**Introduction**

Thank you for choosing our company to help you through this momentous moment in your company’s migration to the cloud. In this document, I will talk through the steps that should be followed to successfully migrate the application PETRA onto the cloud and the potential overall cost for the process.

**Assessment and Planning**

The first step would be to analyze the current infrastructure of PETRA, such as the Windows server versions that would require services such as AWS Application Migration Service and AWS Migration Hub Orchestrator to help automate the transfer of data as simplistic and easy as possible.

AWS also provides storage for MS SQL databases as Microsoft has ended its support for the version which means it will no longer receive security updates, which will leave your database vulnerable to attacks. AWS can help overcome this by providing a range of options such as rehosting to EC2 or even re-platforming entirely to RDS which will fully manage the database for you, allowing for easy resizability and cost efficiency.

Another step that needs to be planned for is any potential downtime that could occur during the migration period, this can be roughly calculated by using the following formula provided by AWS:

Number of Days = (Terabytes \* 8 bites per Byte)/(CIRCUIT gigabits per second \*

NETWORK\_UTILIZATION percent \* 3600 seconds per hour \* AVAILABLE\_HOURS)

By using the provided formula, we can a rough estimate of how long the process would take, which allows you to prepare ahead of time so that the Sales, Build and Fulfilment departments can feel the effects of the process as minimally as possible.

**Environment Setup on AWS**

Once the planning has been completed, the next steps will be preparing and creating the environments on AWS so they're ready for the migration to take place.

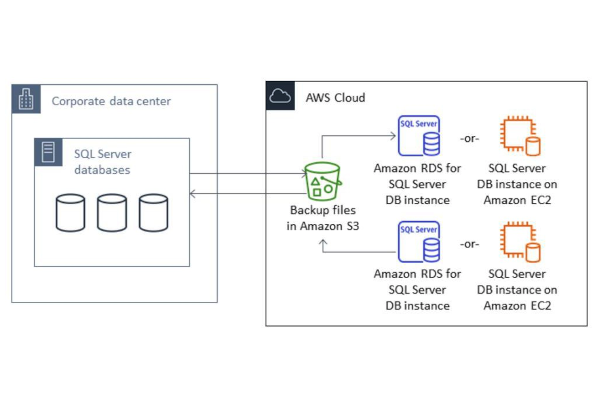
The first thing that needs to be set is the AWS account for the company, this will allow for all the services and support needed for the process as well as for any future help. This will also allow you to create identity and access management (also known as IAM), which lets you set up roles and permissions for users who will have access to the account and limit them to only have the access relevant to their roles, also known as the principle of least privilege.

The next step would be to set up a virtual private cloud as this will allow for network infrastructure that is required for launching different AWS resources as well as allow for the setup of security groups which control the flow of inbound and outbound traffic to the servers/databases which is useful to have a higher control of what can be accessed and what cant be. VPC also gives users the option of route tables, what this does is help to direct the flow of traffic from both internal as well as external(internet-bound traffic/on-premise networks) and so allows for higher control of traffic which makes scalability and adaptation easier as modifying route tables is straightforward.

**Data Migration**

Once the environment has been set and is ready, the next step is to migrate the data from the on-site database to AWS RDS (rational database service) with the help of AWS DMS (database migration service). The first step is to create a replication subnet in the AWS DMS console, the next step is to set up a source and endpoint for the connection between the onsite database and the AWS RDS instance. Once the two points have been set up, you can create the migration task to link them together and configure the schema and table mapping to start the migration process.

Once the initial migration process is completed, you need to start validating the data integrity on the AWS RDS side to ensure there were no errors with the replicated data and to update DNS records with any required changes.



**Application migration**

Once the databases and servers have been successfully transferred across to the AWS with the help of AWS EC2 (elastic compute cloud) which allows for the creation of EC2 instances that act as virtual servers which can have a customized core, memory size and performance that suit the needs of the company.

We can start adjusting the application’s coding to fit the new cloud system, such as the access code to connect with the cloud database which allows for a stronger secure connection and optimal performance.

The next key step would be to shut down the onsite servers and databases as this will allow for more accurate testing of the application and making sure the connection to the cloud is stable and working as expected, whilst also checking to make sure the databases stored have all the correct information that they should. It also allows for any adjustments that may have been initially missed out.  
  
Once the application has been confirmed to be working as intended with the cloud, it allows you to have access to a more cost-efficient running of the overall process and allows for future growth and scalability with the cloud.

**DNS and Active Directory Integration**

DNS (Domain Name System) and Active Directory (AD) integration are needed for efficient network operations. DNS changes domain names into IP addresses, allowing resource location on the internet. With Active Directory, DNS provides name resolution services for domain-joined machines. Active Directory relies heavily on DNS to locate and communicate with domain controllers, ensuring authentication and authorization within the network

Integration between DNS and Active Directory involves DNS namespaces with Active Directory domain names. This integration helps domain controller discovery, allowing for the authentication processes. DNS records are used to direct clients to the appropriate domain controllers. This integration ensures a reliable and smooth network environment, enabling Active Directory services to function through efficient DNS resolution.

**Security implements**

In the security implementation steps of migration, updating firewalls and Network Access Control Lists (NACLs) is important to fortify the cloud environment against potential threats and hacking. Firewalls, both at the application and network levels, can be altered to accommodate changes in the infrastructure. Specifically, firewall rules are modified to permit or deny traffic based on updated IP addresses and configurations in the cloud. This ensures that only authorized connections are allowed, enhancing the overall security posture.

Simultaneously, NACLs, which act as an additional layer of security controlling inbound and outbound traffic at the subnet level, are reconfigured to align with the new cloud infrastructure. By specifying rules for IP addresses, ports, and protocols, NACLs provide control over network traffic. During migration, these controls can be changed to allow communication between components while maintaining a secure environment.

**Overall cost evaluation**

Using the provided rates you are willing to pay, I believe the overall cost for running and migrating to the cloud will cost roughly a total of £21,095.19 for the initial month. This can be broken down into the following:

The monthly cost for using AWS - 95.19

Cloud consultant for roughly 3 days - 6000

Solution architect for 3 days - 3000

Server migration engineer for 5 days - 3250

Database migration engineer for 5 days - 3750

first/second line for 20 days to assist with any issues/help - 5000