KPMG Challenge –

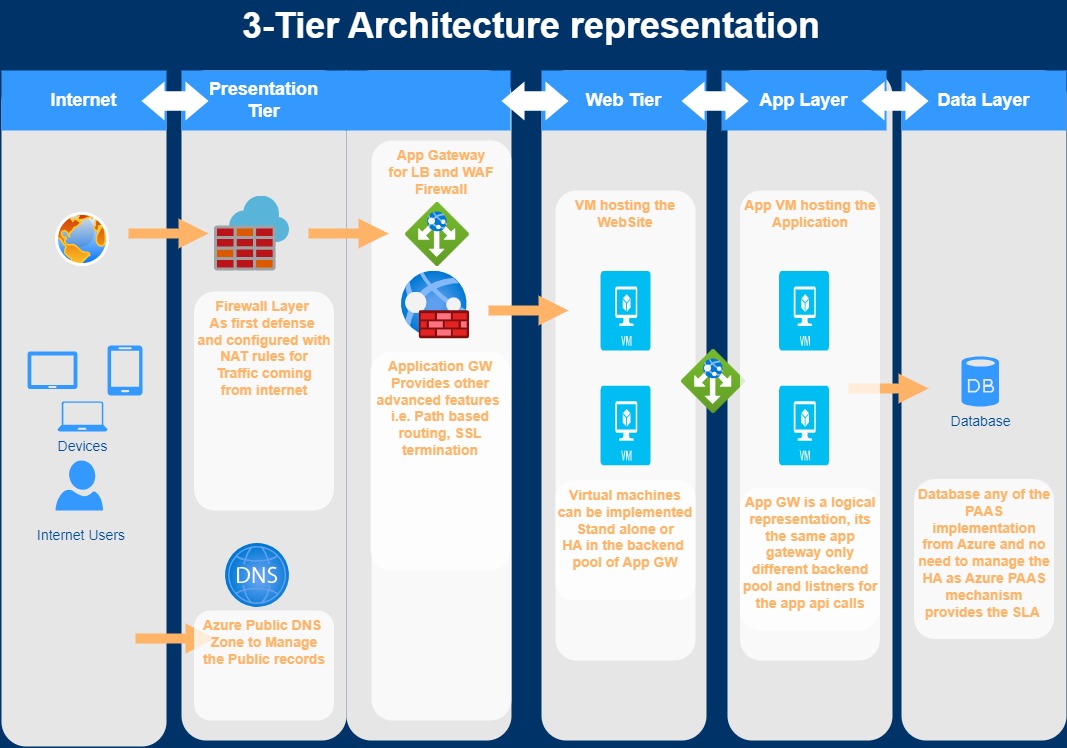
Challenge #1

A 3-tier environment is a common setup. Use a tool of your choosing/familiarity create these

resources in a cloud environment (Azure/AWS/GCP).

Ans: - Web-Apps can be implemented in several ways, below is Sample implementation of a 3- tier architecture.

In the below, the Web and Apps deployed using simple IAAS services and for the Data layer used a PAAS implementation of Data base.



Challenge #2

We need to write code that will query the meta data of an instance within AWS or Azure or GCP

and provide a json formatted output.

Ans: - This PowerShell script sends a GET request to the specified URL with the given headers, retrieves the JSON response, converts it to a PowerShell object, and then formats it as JSON with an indentation of 4 spaces. Finally, it outputs the formatted JSON.

$headers = @{

'Metadata' = 'true'

}

$metadata\_url = 'http://169.254.169.254/metadata/instance?api-version=2021-02-01'

$response = Invoke-RestMethod -Uri $metadata\_url -Headers $headers

$formatted\_output = $response | ConvertTo-Json -Depth 4

Write-Output $formatted\_output

Challenge #3

We have a nested object. We would like a function where you pass in the object and a key and

get back the value.

* Example Inputs

object = {“a”:{“b”:{“c”:”d”}}}

key = a/b/c

This PowerShell function GetValueFromNestedObject accepts a hashtable ($obj) representing a nested object and a string ($key) representing the path to the desired value. It traverses the nested object using the specified key and returns the corresponding value. If the value is not found or if there's an error during traversal, it returns $null. The usage is demonstrated at the end of the script.

function GetValueFromNestedObject {

param (

[Parameter(Mandatory=$true)]

[System.Collections.Hashtable]$obj,

[Parameter(Mandatory=$true)]

[string]$key

)

$keys = $key -split '/'

$value = $obj

try {

foreach ($k in $keys) {

$value = $value[$k]

if (-not $value) {

return $null

}

}

return $value

}

catch {

return $null

}

}

$obj = @{

"a" = @{

"b" = @{

"c" = "d"

}

}

}

$key = "a/b/c"

$value = GetValueFromNestedObject -obj $obj -key $key

Write-Output $value