THE DEPARTMENT OF
COMPUTER SCIENCE
& ENGINEERING
計算機科學及工程學系

[Co-op FYP] Hybrid Cloud platform web application implementation for MPF management for KPMG

By Fortino Tan, Advised by Prof. Zhou Xiaofang

KPMG

Corporate Background

- The laborious administrative process faced by workers whenever they switch employers, necessitating the creation of a new account, creates an inconvenience in monitoring one's retirement
- High administrative costs imposed by MPF trustees → caused by the creation of MPF accounts and the high volume of paper-based transactions

Solution:
eMPF platform → purpose of eMPF:

- The eMPF system enables individuals to have a consolidated view of their retirement savings, including all their previous and current MPF accounts, eliminating the need to individually monitor and manage multiple accounts.
- The eMPF system streamlines administrative processes, reducing reliance on paper-based transactions and ultimately lowering the high administrative costs associated with the MPF system.

Project Background

- MPF Trustees legacy infrastructure → The database of the
- Incompatibility with system to the new eMPF platform
- Additional residual functions that are needed to be implemented for the trustee system

Objectives

- Design and provide the network overview diagram which includes the high-level overview of cloud design, detailed overview of on the bank on premise system and
- Design the detailed system design cloud diagram for the hybrid cloud implementation
- Provisioning all the necessary cloud resources and create hybrid cloud environment
- Deploy all the necessary AWS resources in trustee's environment
- Do SIT testing with the provisioned resources
- Support for development for the BMS (Trustee's internal bank system)

Tech stack

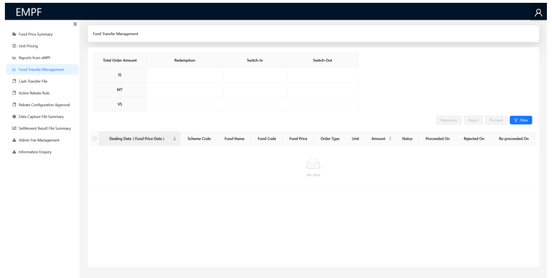
BMS system tech stack:
1. NextJS → front end
2. MSQL → Database

Hybrid Cloud:
1. AWS → Cloud service provider
2. Terraform → Infrastructure as code tool

BMS System

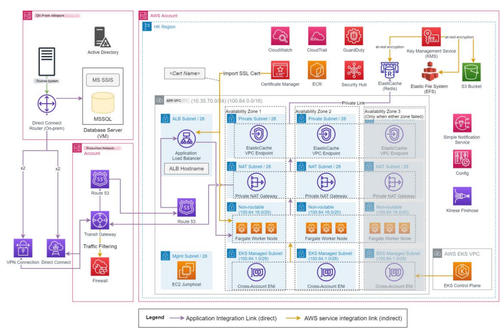
For the internal web application application there will be 4 types of functional process modules:

1. Admin → Employee/Employer enquiry , admin fees reviewing
2. Fund accounting → activities related to funds (create, edit, review and upload)
3. Rebate & Commission → rebate rules and commission rules
4. All teams function → BMS report and eMPF report



The example picture is the example of fund transfer management from fund accounting

Hybrid Cloud platform



The Figure above represents the final phase of the hybrid cloud platform design which Explains the flow from the AWS account to host the BMS system which will connect to the Production network through transit gateway and to the on-premise for eMPF platform

Resource provisioning

```
# module.deployer-ec2.aws_instance.ec2-instance will be created
  resource "aws_instance" "ec2-instance" {
      + ami                                  = "ami-07091c106c3860229"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = false
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = false
      + ebs_optimized                        = true
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = [redacted]-bms-adm-dev-ec2-bastion-linux-rhel-1_profile"
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = [redacted]-bms-adm-dev-ec2-bastion-linux-rhel-1-sshkey"
      + monitoring                           = false
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
```

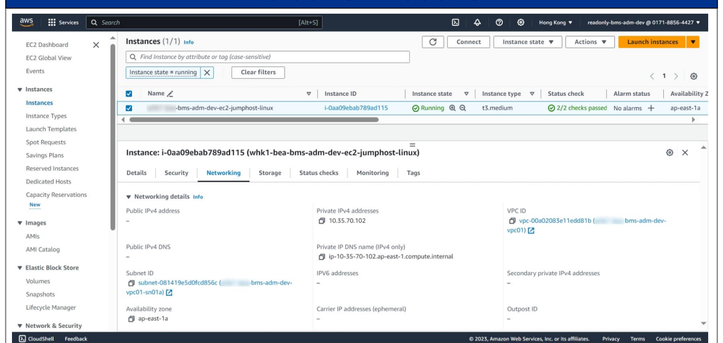
```
ly complete! Resources: 9 added, 0 changed, 0 destroyed.

tputs:

loyer-instance-arn = "arn:aws:ec2:ap-east-1:017188564427:instance/i-01cd61282a756362a"
loyer-instance-id = "i-01cd61282a756362a"
t-updated = "2023-12-11T08:56:45Z"
ed out waiting for input: auto-logout
```

The follow figure is the example of conducting SIT testing which through provisioning terraform resources in the developer KPMG account (AWS account in cloud diagram) and this picture the example of provisioning the EC2 Jumphost

Evaluation Results



Evaluation result is basically evaluating the SIT result which for this case is evaluating the EC2 resources that was provisioned in terraform. The output in the figure is shown from the Production network account and we can see the network details of EC2 Jumphost