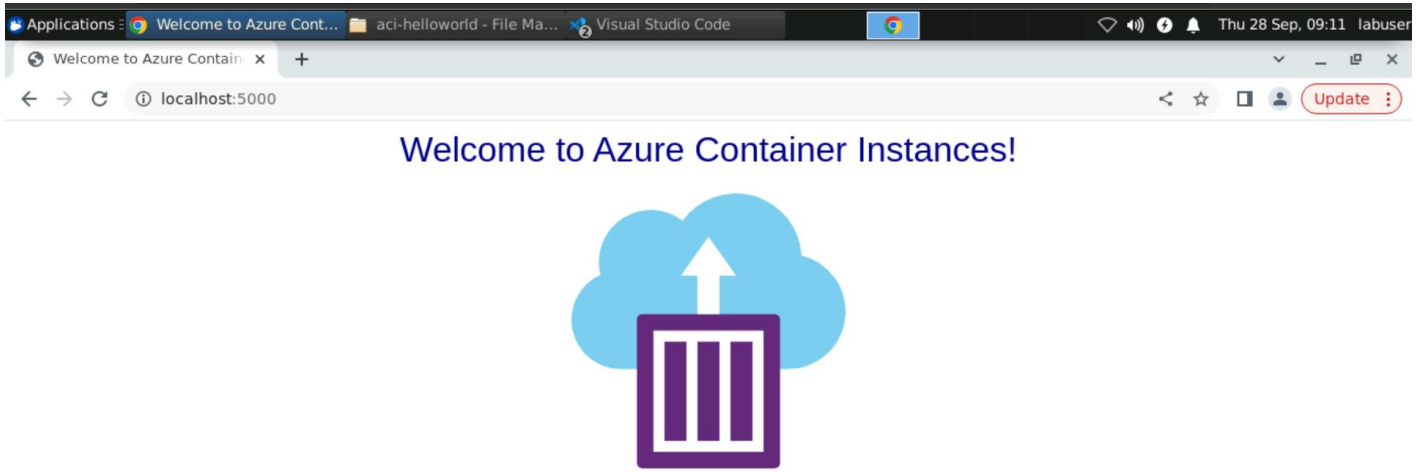

main
ShellBootcamp / (Clone) 2709etljson.py
Top
Code
Blame
66 lines (36 loc) · 1.34 KB
Code 55% faster with GitHub Copilot
Raw
df.display()
# COMMAND ----------
df1=df.withColumn("ingestiondate",current_timestamp()).withColumn("path",input_file_name())
# COMMAND ----------
# MAGIC %sql
# MAGIC create schema if not exists json
# COMMAND ----------
df1.write.mode("overwrite").option("path","dbfs:/mnt/2709storageaccount/raw/json/ohm/json").saveAsTable("json.bronzetable")
# COMMAND ----------
df1.write.mode("overwrite")\
.option("path","dbfs:/mnt/2709storageaccount/raw/json/ohm/json_parquet")\
.format(parquet)\
.saveAsTable("json.bronzetabl_parquet")
# COMMAND ----------
# MAGIC %sql
# MAGIC select count(*) from json.bronzetable

Pushing to GitHub (Using ADF)



DOCKER

- It is a set of platform as a service (PaaS) products that use the operating system level virtualization to deliver software in packages called containers.
- **Docker Container**: Containers are isolated from one another and bundle their own software, libraries, and configuration files; they can communicate with each other through well-defined channels.

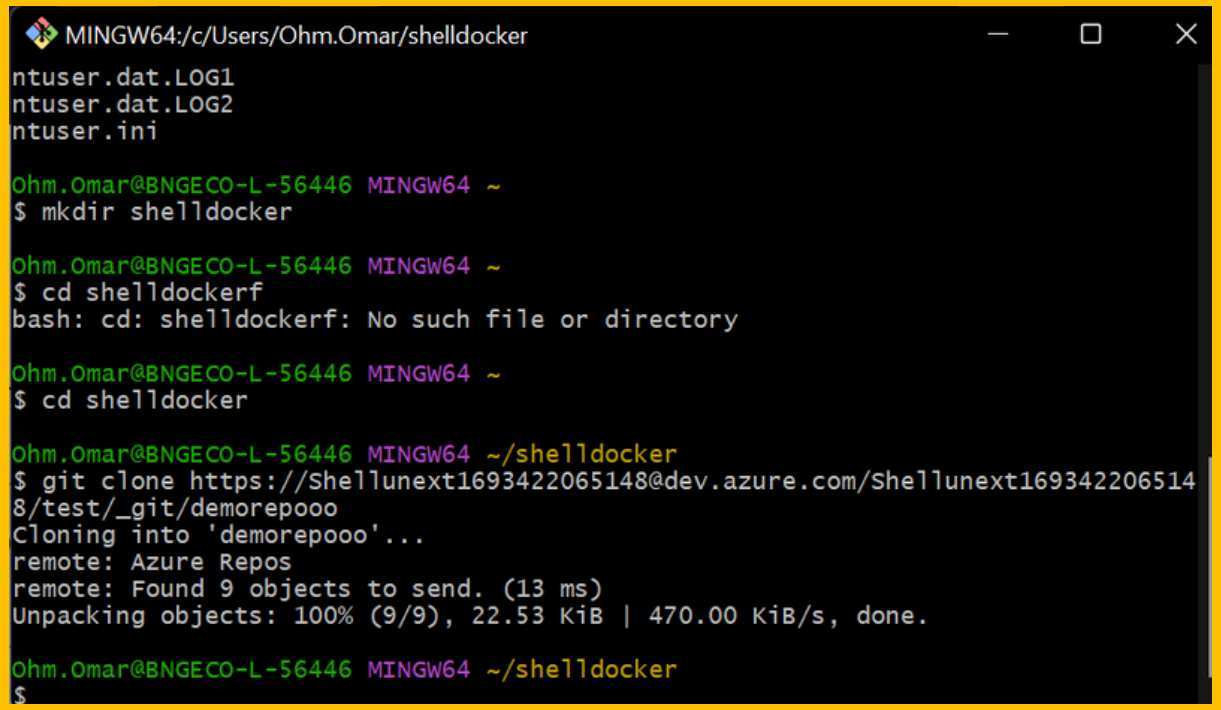


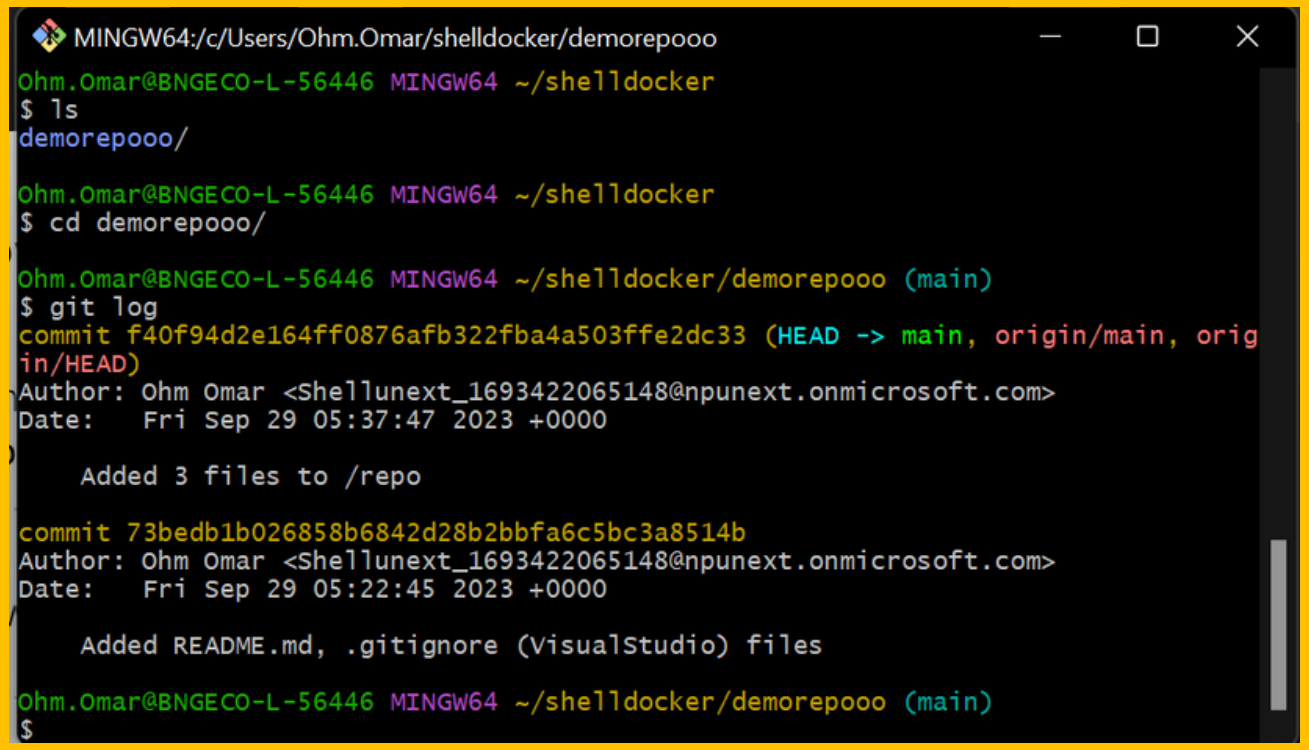
DEVOPS

- Continuous Integration: It drives the ongoing merging and testing of code, which leads to finding defects early.
- Continuous Delivery: Continuous Delivery of software solutions to production and testing environments helps organizations quickly fix bugs and respond to ever-changing business requirements.
- Version Control: Version Control, usually with a Git-based Repository, enables teams located anywhere in the world to communicate effectively during daily development activities.
- Agile/lean: Plan and isolate work into sprints. Manage team capacity and help teams quickly adapt to changing business needs.

DevOps: Waterfall vs Agile Approach

Waterfall Approach	Agile Approach
• Define, analyze, build and test, and deliver • Hard to accurately define requirements, which can change over time, including during development. • Requires change requests and additional cost after delivery.	• Emphasizes constantly adaptive planning, and early delivery with continual improvement. • Development methods are based on releases and iterations. • At the end of each iteration, should have tested working code. • Is focused on shorter-term outcomes.


MINGW64:/c/Users/Ohm.Omar/shelldocker
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
Ohm.Omar@BNGECO-L-56446 MINGW64 ~
$ mkdir shelldocker
Ohm.Omar@BNGECO-L-56446 MINGW64 ~
$ cd shelldockerf
bash: cd: shelldockerf: No such file or directory
Ohm.Omar@BNGECO-L-56446 MINGW64 ~
$ cd shelldocker
Ohm.Omar@BNGECO-L-56446 MINGW64 ~/shelldocker
$ git clone https://Shellunext1693422065148@dev.azure.com/Shellunext1693422065148/test/_git/demorepooo
Cloning into 'demorepooo'...
remote: Azure Repos
remote: Found 9 objects to send. (13 ms)
Unpacking objects: 100% (9/9), 22.53 KiB | 470.00 KiB/s, done.
Ohm.Omar@BNGECO-L-56446 MINGW64 ~/shelldocker
$


MINGW64:/c/Users/Ohm.Omar/shelldocker/demorepooo
Ohm.Omar@BNGECO-L-56446 MINGW64 ~/shelldocker
$ ls
demorepooo/
Ohm.Omar@BNGECO-L-56446 MINGW64 ~/shelldocker
$ cd demorepooo/
Ohm.Omar@BNGECO-L-56446 MINGW64 ~/shelldocker/demorepooo (main)
$ git log
commit f40f94d2e164ff0876afb322fba4a503ffe2dc33 (HEAD -> main, origin/main, origin/HEAD)
Author: Ohm Omar <Shellunext_1693422065148@npunext.onmicrosoft.com>
Date: Fri Sep 29 05:37:47 2023 +0000
Added 3 files to /repo
commit 73bedb1b026858b6842d28b2bbfa6c5bc3a8514b
Author: Ohm Omar <Shellunext_1693422065148@npunext.onmicrosoft.com>
Date: Fri Sep 29 05:22:45 2023 +0000
Added README.md, .gitignore (VisualStudio) files
Ohm.Omar@BNGECO-L-56446 MINGW64 ~/shelldocker/demorepooo (main)
$

test
Overview
Boards
Repos
Files
Commits
Pushes
Branches
Tags
Pull requests
Advanced Security
Pipelines
Test Plans
Project settings

Added ecdc_cases.csv to /FirstDivision

Completed !1 Ohm Omar proposes to merge FirstDivision into main

Overview Files Updates Commits

Ohm Omar completed this pull request Just now Cherry-pick Revert

Merged PR 1: Added ecdc_cases.csv to /FirstDivision
9d039ad1 Ohm Omar Just now

Show details

No merge conflicts
Last checked Just now

Description

Added ecdc_cases.csv to /

Show everything (2)

Reviewers

Add

Required

No required reviewers

Optional

No optional reviewers

Tags

No tags

Work items

No work items

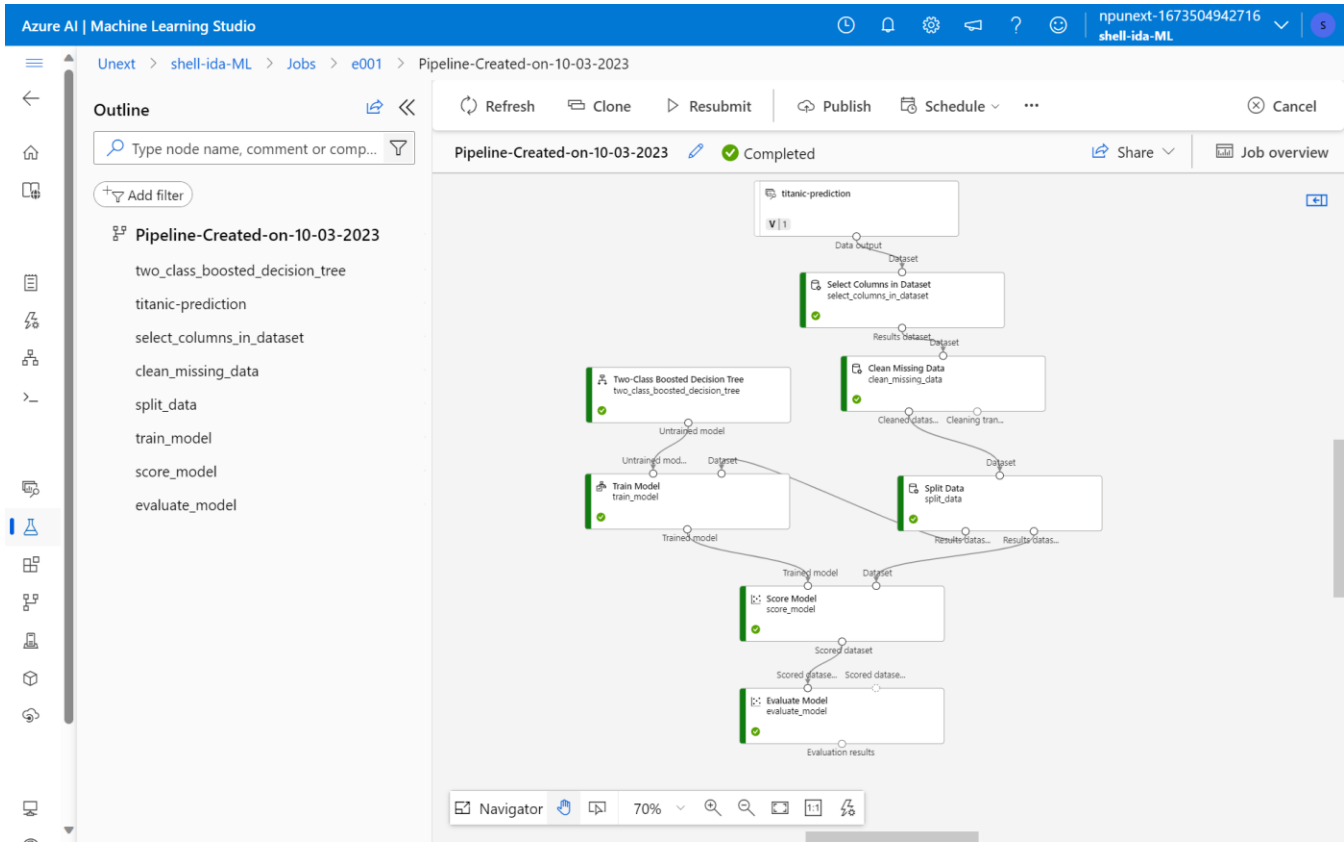
Week 7 Custom IDA Training

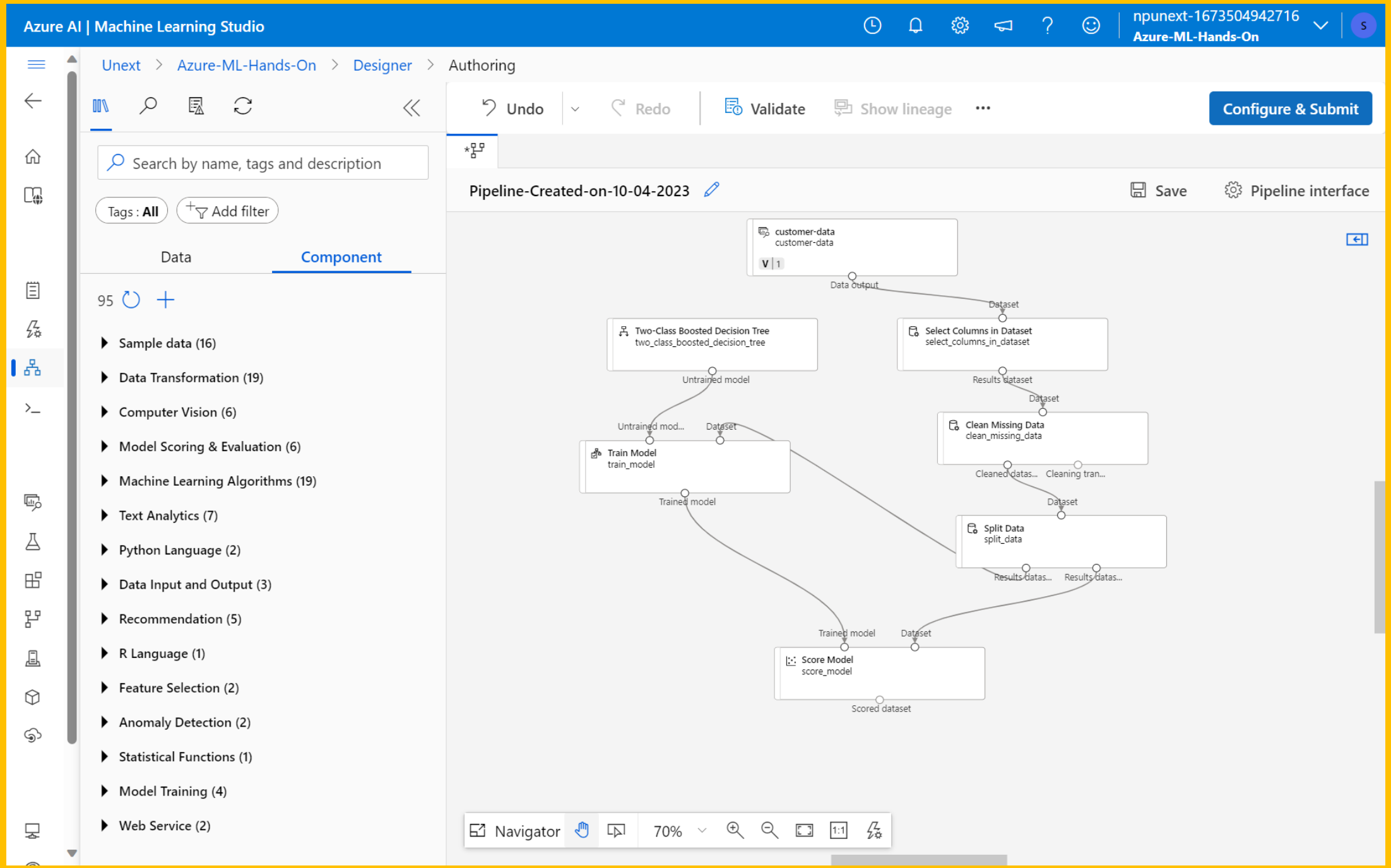
3 Oct 2023 – 6 Oct 2023

- Azure ML Studio
- Case Study Implementation & Demo

AZURE ML STUDIO

- Microsoft Azure Machine Learning Studio is a collaborative, drag-and-drop tool where you can use to build, test, and deploy predictive analytics solutions on your data.
- Machine Learning Studio publishes models as web services that can easily be consumed by custom apps or BI tools such as Excel.
- ML Studio is where data science, predictive analysis, cloud resources, and your data meet.




Azure AI | Machine Learning Studio
npunext-1673504942716
Azure-ML-Hands-On
Unext > Azure-ML-Hands-On > Designer > Authoring
Undo
Redo
Validate
Show lineage
Configure & Submit
Search by name, tags and description
Tags : All
Add filter
Data
Component
95
Sample data (16)
Data Transformation (19)
Computer Vision (6)
Model Scoring & Evaluation (6)
Machine Learning Algorithms (19)
Text Analytics (7)
Python Language (2)
Data Input and Output (3)
Recommendation (5)
R Language (1)
Feature Selection (2)
Anomaly Detection (2)
Statistical Functions (1)
Model Training (4)
Web Service (2)
Pipeline-Created-on-10-04-2023
Save
Pipeline interface
customer-data
customer-data
Data output
Dataset
Two-Class Boosted Decision Tree
two_class_boosted_decision_tree
Untrained model
Select Columns in Dataset
select_columns_in_dataset
Results dataset
Dataset
Clean Missing Data
clean_missing_data
Cleaned datas...
Cleaning tran...
Untrained mod...
Dataset
Train Model
train_model
Trained model
Dataset
Split Data
split_data
Results datas...
Results datas...
Trained model
Dataset
Score Model
score_model
Scored dataset
Navigator
70%

Evaluate Model

Overview | Parameters | Outputs + logs | Metrics | Child jobs | Images | Code | Explanations (preview) | Fairness (preview) | Monitoring

Refresh | Create custom chart | View as... | Current view: Local | Edit view

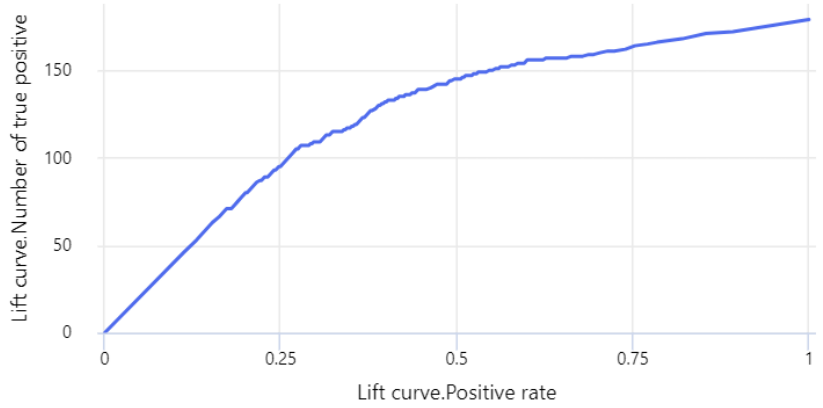
Select metrics

Accuracy	AUC	F1 Score	Precision	Recall
0.7842697	0.8353005	0.7192982	0.7546012	0.6871508

Confusion matrix

Chart visualization not available for non-numeric values.

Lift curve



Precision-recall curve

ROC curve

MARKETING FOR FINANCE CASE STUDY

BATCH 1 - GROUP 7

ANUBHAV BAGRI
DAKSH MALIK
EESHITA DEEPTA
NEHA JUYAL
SHARAD GUPTA

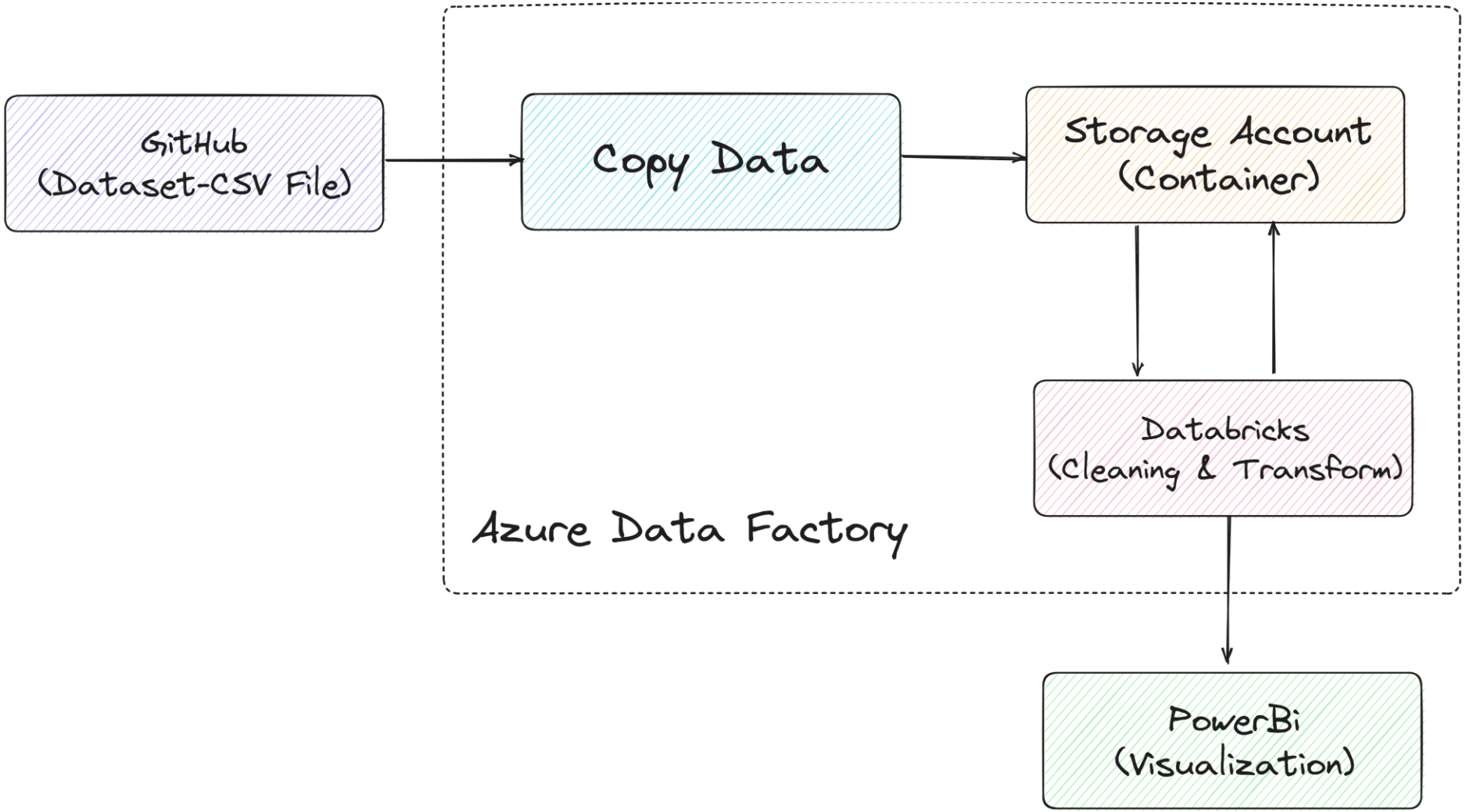
UNDER THE GUIDANCE OF ANANDH KUMAR M

PROBLEM STATEMENT

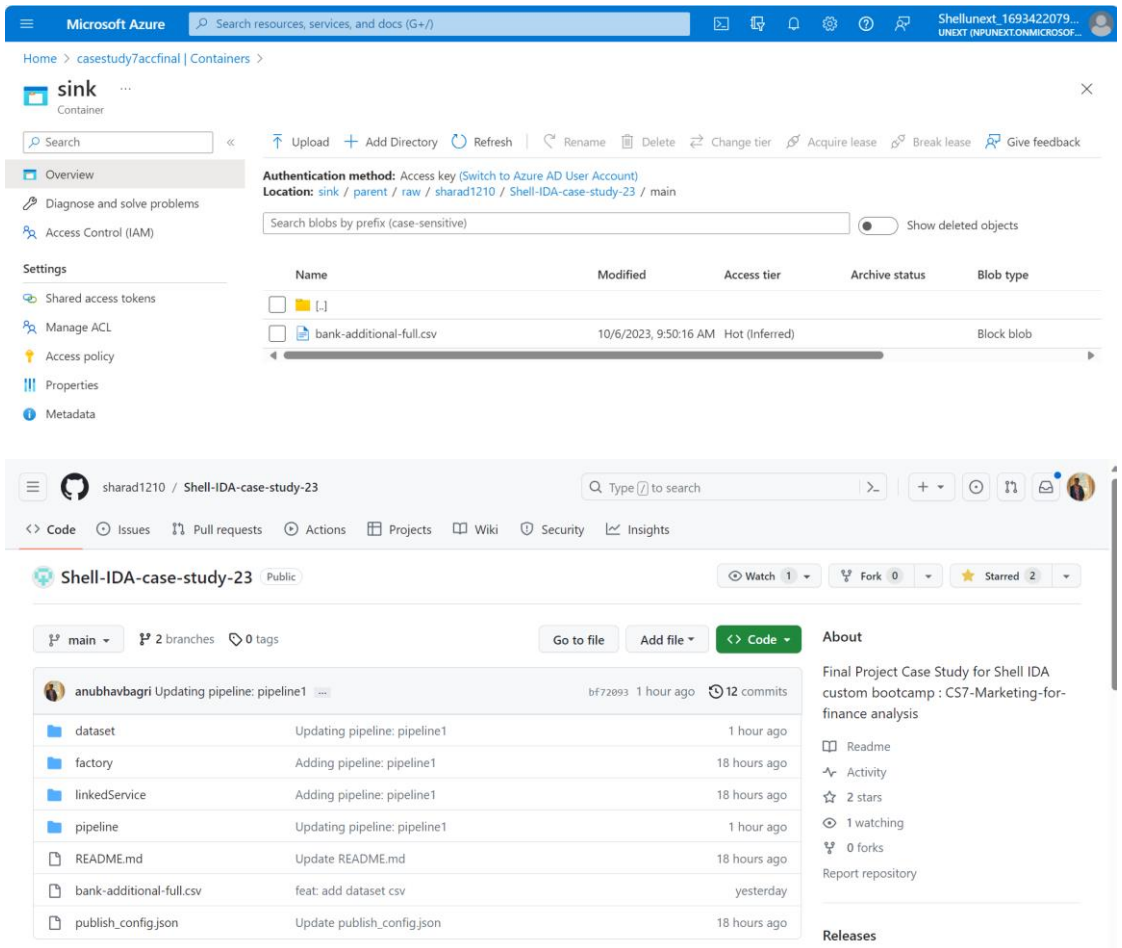
- The primary issue revolves around the challenge faced by bank officials in effectively targeting the appropriate individuals for a successful campaign.

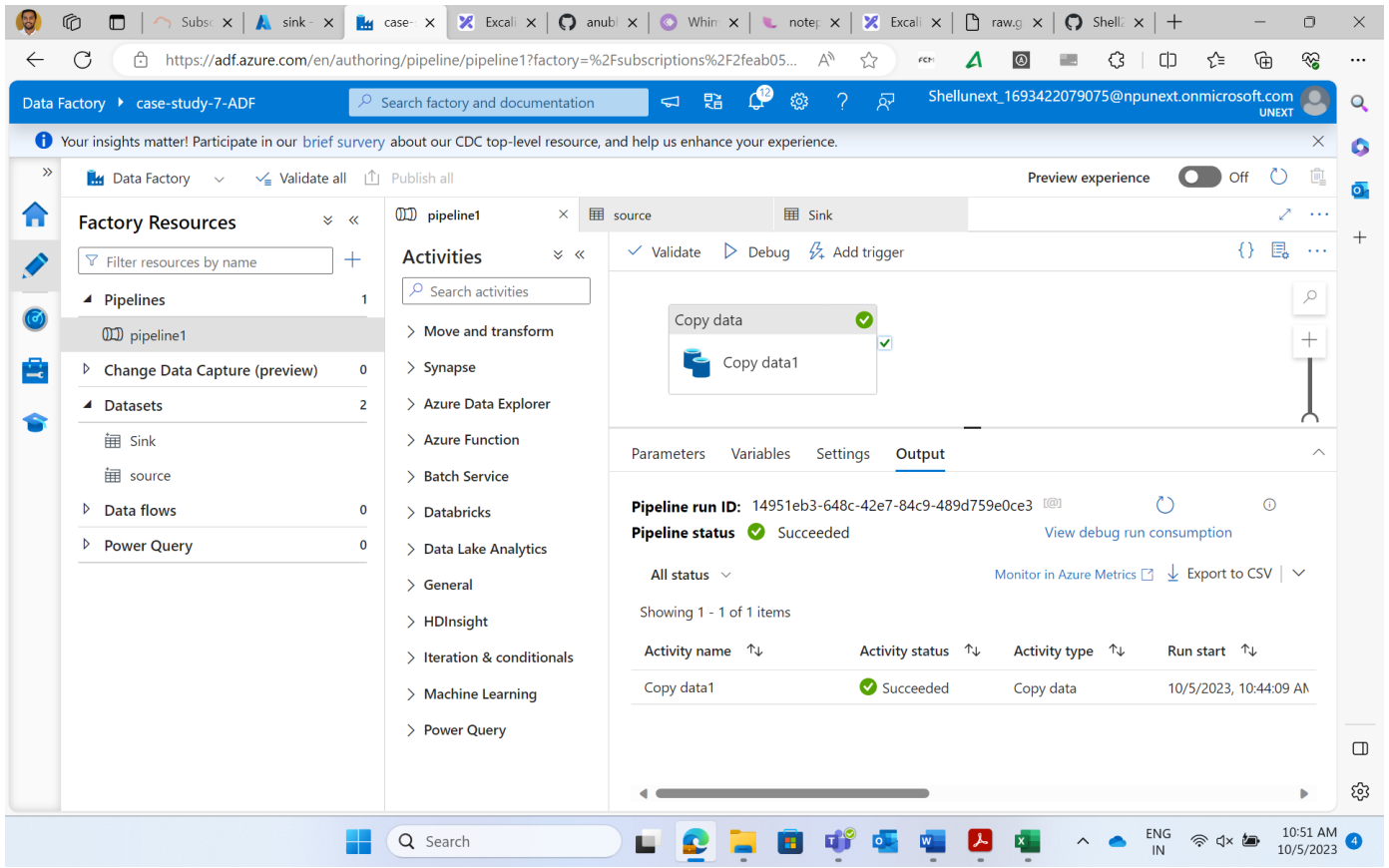
- Overall, the goal is to use data-driven insights to optimize marketing efforts and increase the success of the term-deposit campaign while ensuring data privacy and compliance with regulations.

ARCHITECTURE DESIGN

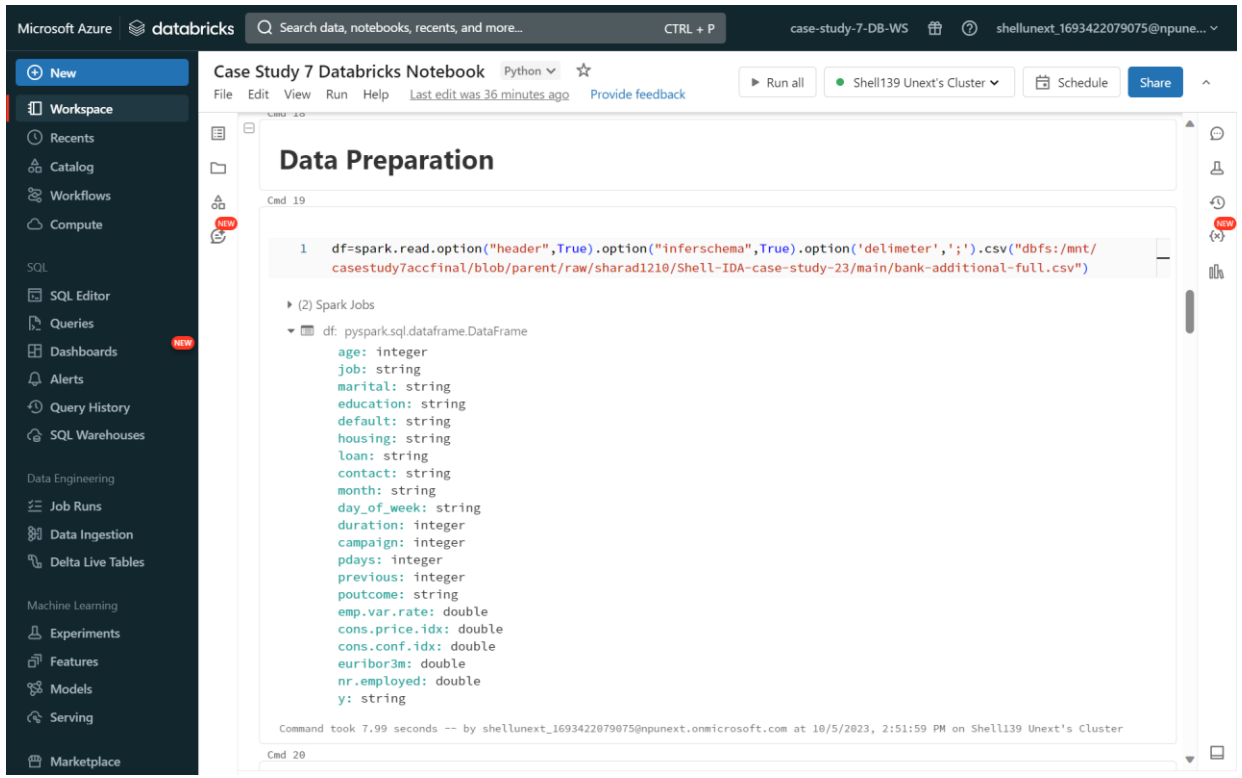


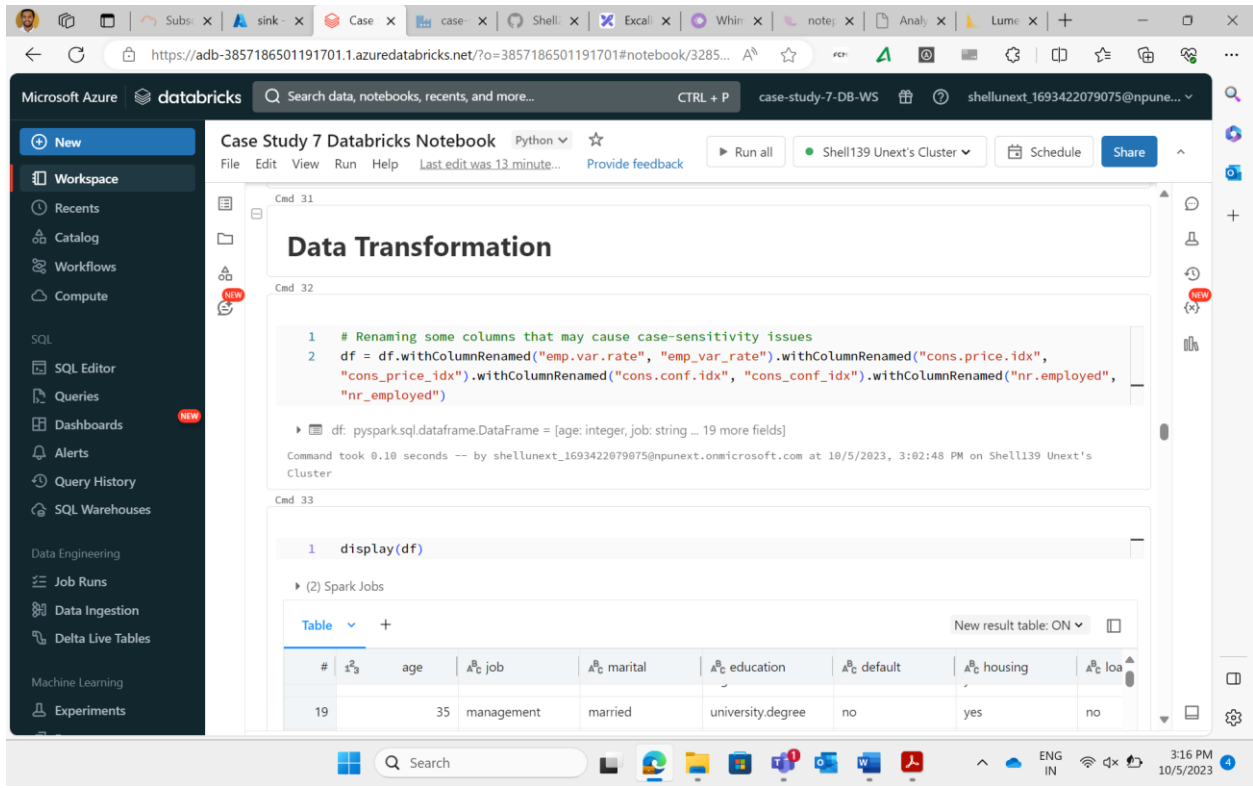
DATA LAYERS SNAPSHOTS



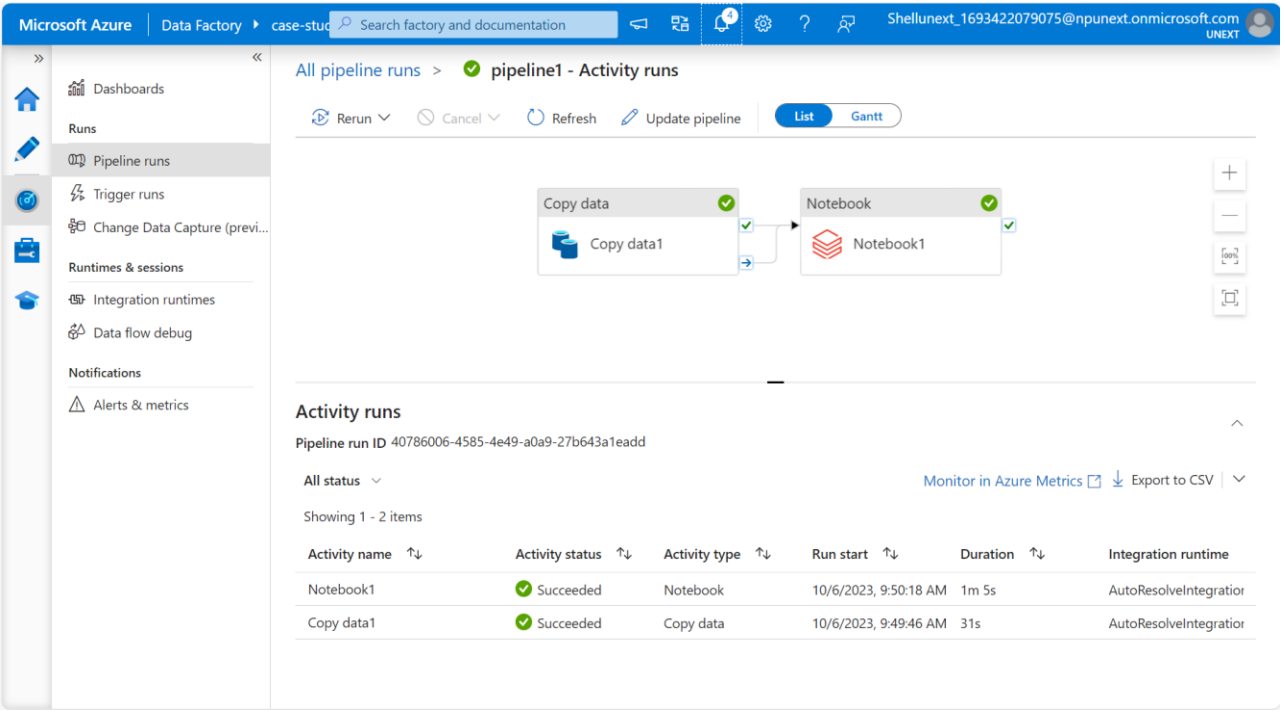


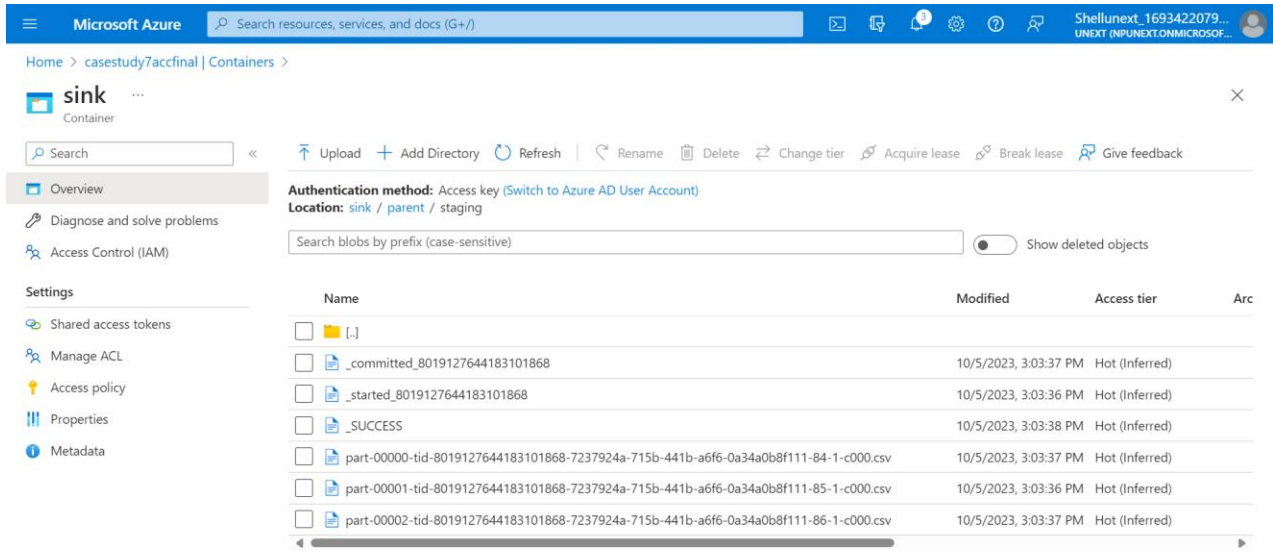
DATABRICKS NOTEBOOK



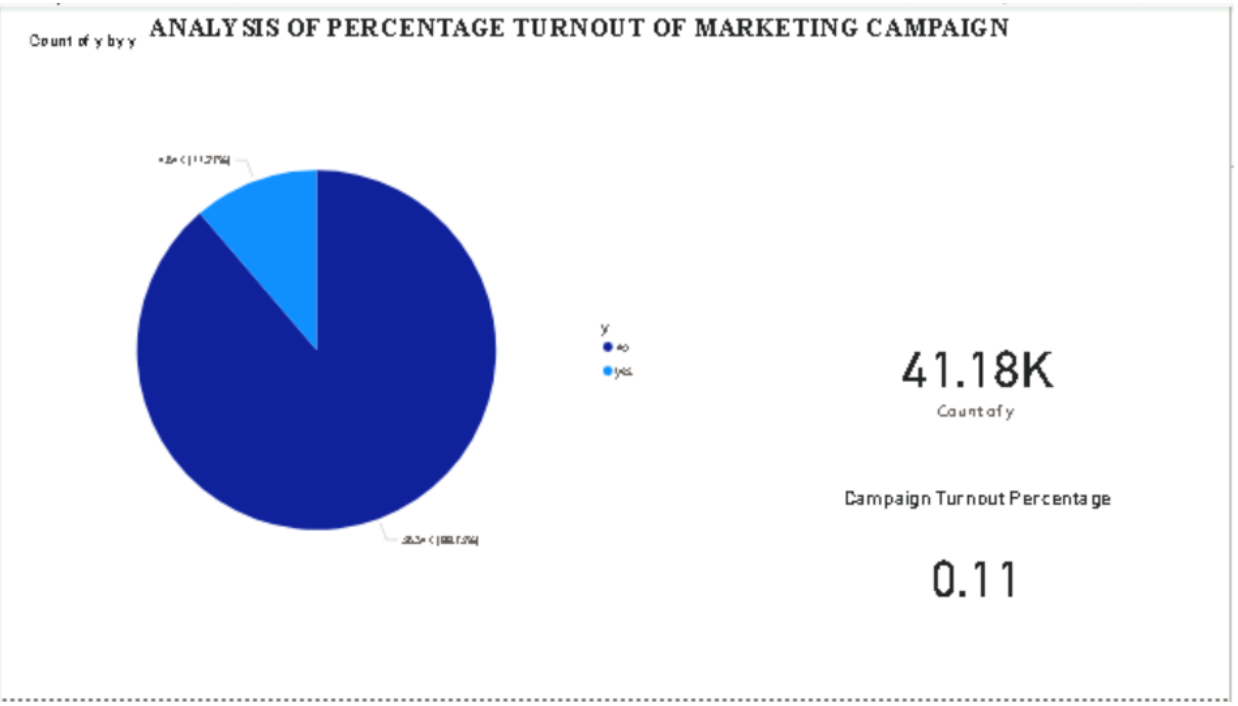


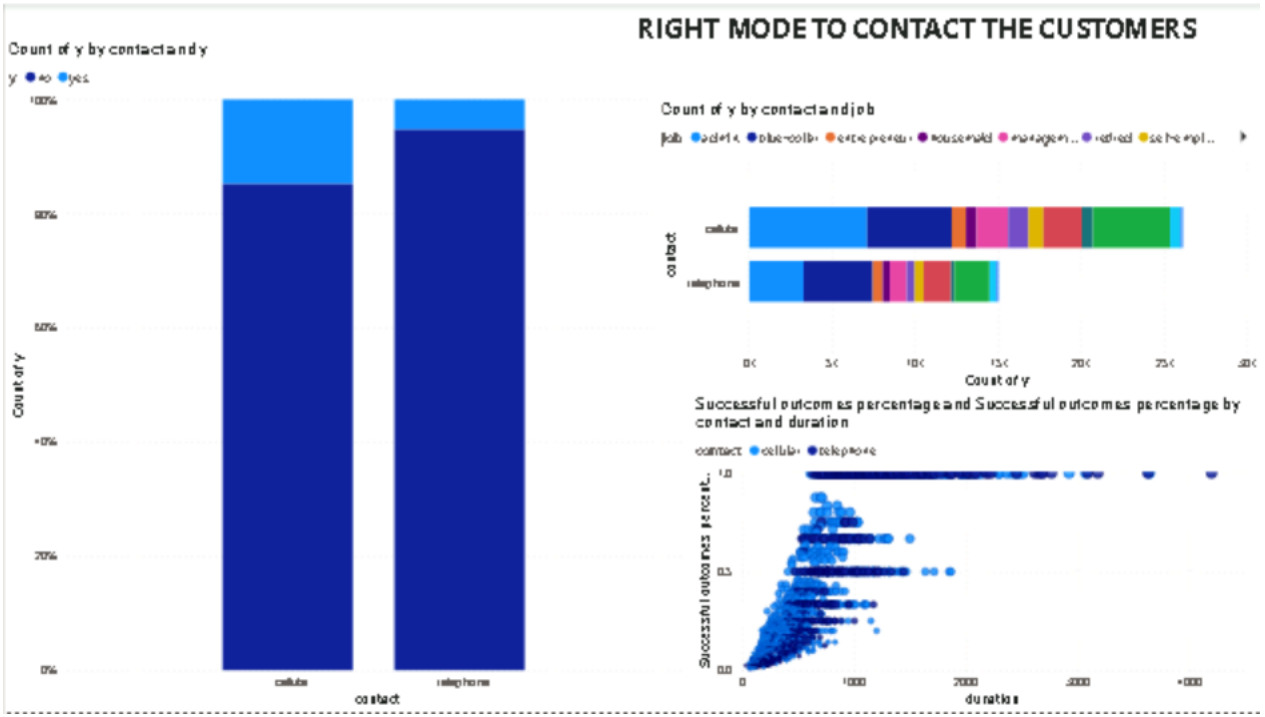
DATA LAYERS SNAPSHOTS



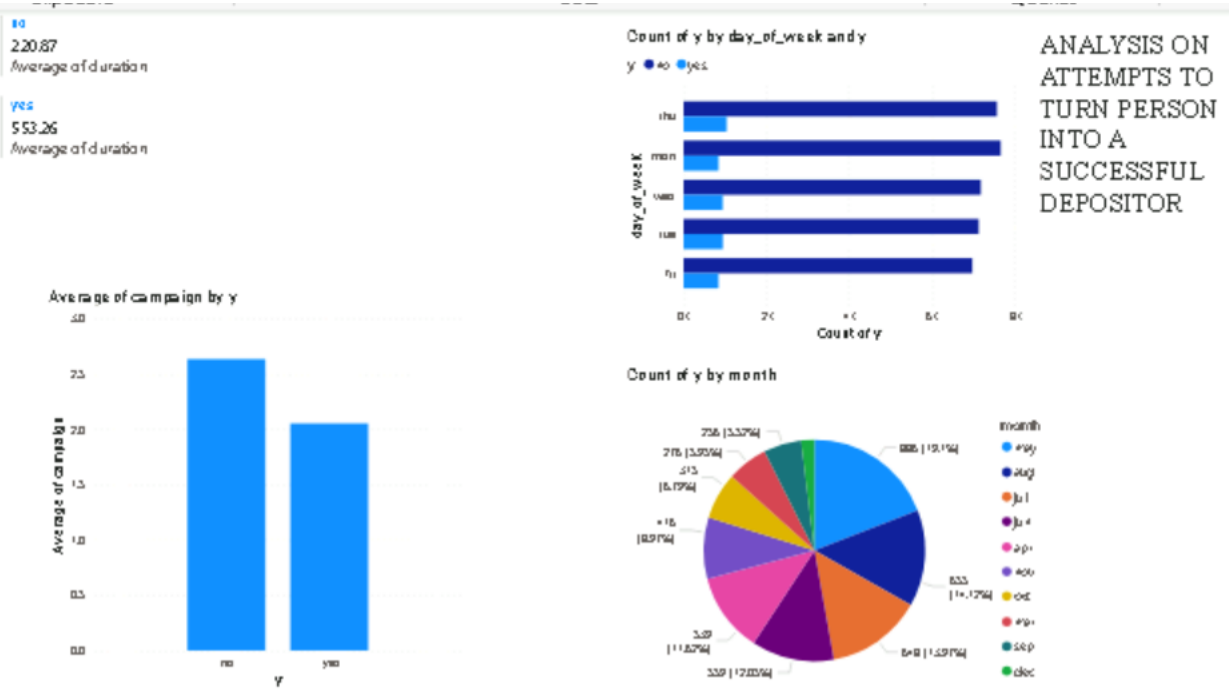


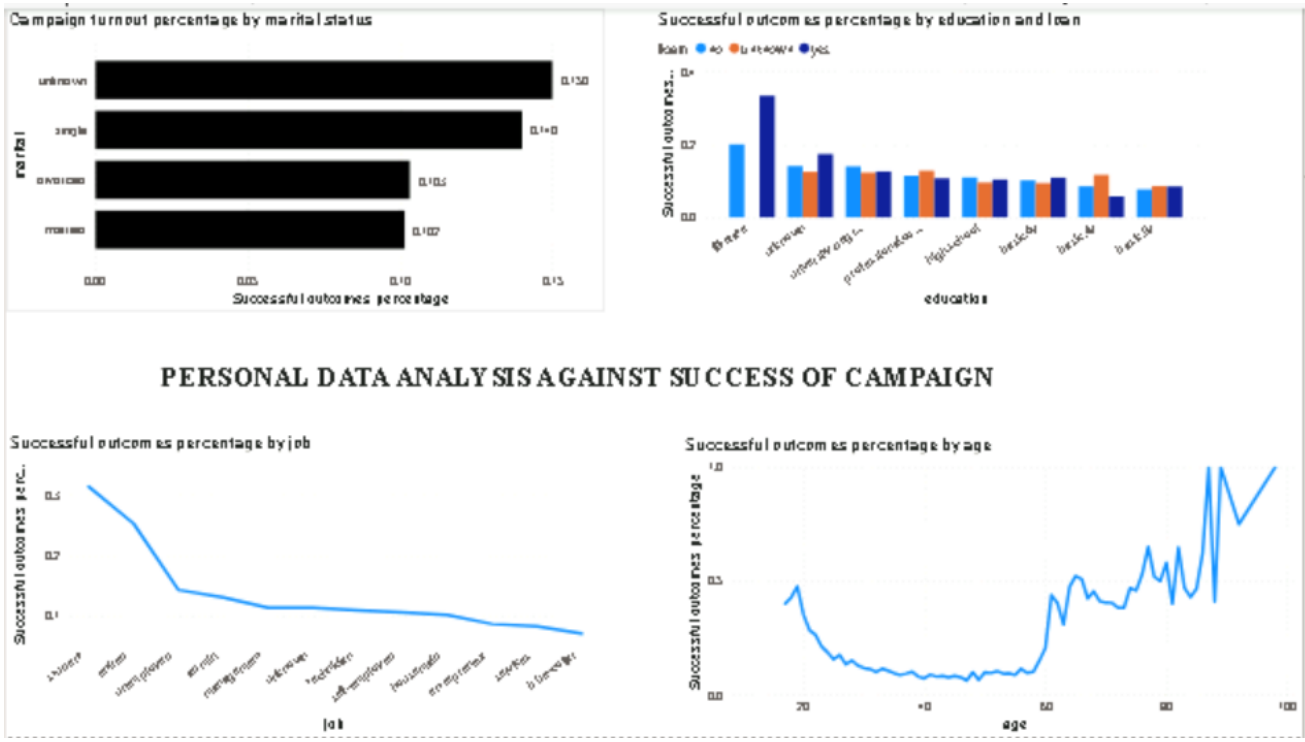
POWERBI DASHBOARD





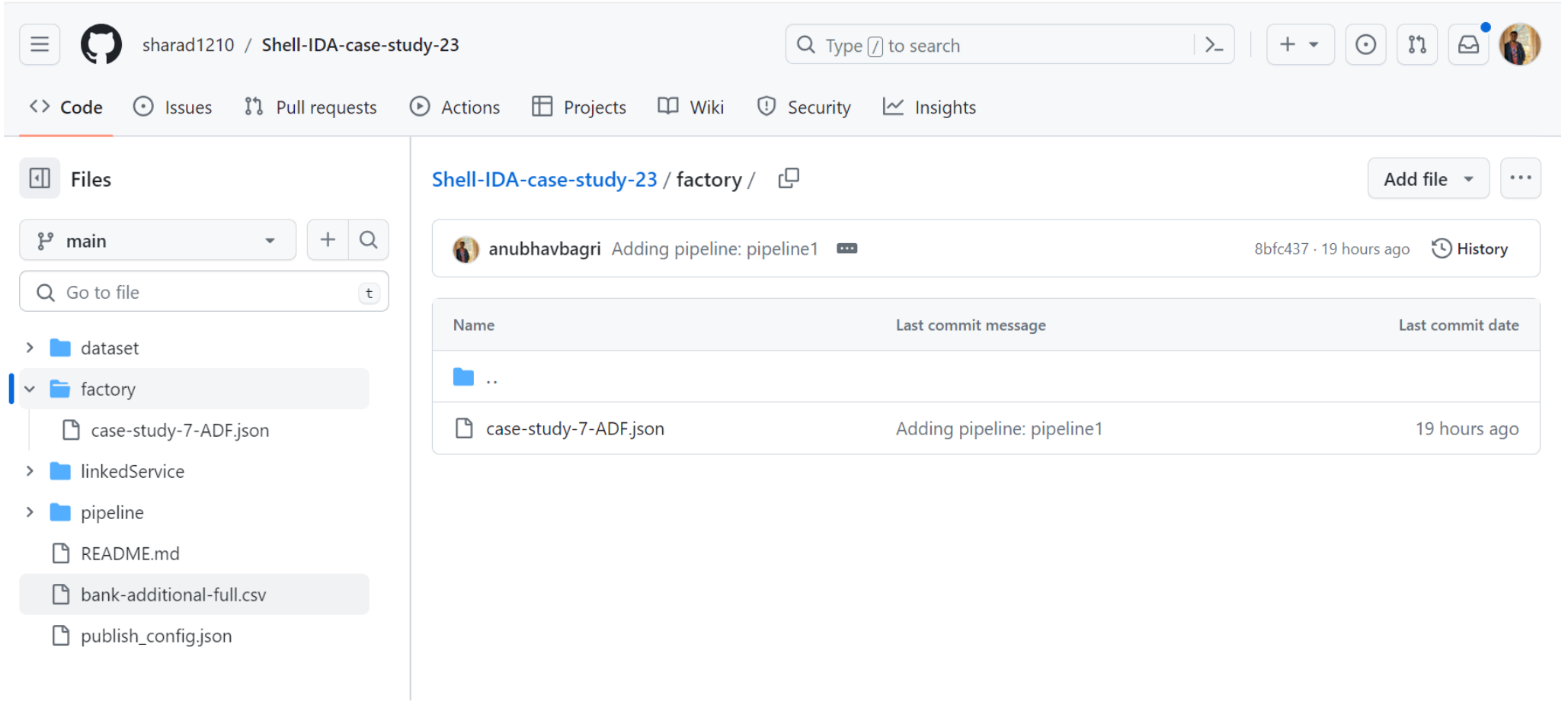
POWERBI DASHBOARD





GITHUB REPOSITORY AND SOURCES

[sharad1210/Shell-IDA-case-study-23:]



CHALLENGES AND LEARNINGS

Challenge: Cost management and cost efficiency.
Learning: In azure we pay for what we use. Different ways to reduce the over expenditure in azure would be to make the right resource selection (selecting resources that aligns with the project need), resource sizing (selecting resource size that matches workload requirements), cost controls like budget alert.

Challenge: Data integrity
Learning: Column names were named in an incorrect way due to which it wasn't working in databricks. The column names were changed replacing the '.' with an '_' for proper functionality.

Challenge: Data Security
Learning: For security purposes we have used access token for mounting the data from azure storage account to Databricks along with permissions to read and write for more secure access. We also created access token for the databricks to transfer the data securely to power bi.

Thank you