1. **How do you connect from Azure to AWS**

<https://youtu.be/TtlKFgfN3PU?si=OyYLFtidSOVxRTNd> – AWS security

<https://youtu.be/r2YO6QSqJog>

<https://youtu.be/_IOZ8_cPgu8?si=njp53iNF9f9bLceu> – Network basics

<https://youtu.be/qGpMQIzGw_w?si=q9tEtMzhwMsYnUVp> - Image of Site to site connection (Azure and AWS)

**Configure Azure**

1) Create the resource group, name : RG-Azure\_AWS, Region : East US

2) create **virtual network**  IP address : [172.10.0.0/16](http://172.10.0.0/16), subnet [172.10.0.0/24](http://172.10.0.0/24), resource group required to create the virtual network

3) create the **Virtual network gateway**, gate way types, VPN, express Route ,VPN type :  route based , policy based (legacy application)

  Resource and virtunal network required to create the VPN gateway

  Subnet can create VPN gateway itself

**Configure AWS**

1) create **VPC**(virtual private cloud)

2) creat subnet [10.10.0.0/16](http://10.10.0.0/16), subnet ip address  [10.10.0.0/24](http://10.10.0.0/24)

3)  **Virtual private network** under that 4,5 and 6 available

4) **customer gateways** , you need to give public IP address of Azure which is created under point 3 in azure

5) **virtual private gateways ,** attach to VPC

6) **site to site VPN** , need to provide the virtual private gateway,customer gateway subnet Azure IP address, we will get two tunals

7) download the **configuration file from VPN connection**

          get the shared key, outside IP address (Cutomer gateway IP and Virtual private dateway IP) , inside of IP address

**Connecting to Azure and AWS**

**These steps in azure**

1) **Local network gateways** , provide the AWS outbound Virtual IP address, which present in the config file

     and address range : 10,10.0.0.6

 2)  Under Virtual network gateway, go to **connections**, click add connections, aws SHARED KEY need to provided and local network gateway

**In AWS**

create internet gateway and attach to VPC

Rote table add the azure IP address [172.10.0.0/24](http://172.10.0.0/24), [0.0.0.0/0](http://0.0.0.0/0)  and internet gateway for internet , allow the other networks

Create the EC2 Instance and ping the IP address of Azure

Under the security of EC2 instance, Security group, add the azure IP address [172.10.0.0/24](http://172.10.0.0/24) , SSH

ICMP or SSH can be added to security group

**2)How do you find data source in cloud ? how do you confirm where the data are coming from**

ANS) When working with data in the cloud, there are a few methods you can use to find the data source and confirm its origin:

**Data Catalogs and Metadata:** Cloud platforms often provide data cataloging and metadata management services. These services help organize and document information about your data sources. By accessing the data catalog, you can obtain details about the data source, such as its location, format, and origin.

**Data Logging and Monitoring:** Cloud services typically offer logging and monitoring capabilities that track data movement and activities. By examining the logs and monitoring data, you can identify the sources of incoming data and track their flow through various services.

**3)how aws connect to on premise with examples**

AWS Direct Connect establishes a dedicated network connection between your on-premises data center and AWS.

You can provision a physical connection by partnering with an AWS Direct Connect location or use a hosted connection via an AWS Direct Connect partner.

This provides a private and dedicated network link with high bandwidth and low latency between your on-premises environment and AWS services.

VPN (Virtual Private Network):

AWS supports VPN connections, allowing you to create a secure connection between your on-premises network and AWS.

You can use a hardware VPN appliance or software-based VPN client to establish an encrypted connection over the public internet.

AWS provides a Virtual Private Gateway that acts as the VPN endpoint on the AWS side, and you would configure the corresponding VPN device or software on your on-premises network.

AWS Transit Gateway:

AWS Transit Gateway simplifies network connectivity between multiple VPCs, on-premises networks, and remote networks.

It acts as a hub that allows you to consolidate your network connectivity and control traffic flow between various environments.

By establishing a VPN connection or using AWS Direct Connect with AWS Transit Gateway, you can connect your on-premises environment to AWS and route traffic efficiently.

AWS Storage Gateway:

AWS Storage Gateway enables you to integrate on-premises applications with AWS storage services like Amazon S3, Amazon EBS, or Amazon Glacier.

It provides a virtual appliance that you can deploy on-premises, acting as a bridge between your local applications and AWS storage.

You can use different modes of the Storage Gateway, such as File Gateway, Volume Gateway, or Tape Gateway, based on your specific storage requirements.

Hybrid DNS Resolution:

**3A) What is your strategy for data migration from on prem to cloud**

**DMS migration**

Create DMS replication instance , which will be used in the endpoint testing

Create source and target database endpoints

Create data migration task

<https://youtu.be/3v3Q7WVbsT4>

RDP – remote desktop protocal

**4)To establish an AWS Direct Connect connection between your on-premises environment and AWS, you need to follow these steps:**

Prerequisites:

Have an AWS account with appropriate permissions to create and manage Direct Connect resources.

Identify the AWS Direct Connect location that is geographically closest to your on-premises environment.

Ensure you have a compatible network device (router) that supports Direct Connect.

Set up the Direct Connect connection in the AWS Management Console:

Log in to the AWS Management Console.

Open the Direct Connect console.

Click on "Create Direct Connect Connection" and provide the necessary information:

Connection name: A unique name for the connection.

Location: Choose the AWS Direct Connect location closest to your on-premises environment.

Bandwidth: Select the desired bandwidth for the connection.

VLAN: Specify the VLAN ID for the connection.

Review and confirm the connection details.

Create a virtual interface:

After creating the Direct Connect connection, you need to create a virtual interface to connect the on-premises network to AWS.

Specify the connection ID of the Direct Connect connection you created.

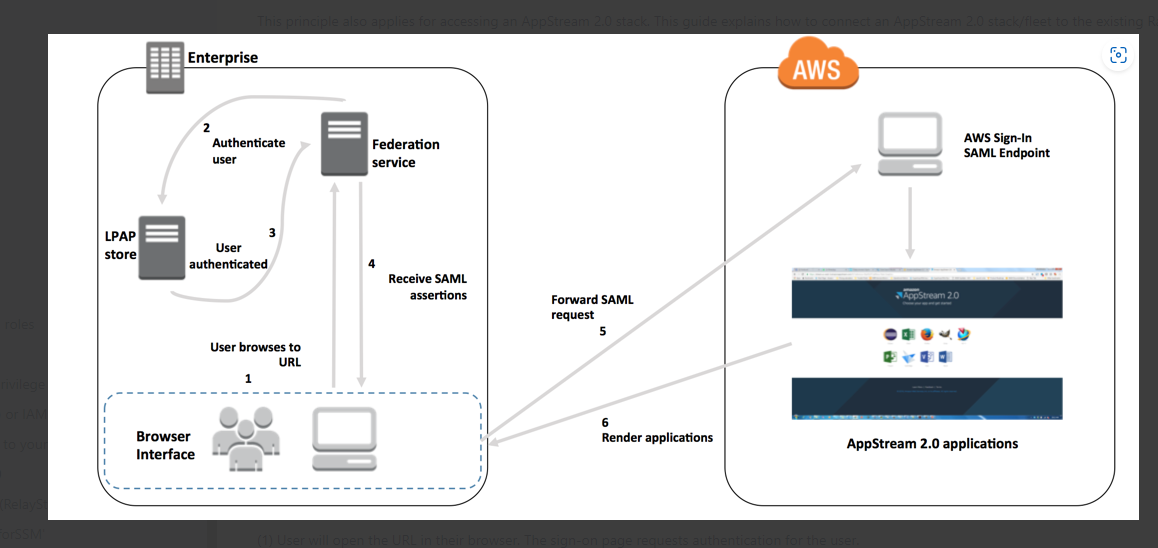
Provide the BGP (Border Gateway Protocol) information for routing between your on-premises network and AWS.

Configure your network device for Direct Connect:

6) How AWS identity and access management, integration with active directory

[AAD SAML Integration for AppStream 2.0 - AWS - Confluence (rabobank.nl)](https://confluence.dev.rabobank.nl/pages/viewpage.action?spaceKey=AWS&title=AAD+SAML+Integration+for+AppStream+2.0)

AWS identity and access management, integration with Azure active directory



1) User will open the URL in their browser. The sign-on page requests authentication for the user.

(2) The federation service requests authentication from the organization's identity store.

(3) The identity store authenticates the user and returns the authentication response to the federation service.

(4) On successful authentication, the federation service posts the SAML assertion to the user's browser.

(5) The user's browser posts the SAML assertion to the AWS Sign-In SAML endpoint (<https://signin.aws.amazon.com/saml>). AWS Sign-In receives the SAML request, processes the request, authenticates the user, and forwards the authentication token to AppStream 2.0.  
Note: If the user's Rabo Account is a member of multiple roles within you AWS account they will be asked to select which role to assume in AWS : instruct them to select the name of the **Appstream** one.

(6) Using the authentication token from AWS, AppStream 2.0 authorizes the user and presents applications to the browser.

S3 bucket access

System manager

1. Create self signed certificate in AWS Certificate Manager (ACM)

Hyper text transfer portal (HTTP – enable client communication with web ser

Local Network Communication: In closed or local network environments, self-signed certificates can be used to secure communication between different devices, services, or applications. This helps ensure that data transmitted over the network remains encrypted and protected

2) In stal and configure ADFS

3) Download federation metadata XML file

AWS

4) Create identity provider and IAM roles in AWS console

 -ARN of SAML provider , ARN of role to assume

5) Configure AWS as trusted replying party in ADFS

6) configure claim rules for the AWS relaying party

Adding name id

 adding a role session name

 adding role attributes (AD groups  & role)

7) Testing the configuration

<https://youtu.be/fyF0cPP0xfo>

1) Configure the active directory service

2) active directory federation services - identity  provider

2) Finally config AWS to trust ADFS

user login using his browser login the active directory web page

it is validate the user credential with local database and send out the same back to the user browser

post the saml assetion request to AWS, which rerun the tmp credential to the browser

 then user able to login the AWS services

SetSPM -a host/localhost adfsrv

self sign cretificate

user access the ADFS site, which run on Https ://

user id autnticate with active directory . user receive teh authentication response

browser post the sample package to end points to AWS

user is redirect to AWS console

**6) how do you achieve anonymization of data especially if the data are personal and are sensitive information in AWS cloud**

Encryption:

Utilize encryption techniques to protect the data at rest and in transit.

AWS provides services like AWS Key Management Service (KMS) to manage encryption keys and AWS Certificate Manager (ACM) for managing SSL/TLS certificates.

**Data Masking:**

Apply data masking techniques to replace sensitive information with fictional or altered data.

AWS services such as Amazon RDS for Oracle or Amazon Redshift support built-in data masking capabilities to help protect sensitive data.

**Tokenization:**

Tokenization involves replacing sensitive data with unique tokens while maintaining referential integrity.

AWS services like Amazon S3 and Amazon DynamoDB can be used with tokenization techniques to preserve data privacy.

**Data Lifecycle Management**:

Implement data lifecycle management policies to control data retention, archival, and disposal.

AWS services like Amazon S3 and Amazon Glacier offer lifecycle management features to automate data retention and archival processes.

**Identity and Access Management (IAM**):

Implement proper IAM policies and roles to control access to sensitive data.

Use AWS IAM features like fine-grained permissions, multi-factor authentication (MFA), and temporary security credentials to limit access to sensitive data.

**Data Minimization:**

Reduce the amount of personal and sensitive data stored or processed in the cloud.

Implement data minimization practices by only collecting and retaining the necessary data for business purposes.

**Compliance and Governance:**

Leverage AWS services and features that assist with compliance and governance requirements.

AWS provides services like AWS CloudTrail for audit logging, AWS Config for configuration management, and AWS CloudWatch for monitoring and alerting.

Privacy by Design:

Adopt privacy by design principles to ensure that privacy considerations are incorporated into the design and implementation of your AWS architecture.

Evaluate and implement AWS services and features that align with privacy and data protection requirements.

**7) Why do like to Join IHIS**

Ans) looking for long term opportunity in Singapore

**8) What frustrates you ?**

Ans) I will not be frustrated. Most of the time I will maintain the transference with the team and the management. Its always help me to maintain the positive environment

**9) How to trouble shoot the auto scaling issues**

Auto scaling service is available in AWS

1)check your scaling policies to see whether an event triggers more than one policy

2)vefiry if a scal out policy and a scale in policy are triggered at the same time

3)check auto scaling group already reached its minimum or maximum number of instances

4)check instances are in a cooldown period and instance warmup period

5)check if there is a lifecycle hook configured for your amazon ec2 auto scaling group

6)Check is there is any schedule action configure which conflicting the auto scaling group

**7) Check the lifestyle hooks and update the package – Nothing to update the package**

8)Check suspended processes for your auto scaling group

**10)How to troubleshot the EC2 instance**

[**https://youtu.be/ZXQObhB7gBA**](https://youtu.be/ZXQObhB7gBA) **- AWS troubleshooting**

[**https://youtu.be/aRDewRsoWVQ**](https://youtu.be/aRDewRsoWVQ) **- Glue optimization**

<https://youtu.be/vdBsfbo1Pik?si=5z6CNpwh1IL73uD5> - DR setup

[AWS troubleshooting | AWS ELB troubleshooting | AWS ALB issues | AWS issues | Video-2 - YouTube](https://www.youtube.com/watch?v=tfXbPeenNHQ&list=RDCMUCoTtHeJZfbire-Akqk2Vapg&index=2) - ELB

Security group status

Instance current status

Ec2 instance status check – 2/2

Under the monitor and troubleshoot - Get instacnescreenshot

System logs

Restore from latest snapshot

Check cloud trail – any recent changes to the instance

Security 🡪 Network ACLS (network access control list ) - Inbound rules are missing

Internet gateway – default internet attached to EC2 or not

Route table

**11) How to troubleshot the S3 instance**

1. Check the IAM user policies – amazonS3fullaccess policy attached to the user
2. Storage S3 full acess provided
3. Check the bucket has any custom policy
4. Update the custom policy

Access is denied for specific IP range

**12) Troubleshot the Lamda function**

[**https://youtu.be/ZXQObhB7gBA**](https://youtu.be/ZXQObhB7gBA) **- AWS troubleshooting**

* Check the correct AWS event bridge trigger properly configured mapped to the EC2 instance
* Check the **event trigger** configured with right Json or not
* Check the lamda code locally and confirm source code has any errors
* Check the lamda configuration setting all are configurated properly
* Check on lamda cloudwatch log group
* Cloud trail has been enabled for the region where lamda has been configured
* Amazon **Event Bridge ( ASW services)** event driven application at scale

**AWS Workspace**

AWS Workspace (nothing but create the virtual servers)

End user computing 🡪 Workspaces 🡪 setup the directory

Launch workspace , create the login user, one workspace can use update to 20 users

User get the email to setup the user name and password , select the Windows or Linux and enter the rigistation code which you will get via email

**Load balance real time issues**

Application load balancer showing 502 bad gateway

Check the underline EC2 instance is executing or not

Check Netwrok ACL has any blocking IP address

Load balancer health check configuration need to be verified

Restart the application service inside of backend EC2 instance

**13) Jupyter notebooks and Sagemaker notebooks**

Go to Amazon SagaMaker 🡪 go to notebook🡪 instance 🡪 Enter the name and IAM role, instace details

Can schedule and write and execute the code

**Disaster recovery**

RTO \_ restore time objectives – 30 min

RPO – Restore point objects

AppStream

------------

1. Go to Images

Go to Image builder - Lunch the image buildegvr – subnet, security, iAM profile, VPC

Run the image which you created in the above step

1. Adminisrator - go to Image assistance

**Add application**

1. Fleets – configure the image, much capity , on demand, subnet
2. Stacks - Fleet tag to stacks - generate the URL for ap

**Network Main componeds**

IP

SUBNET - You can deploy AWS resources, such as EC2 instances, RDS databases, and Lambda functions, within specific subnets. This helps isolate resources and control their network access and traffic flow

Gateway (ROuter) 🡪 to connect between LAN and WAN we willwewe

Main parts of Router

NAT - network address translation

DMZ- DE militarize zone - it allowed specific IP to communicate between LAN and WAN

Firewall - stop communicate with LAN and WAN

Port forwarding – Outside network Wide area network can talk to Local area network

<https://youtu.be/_IOZ8_cPgu8>

what is the purpose of Cloudera ? where to use and what ?

ChatGPT

Cloudera is a company that provides a distribution of Apache Hadoop and related open-source software, along with additional enterprise-grade tools and services. Cloudera's main offering is the Cloudera Data Platform (CDP), which is designed to enable organizations to effectively manage and analyze large volumes of data across on-premises, cloud, and hybrid environments.

The purpose of Cloudera and CDP is to provide a comprehensive data management and analytics platform that allows businesses to process, store, and analyze big data at scale. Here are some key components and use cases of Cloudera:

1. Apache Hadoop: Cloudera leverages the power of Apache Hadoop, an open-source framework that enables distributed storage and processing of large datasets across clusters of computers. Hadoop's core components, such as the Hadoop Distributed File System (HDFS) and the MapReduce processing framework, form the foundation of Cloudera's data platform.
2. Data Storage and Processing: Cloudera provides a scalable and fault-tolerant data storage layer through HDFS, allowing organizations to store and process vast amounts of structured, semi-structured, and unstructured data. This enables data ingestion, transformation, and batch processing of data for various analytics use cases.
3. Data Warehousing: Cloudera offers tools like Apache Impala (incubating) and Apache Hive for interactive SQL querying and data warehousing. These tools allow users to run ad-hoc queries and perform analytics on large datasets stored in Hadoop, providing a familiar SQL interface for data exploration and reporting.
4. Data Engineering: Cloudera provides tools and frameworks for data engineering tasks such as data integration, ETL (Extract, Transform, Load), and data pipeline management. Tools like Apache Kafka and Apache NiFi help with real-time data ingestion, while Apache Sqoop facilitates data transfer between Hadoop and traditional relational databases.
5. Advanced Analytics and Machine Learning: Cloudera supports advanced analytics and machine learning workloads through integration with tools like Apache Spark, Apache HBase, and Cloudera Data Science Workbench. These tools enable data scientists and analysts to perform data exploration, develop machine learning models, and extract valuable insights from large datasets.
6. Data Security and Governance: Cloudera provides features and capabilities for data security, governance, and compliance. These include authentication, authorization, data encryption, auditing, and fine-grained access controls, ensuring data privacy and compliance with regulatory requirements.

Cloudera's platform is used across various industries and use cases, including cybersecurity, fraud detection, healthcare analytics, financial services, IoT analytics, and more. It empowers organizations to harness the power of big data and derive valuable insights to drive business decisions, improve operational efficiency, and gain a competitive edge.

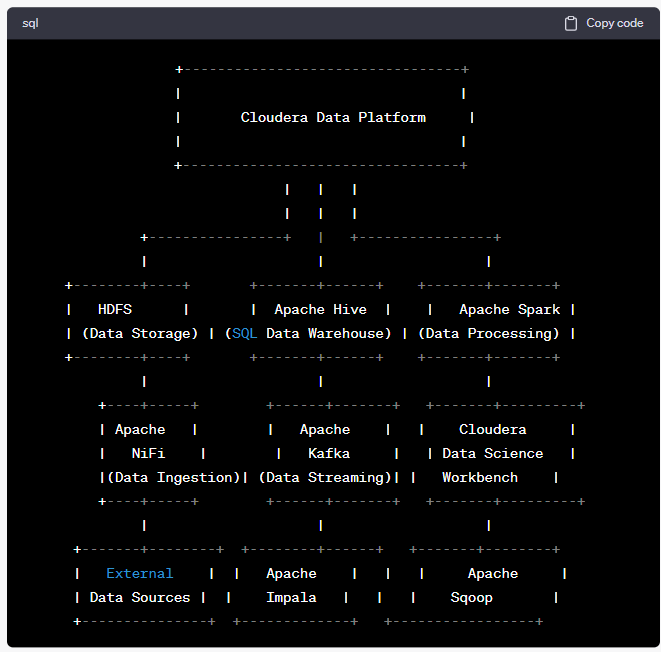
any diagram ?

ChatGPT

Unfortunately, as a text-based AI model, I cannot directly provide visual diagrams. However, I can describe a high-level diagram illustrating the components and interactions in a Cloudera-based architecture.

Here's a textual representation of a typical Cloudera architecture:

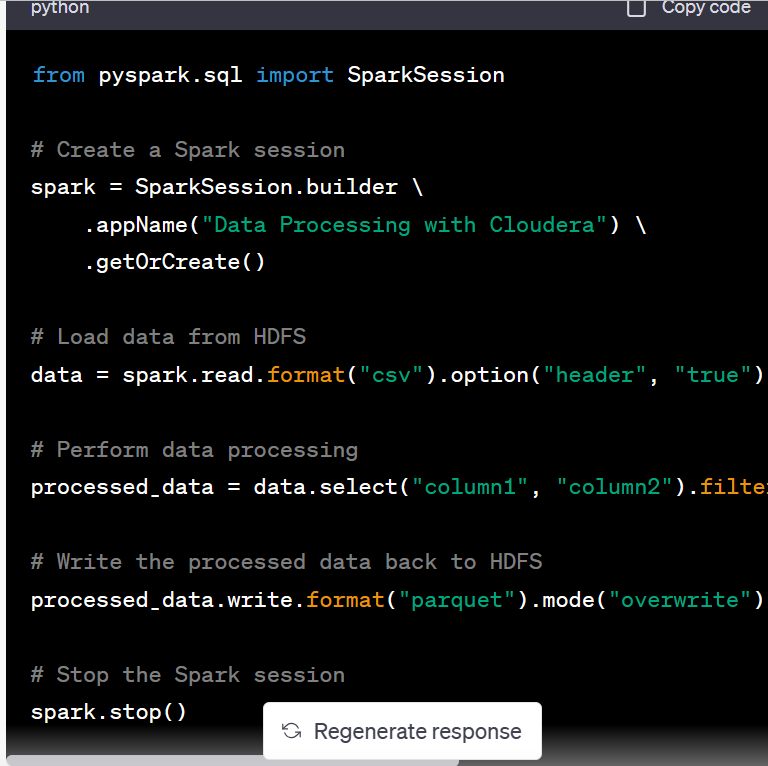
In this diagram:



* The Cloudera Data Platform (CDP) serves as the core component, providing management and coordination for the entire platform.
* The Hadoop Distributed File System (HDFS) is the underlying distributed file storage system, used for storing and managing large datasets.
* Apache Hive and Apache Impala provide SQL interfaces for data warehousing and interactive querying on the data stored in HDFS.
* Apache Spark is used for distributed data processing and analytics, supporting batch processing and machine learning workloads.
* Apache NiFi and Apache Kafka facilitate data ingestion, streaming, and integration from various external data sources into the Cloudera platform.
* Cloudera Data Science Workbench provides an environment for data scientists to develop and execute machine learning models and advanced analytics.
* Apache Sqoop enables data transfer between Hadoop and traditional relational databases.

Please note that this diagram is a simplified representation, and in actual implementations, there can be additional components, services, and interactions based on the specific requirements and architecture of the Cloudera deployment.

This is just a basic example to illustrate the usage of Apache Spark with Cloudera's platform. You can build upon this code to include more complex data processing operations, additional transformations, and integrations with other Cloudera services or tools as needed.



This is just a simple example to demonstrate the usage of HiveQL with Cloudera's platform. You can expand on this code by creating more complex table structures, defining partitions, performing joins, aggregations, and other SQL operations supported by HiveQL.



**NTUC Income**

1) What will be your suggest for Pull or push mechanism of the data

Ans) Evaluate the options based on factors such as **data size, transfer speed, security, and cost**.

1. **Amazon S3 Transfer Acceleration**: This method is suitable for **large data transfers** over

long distances, reducing the time taken to upload or download data to/from Amazon S3.

1. **AWS Snowball:** This is a **physical data transport** solution for large-scale data transfers. You request a Snowball appliance from AWS, which is a secure storage device.
2. **AWS DataSync:** This service is designed for **online data transfer** between on-premises storage and AWS. It can be used for **one- time migrations** or for continuous replication.
3. **AWS Direct Connect:** This method is used for establishing a **dedicated network** connection between your on-premises infrastructure and AWS. With Direct Connect, you can transfer data directly over a private network connection, bypassing the public internet, which can provide increased security, reliability, and lower latency.

2) How to do you decide which tool is best

1)**Integration and interoperability:** Assess how well the tool integrates with your existing systems,

2)**Security and Compliance:** Assess the security measures, compliance certifications, and data protection capabilities of both Azure and AWS

3)**Cost and licensing:** Evaluate the pricing model, licensing terms, and any additional costs associated with each tool.

4)**Proof of concept or trial:** If possible, conduct a proof of concept or trial period to test the tool in a controlled environment

5)**Assess performance and scalability:** Consider the performance benchmarks, scalability, and resource requirements of each tool

6)**Consider ease of use and user experience:** Evaluate the user interface, ease of installation, conf

3) what are thinking will be covered as part of data governance

1. what is the data lineage

Data lineage refers to the ability to track and understand the origin, movement, transformation, and destination of data as it flows through various systems, processes, and transformations within an organization.

**Impact Analysis:** When changes are made to data sources, transformations, or systems, data lineage helps identify the potential impacts on downstream processes, reports, or analytics.

**Data Quality Management:** By understanding the data lineage, organizations can identify data quality issues, track their origins, and implement corrective actions.

1. how do you do the data quality

1)**Data Profiling:** Conduct data profiling to understand the current state of your data. Analyze the data to identify patterns, and data quality issues.

2)**Data Cleansing:** Implement data cleansing processes to address data quality issues. This involves performing actions such as removing duplicate

3)**Data Validation and Verification:** Implement data validation checks to ensure that data meets predefined rules, constraints, or business logic.

4) **Data Governance:** Establish data governance processes and practices to enforce data quality standards

6) what kind of tool you used for data governance

Ans) Collibra

**Compliance and Regulatory Support:** Collibra helps organizations comply with data privacy regulations and other compliance requirements

**Integration Capabilities:** Collibra offers integration capabilities with various data sources, data management tools, and data processing platforms. It can integrate with data integration tools, data quality tools, metadata repositories, and data lineage tools.

**Data Quality Management:** Collibra includes data quality management features to assess, monitor, and improve data quality.

**Data Lineage**: Collibra provides data lineage capabilities that allow organizations to track and visualize the lineage of their data

**Data Catalog:** Collibra offers a data cataloging feature that allows organizations to create a centralized inventory of their data assets.

**Data Governance Framework:** Collibra enables organizations to establish and enforce data governance frameworks and policies

**BNP paribas**

1. What is the difference between Datawarehouse and Data lake
2. How do you Extract, Load, Transform the unstructured structure data

Extraction – use AWS kinesis or API to extract the data or Python script

Storage : Amazon S3

Load the unstructured data to the data lake use the Apache spark

Transformation : Add metadata and structure to the data , use tools like Apache Hive, spark SQL

1. What is difference between, technical, application and data Architect

Application Architect:

An application architect is responsible for designing and developing software applications. They work with stakeholders to identify business requirements and then create an architectural plan to meet those requirements. They focus on the overall structure of the application, including its user interface, components, and integration with other systems. They also oversee the development team to ensure that the application is developed according to the architectural plan.

Data Architect:

A data architect is responsible for designing and managing the organization's data architecture. They work closely with stakeholders to identify the organization's data needs and then create a plan to meet those needs. They focus on data modeling, database design, and data security. They also oversee the implementation of data-related projects, ensuring that they are developed according to the data architecture plan.

Technical Architect:

A technical architect is responsible for the technical design and implementation of software systems. They work with stakeholders to identify technical requirements and then create a plan to meet those requirements. They focus on the selection of hardware and software technologies, system integration, and performance optimization. They also oversee the development team to ensure that the system is developed according to the technical architecture plan.

In summary, an application architect focuses on the design and development of software applications, a data architect focuses on the organization's data architecture, and a technical architect focuses on the technical design and implementation of software systems.

1. **what is the difference between tableau, power BI and business objects which is best tool**

Tableau, Power BI, and Business Objects are all popular business intelligence tools used for data visualization and analysis. Here are some differences between these tools:

User interface: Tableau has a very intuitive and **user-friendly** interface, which makes it easy for users to create and share visualizations. Power BI and Business Objects have **more complex interfaces** and may require more training to use effectively.

Data integration: **Power BI is tightly integrated with the Microsoft ecosystem, making it easy to connect with other Microsoft tools such as Excel and SQL Server**. Business Objects is part of the SAP ecosystem and is designed to work with SAP data sources. Tableau has broader support for different data sources, including cloud-based data warehouses like Amazon Redshift and Google BigQuery.

Pricing: Power BI has a low-cost entry point with a free version and affordable pricing for the full version. **Tableau** and Business Objects are generally **more expensive**, especially for larger organizations with many users.

Advanced features: **Tableau is known for its advanced visualization capabilities and flexible dashboards,** which allow users to create highly customized visualizations. Power BI has strong integration with machine learning tools and is well-suited for predictive analytics. Business Objects is known for its robust reporting capabilities and integration with SAP's enterprise resource planning software.

Ease of deployment: Power BI is a cloud-based tool, which makes it easy to deploy and manage. Tableau and Business Objects can be deployed on-premises or in the cloud, but may require more IT support to manage.

In terms of which tool is best, it ultimately depends on your organization's specific needs and priorities. All three tools have their strengths and weaknesses, and the best tool for you will depend on factors such as your data sources, budget, and user requirements.

1. **A data architect document typically covers**

**Data Governance:** An overview of the data governance framework, policies, and processes and security of data across the organization.

**Data Modeling:** Guidelines and best practices for data modeling, including the use of conceptual, logical, and physical data models

**Data Integration:** Strategies and approaches for integrating data from various sources, systems, and applications.

**Data Storage and Infrastructure:** Considerations and recommendations for data storage and infrastructure,

**Data Security and Privacy:** Policies, practices, and guidelines for ensuring data security,

**Data Quality Management:** Strategies, processes, and tools for assessing, monitoring, and improving data quality

**Data Analytics and Reporting**: Guidelines for enabling data analytics and reporting capabilities, including data visualization

**Data Migration and Conversion**: Strategies and best practices for migrating and converting data from legacy systems to new platforms

**Data Lifecycle Management**: Considerations for managing the lifecycle of data, including data retention policies, data archiving,

**Metadata Management:** Strategies and tools for capturing, managing, and leveraging metadata,

**Data Governance Roles and Responsibilities:** Description of the roles and responsibilities of different stakeholders involved in data governance

**Data Architecture Standards and Guidelines:** A set of standards, guidelines, and best practices

1. What are different type of the files ( Parket, Avro, ORC ) which one use when
2. What are the dimensions and Facts
3. Is static data dimension or fact ?
4. What are parameters will be consider for data injection ?
5. How do you do the data visualization
6. How do you migrate the report from the BO to Power BI

Develop the reports newly

1. If you have same filed available in 10 tables, how do you extract that filed in your report
2. Is SCD Type2 can do it data lake ? Ans) can’t, that is the why Dataware house is required

**OCBC**

1. How do you do the impact analysis when you have 50 data marts ? do you have any framework
2. How do you handle invalid data ? is any process
3. How do you do the GL reconciliation
4. When you have a loan entry which are tables get entry
5. What is B-Key and B-MAP
6. Where do you check how many records inserted, updated, is any framework
7. How to do you do the MLoad using the BODS
8. Which ETL tool used for Bigdata
9. Do you have any error handling framework

**OBIEE**

1. How to view the SQL in OBBIE

**Linux**

1) how to find count of file in Linux

1. how to Unzipe the file

Linux

zip –r zippeddocs.zip path of the file

unzip zippeddocs.zip

Unix

Zip the file : GZIP filename

View the Zip file : ZCAT filename

Unzip the file : GUNZIP filename

Diskspace : df –k

Diskusage : df –h

File \*

1. Copy only top 10 records
2. How to stop the job in Linux
3. How to find the long running job in Linux

6) difference between RM and RMR

**Hadoop**

2) How to create empty file in Hadoop

Create text fie in local directory and load to Hadoop

HFS –df –put textfile.txt /mydata/textfolder/

1) What is fetch in sqoop

1) How to concet hadoop using tableau

2) how to do the parallel extract in sqoop ( 200t ables) oozie

3) how you encrypt the password in sqoop

4) How to grant the access to the user

5 ) how to see the version number in haddop and tableaue

Hadoop Version

**Tableau**

2) what is project leader role

3) what is difference between system administrator and Site administrator

4) how to set the user access in table

5) Difference Parameter and Set

5) when will use the set

6) what is mean by site

7) How many tables can join – 32

8) help version

1. Right-click on the TWBX and choose 7-Zip->Open Archive.
2. Right-click on the .twb and choose View.
3. Find the <workbook> tag and **version** attribute.
4. Close the window.

9) where we can see the SQL

**Oracle**

* 1. What is replication in Oracle

2) How to store jpg files in oracle , data type

3) how to replace alpha bet in oracle

1. REPLACE(your\_column, CHR(13), ' ')

The TRANSLATE function is similar to REPLACE, but it allows you to replace multiple characters at once, in one function

TRANSLATE(your\_column, CHR(10) || CHR(13, '  ')

You can also use the REGEXP\_REPLACE function to replace special characters.

1. SELECT REGEXP\_REPLACE(your\_column, '[^0-9A-Za-z]', '')
2. FROM table;

4) what is difference between case and decode

5) what is mean by raw data type

6) what is difference between Translate and replace

8) varchar and varchar2 difference

**AWS Glue performance steps**

How to do the AWS Glue performance tunning , explain with the example

ChatGPT

Performance tuning in AWS Glue involves optimizing your ETL (Extract, Transform, Load) jobs to run efficiently and quickly. AWS Glue offers several ways to enhance job performance, and here's a step-by-step guide with an example to help you understand how to do it:

**Example Scenario**: Let's say you have a Glue ETL job that reads data from an S3 bucket, performs transformations, and writes the results to another S3 bucket. You want to improve the job's performance.

**1. Data Partitioning**:

* **Scenario**: Your data is stored in S3 and is partitioned by date. Each partition represents a day's worth of data.
* **Action**: Modify your ETL job to take advantage of partitioning. Use partition filters to read only the data relevant to your processing. For example:

goCopy code

```python

makefileCopy code

datasource0 = glueContext.create\_dynamic\_frame.from\_catalog(database = "your-database-name", table\_name = "your-table-name", transformation\_ctx = "datasource0") # Filter to process only data for a specific date (partition) filtered\_data = Filter.apply(frame = datasource0, f = lambda x: x["date\_column"] == "2023-09-25") # Perform your transformations on the filtered data # ... # Write the results back to the same S3 partition glueContext.write\_dynamic\_frame.from\_catalog(frame = transformed\_data, database = "your-database-name", table\_name = "your-table-name", transformation\_ctx = "datasink0") ```

* **Benefits**: Partitioning reduces the amount of data processed, improving both speed and cost efficiency.

**2. Data Compression**:

* **Scenario**: Your data is in a text format, but it's large and takes a long time to process.
* **Action**: Compress your data using a more efficient format like Parquet or ORC. Modify your ETL job to read and write data in the compressed format. For example:

goCopy code

```python

perlCopy code

datasource0 = glueContext.create\_dynamic\_frame.from\_catalog(database = "your-database-name", table\_name = "your-table-name", transformation\_ctx = "datasource0") # Compress the data into Parquet format compressed\_data = glueContext.create\_dynamic\_frame.from\_catalog(database = "your-database-name", table\_name = "your-table-name", format = "parquet") # Perform your transformations on the compressed data # ... # Write the results back to S3 in Parquet format glueContext.write\_dynamic\_frame.from\_catalog(frame = transformed\_data, database = "your-database-name", table\_name = "your-table-name", format = "parquet", transformation\_ctx = "datasink0") ```

* **Benefits**: Compressed data reduces storage costs and speeds up data read and write operations.

**3. Parallelism**:

* **Scenario**: Your ETL job is running on a large dataset but only uses a single worker.
* **Action**: Increase the number of workers in your Glue job configuration. This allows multiple tasks to run in parallel. For example, in the AWS Glue console, navigate to your job and increase the "Number of workers" in the job configuration.
* **Benefits**: Parallel processing can significantly reduce job execution time.

**4. Caching and Memoization**:

* **Scenario**: Your job performs expensive data lookups or calculations that are repeated for each record.
* **Action**: Use Glue's caching and memoization capabilities. For example, you can cache lookup tables in memory or use the **Map.apply** function to memoize results of expensive computations.

goCopy code

```python

pythonCopy code

from pyspark.sql.functions import udf from pyspark.sql.types import StringType @udf(StringType()) def expensive\_calculation(x): # Perform expensive calculation return result # Apply memoization to the expensive calculation memoized\_data = Map.apply(frame = transformed\_data, f = lambda x: (x, expensive\_calculation(x))) ```

* **Benefits**: Caching and memoization reduce redundant computation, improving job performance.

**5. Monitoring and Profiling**:

* **Scenario**: You're unsure where bottlenecks in your job are occurring.
* **Action**: Use Glue's built-in monitoring and profiling tools. Monitor job metrics, check the job logs, and analyze execution plans to identify and address performance issues.
* **Benefits**: Real-time monitoring and profiling help you identify and resolve performance bottlenecks.

Remember that performance tuning is an iterative process. You should continuously monitor and profile your ETL jobs to identify areas for improvement and apply optimizations as needed. AWS Glue provides a range of tools and options to help you fine-tune your ETL workflows for optimal performance and cost-efficiency.

Top of Form

Regenerate

S3 🡪 AWS Catalog 🡪 developer end point🡪 Segemaker

Developer endpoint🡪 it is one of the method to use to sage maker notebook to Execute the code line by line

sagemaker

Create Spark library and context

DF = Gluecontext.create\_dynamic\_frame.**from\_catalog**( database = XXX, table\_name = ‘XXX)

push\_down\_predicate – it is used to bring only required data to the memory

* 1. Parallel Reads

Use hashexpression/Hashfield which does even distribution of values to spread the data between partitions

Create\_dynamic\_frame.**From\_option method**

In the SQL if you include hashfiled : Region ( partition column) hash partitions : ‘5’ , it will read the data five partition queries

**AWS Glue main componnets**

Data catalog

Databases

--Tables

-- Connection

Crawlers

* Classifiers

Settings

Amazon Kinesis 🡪 delivery streams

Kinesis SDK and Kinesis proceduere library - put the data in kinesis data stream

Create data stresm 🡪 Lamda function to pickup and load to DynoDB

AWS cloud 9🡪 Amazon kinesis data streams – Amazon kinesis data firehouse 🡪 – Amazon S3🡪 lamda for transformation 🡪 AWS Glue 🡪 Amazon Athena

Cloud formation - stacks

Lamda – function

**AWS code commit**

<https://youtu.be/e90rz62rk-8>

1. Required IAM user, code policy associated with this user
2. HTTPS git credentials for AWS code commit
3. Main components

* Repository, build, deploy

1. Under the repositories

Code

Pull requests

Commits

Branches

1. Clone HTTPS to connect the local machine Using CLI

CLI commands

Git Clone “URL” , enter the IAM credentials

Git status - check the status

Git add - a add the contents

Git commit – m ‘’upload the continent”

Get Push - Push to code commit

1. Add the Application, like EC2 instance
2. Create pipelines to push the code to EC2 instance

Create branch

Git checkout -b << Branch Name >> (manage branch )

Git push – set upstream origin my- feature

Create pull request

REST, PUT, and GET are HTTP methods used in API design. Each serves a specific purpose, and they have different characteristics:

1. **GET**:
   * **Purpose**: Used to retrieve data from a specified resource.
   * **Idempotent**: Yes. Repeated GET requests to the same resource should have no side effects.
   * **Safe**: Yes. GET requests should not change the server's state.
   * **Usage**: It's commonly used for read-only operations, such as fetching data, viewing a webpage, or accessing a resource's current state.
   * **Parameters**: Data is usually sent as query parameters in the URL.
2. **PUT**:
   * **Purpose**: Used to update a resource or create it if it doesn't exist at a specified URI.
   * **Idempotent**: Yes. Repeated PUT requests to the same resource should have the same result.
   * **Safe**: No. PUT requests typically modify server state.
   * **Usage**: PUT is used to update an existing resource with new data. It replaces the resource at the specified URI with the provided data.
   * **Parameters**: Data is usually sent in the request body as the updated representation of the resource.
3. **REST** (Representational State Transfer):
   * **Purpose**: REST is not an HTTP method but an architectural style that uses HTTP methods (including GET and PUT) to interact with resources. It emphasizes stateless communication and resource-based URLs.
   * **Idempotent**: GET, PUT, DELETE (another HTTP method), and even POST (with specific conditions) can be idempotent in a RESTful design.
   * **Safe**: GET is considered safe in REST. Other methods may or may not be safe depending on how they are implemented.
   * **Usage**: RESTful APIs use a combination of HTTP methods to create, read, update, and delete resources. They follow a stateless client-server model and often use JSON or XML for data representation.

Other HTTP methods available for API usage include:

1. **POST**:
   * **Purpose**: Used to submit data to be processed to a specified resource.
   * **Idempotent**: No. POST requests typically result in a new resource creation or an update to an existing resource, and they may not be idempotent.
   * **Safe**: No. POST requests typically modify server state.
   * **Usage**: POST is commonly used for creating new resources on the server, submitting form data, or making non-idempotent changes.
2. **DELETE**:
   * **Purpose**: Used to request the removal of a resource at a specified URI.
   * **Idempotent**: Yes. Repeated DELETE requests to the same resource should have the same result (the resource is deleted).
   * **Safe**: No. DELETE requests modify server state by removing a resource.
   * **Usage**: DELETE is used to delete resources on the server.

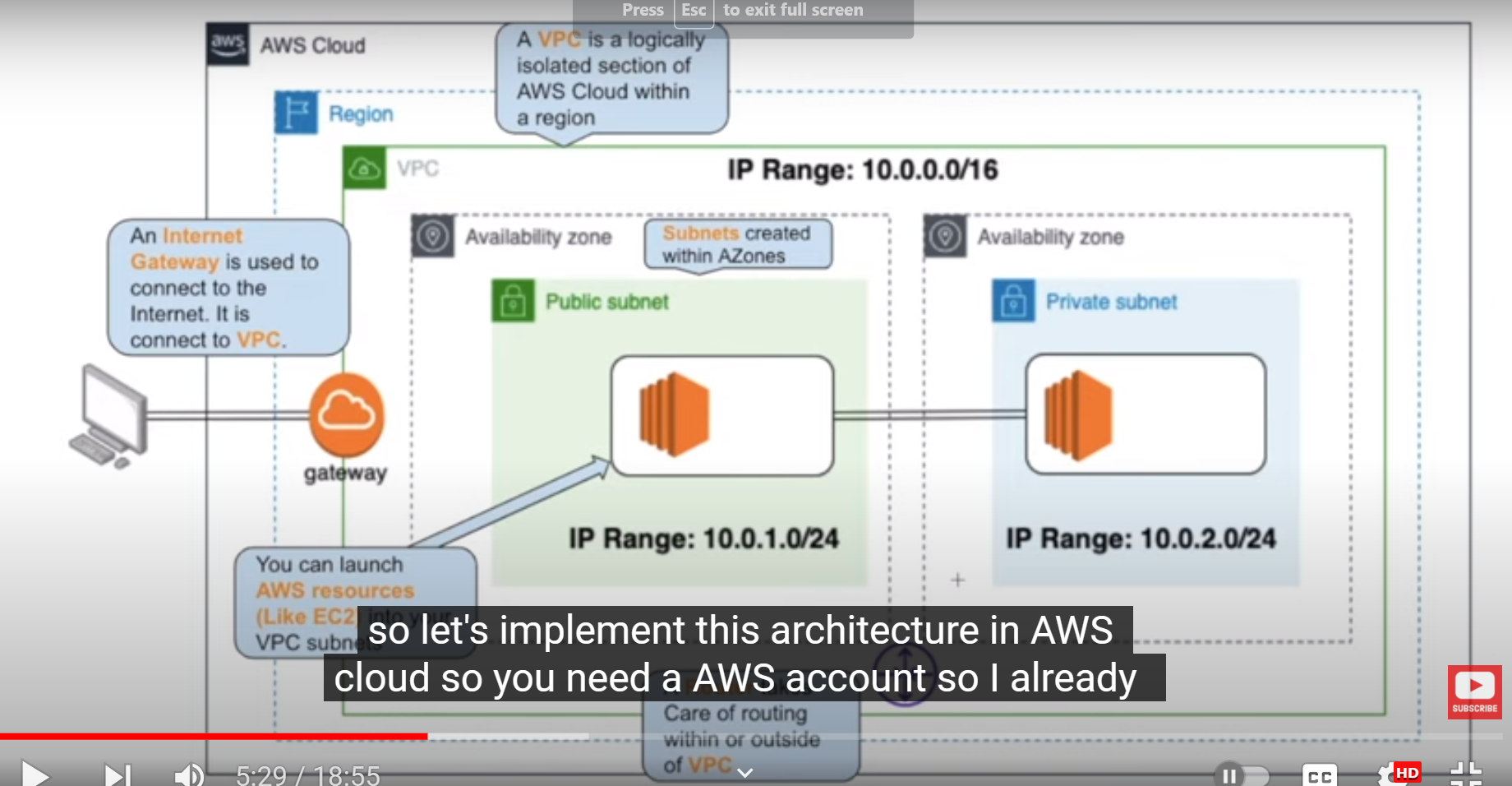
The choice of which HTTP method to use depends on the specific requirements of your API and the action you want to perform:

* Use **GET** for retrieving data.
* Use **PUT** to update or create a resource at a specific URI.
* Use **POST** to submit data to be processed (e.g., creating new resources or making non-idempotent changes).
* Use **DELETE** to remove a resource at a specific URI.

RESTful APIs often use a combination of these methods to provide a full range of CRUD (Create, Read, Update, Delete) operations on resources. The choice of which method to use for a particular operation should align with the HTTP method's semantics and the principles of RESTful design.

Top of Form

<https://youtu.be/fkcgpltZZfw?si=7E7y6O4-p3zt-MPv> - VPC, Route table, internet gateway



explain with the example for VPC, subnet, router table , ACL, internet gateway in AWS

ChatGPT

In AWS (Amazon Web Services), several networking components are used to create isolated and secure environments for your cloud resources. Let's break down these components with examples:

1. **Virtual Private Cloud (VPC):**
   * A VPC is a logically isolated section of the AWS cloud where you can launch AWS resources. It acts as your own virtual network within AWS.
   * Example: Imagine you're building a web application. You can create a VPC to isolate your application's resources from other AWS customers. This VPC will serve as the network infrastructure for your application.
2. **Subnet:**
   * Subnets are subdivisions within a VPC. They allow you to partition your VPC into smaller, manageable networks.
   * Example: In your web application's VPC, you can create subnets for different purposes, like a public subnet for web servers and a private subnet for databases. This separation helps in controlling access and security.
3. **Route Table:**
   * A route table contains rules (routes) that determine where network traffic is directed within a VPC.
   * Example: You might have a route table that directs traffic destined for the internet to an Internet Gateway (IGW) and traffic destined for your corporate network to a Virtual Private Gateway (VGW) for site-to-site VPN connectivity.
4. **Access Control List (ACL):**
   * An ACL is a network-level firewall that controls traffic entering or leaving a subnet. It acts as a security layer at the subnet level.
   * Example: You can create an ACL for a subnet to allow or deny specific types of traffic. For instance, you can configure an ACL to allow inbound HTTP and HTTPS traffic while denying all other inbound traffic to a public subnet.
5. **Internet Gateway (IGW):**
   * An IGW is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet.
   * Example: To make your web application publicly accessible, you attach an IGW to your VPC and configure your route tables to direct traffic from your public subnet to the IGW. This allows users on the internet to access your web servers.

Here's a simplified illustration of how these components work together:

* You have a VPC for your web application.
* Within that VPC, you create two subnets: one public subnet for web servers and one private subnet for databases.
* You configure a route table for the public subnet to send traffic to the IGW, making it accessible from the internet.
* An ACL is associated with each subnet. The public subnet's ACL allows HTTP and HTTPS traffic while the private subnet's ACL restricts all inbound traffic.
* The IGW facilitates communication between your public subnet and the internet.

By configuring and using these AWS networking components effectively, you can build secure and scalable cloud infrastructures for your applications.

Top of Form

Regenerate

The equivalent service to Azure File Storage in AWS is Amazon Elastic File System (EFS).