FishTank Ltd proposal

Migrating PETRA to AWS Cloud

There are 7 common migration strategies for moving applications to the cloud these strategies need to be considered for each application before migrating and consist of:

1. Refactor/re-architect – Application is modified so that it can take full advantage of cloud-native features to improve agility, performance, and scalability.
2. Replatform (lift and reshape) – Make cloud optimisations to take advantage of cloud capabilities without changing the core architecture.
3. Repurchase (drop and shop) – Switch to a different product by moving from a traditional license to a SaaS model.
4. Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities.
5. Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations.
6. Retain (revisit) – Some applications may simply not be ready to migrate to the cloud at the current time and must remain on premises and can be re-evaluated for migration at a future date.
7. Retire – Decommission the application as it is no longer needed.

For a company of your size, it would be recommended to use the lift and shift method as it helps reduce migration complexities and lowers timelines enabling you to switch to cloud faster.

Using AWS Application Discovery Service can be the first step in migrating PETRA over the AWS cloud. This helps plan migration by collecting usage and configuration data about your on-premises servers and databases. You'll then be able to view the discovered servers and group them into applications and track the migration status of each application. This service offers two ways of performing discovery and collecting data:

1. Agentless Discovery: performed by deploying the Application Discovery Service Agentless collector through your VMware vCenter. After agentless collector is configured, it identifies virtual machines and any hosts associated with vCenter. It then collects the static configuration data: Server hostnames, IP addresses, MAC addresses, disk resource allocations, database engine versions and database schemas.
2. Agent-based discovery: performed by deploying the Application Discovery Agent on each of your VMs and physical servers. It collects static configuration data, detailed time-series

system-performance information, inbound and outbound network connections and processes that are running.

AWS Migration Hub can be used to create an inventory of your applications and evaluate their compatibility with AWS and can help you plan, track and manage your migration projects.

The next step after assessing and planning the migration would be to mobilise the migration. You can use Application Migration Service to quickly lift and shift physical, virtual, or cloud servers without compatibility issues, performance impact, or long cutover windows. This service continuously replicates your source servers to your AWS account. Then, when you're ready to migrate, Application Migration Service automatically converts and launches your servers on AWS with minimal downtime.

The final step would be migrating the workload to AWS cloud and there are steps to consider during the migrate phase:

- Prepare the AWS environment – Before you begin the migration process, you must prepare the AWS environment by creating an Amazon Machine Image (AMI) and setting up a VPC where you're migrating the workload.
- Configure the migration – Configure the migration by selecting the source server and specifying the target instance type, storage, and network settings.
- Perform the migration – After the configuration is complete, perform the migration. The process involves replicating the data, testing the migrated workload, and performing final cutovers to switch over to the migrated workload.
- Validate the migration – After the migration is complete, there are checks in place to ensure that the migrated workload is functioning as expected. Tests are performed to ensure that the security and compliance requirements are met.
- Optimise the migrated workload – The migrated workload is then optimised by resizing the instance, configuring auto-scaling, and implementing cost-saving strategies such as Reserved Instances or Spot Instances.
- Monitor and manage the migrated workload – The workload is then continuously monitored and managed to ensure optimal performance and security. Amazon CloudWatch is a useful tool for monitoring this phase.

Detailed pricing

Using the PETRA Asset list provided by you, we have come up with an estimate of the running costs for FishTank. Here is a list of all the services that are needed to migrate successfully to the cloud. The total monthly cost of running these services come to: 2,645.75 USD.

Service Name	Status	Upfront cost	Monthly cost	Description
Amazon Aurora MySQL-Compatible	-	0.00 USD	184.69 USD	Database
Amazon Aurora MySQL-Compatible	-	0.00 USD	184.69 USD	Database
Amazon EC2	-	0.00 USD	785.36 USD	PETRA App Servers
Amazon EC2	-	0.00 USD	386.28 USD	Web Servers
Amazon EC2	-	0.00 USD	193.05 USD	Print Servers
Amazon Elastic File System (EFS)	-	0.00 USD	327.68 USD	Shared Storage
AWS Directory Service	-	0.00 USD	584.00 USD	Domain Controller
AWS Application Migration Service	-	0.00 USD	0.00 USD	-

Adding onto this with the day rates listed in the request, the first month, all roles will be needed as migration engineers and cloud consultants need to be employed to oversee the migration. This comes out to a total of £108,000 a month for however long it takes for the migration to occur within FishTank. This will then decrease as engineers and consultants may not be required. However, cloud support roles should still be retained after migration to oversee any problems or issues that may arise. This monthly cost will be around £20,000 (Business Analyst, First/Second line Cloud support, Third line Cloud support).

This brings the total running cost including both cloud roles and cost of using cloud services to 2,645.75 USD + 20,000 GBP.

Network Design

A VPC will be used in the network design for hosting resources and makes it able to create subnets, select your own IP address range and configure route tables and network gateways.

Within the VPC, subnets are a range of IP addresses that can launch AWS resources.

- Public subnets allow traffic to flow in either direction and requires an internet connection.
- Private subnets allow indirect access to the internet. Traffic stays within your private network.

Internet gateways are needed in this network so that subnets can communicate between resources in the VPC and the internet by providing a target for internet-routable-traffic. It also protects IP addresses on your network by performing NAT

Route tables are a set of routes that the VPC uses to direct network traffic.

Security Groups and NACLs

Here is an idea of the security groups and NACLs that could be used when migrating over to the cloud.

Source	**Protocol**	**Port**	**Comments**
Web Servers	TCP	443	Allows inbound HTTPS from internet
Web Servers	TCP	3389	Allows inbound RDP from internet (for administrators)
App Servers	TCP	9000	Allows inbound from Web Servers
App Servers	TCP	3389	Allows inbound RDP from internet (for administrators)
Database Servers	TCP	3389	Allows inbound RDP from internet (for administrators)
Database Servers	TCP	unknown	Allows inbound from unknown port (from app servers)
Active Directory	TCP	all	Allows inbound from all ports and servers (for Active Directory access)
Public Subnet NACL	TCP	80 443	Allows HTTP/HTTPS from internet
Public Subnet NACL	TCP	3389	Allow RDP from internet
Public Subnet NACL	TCP	all	Allow all traffic from the private subnets
Private Subnet NACL:	TCP	all	Allow traffic from public subnet
Private Subnet NACL:	TCP	all	Allow traffic from Active Directory subnet

Schematic diagram of cloud architecture

Here is a visualisation of your architecture when your system has been fully migrated to the cloud.