**Technical Challenge**

We understand your time is precious, but as discussed, we ask each applicant to take on our small challenges. There is no right or wrong approach and we're certainly not expecting war and peace 😊. We would expect no more than 30 minutes per challenge, if at the end of the time you haven’t completed don’t worry just send us what you have. (Uploaded to a public GIT repository – ensuring there is nothing KPMG sensitive in the test below there is no need for anything to be)

**Challenge #1**

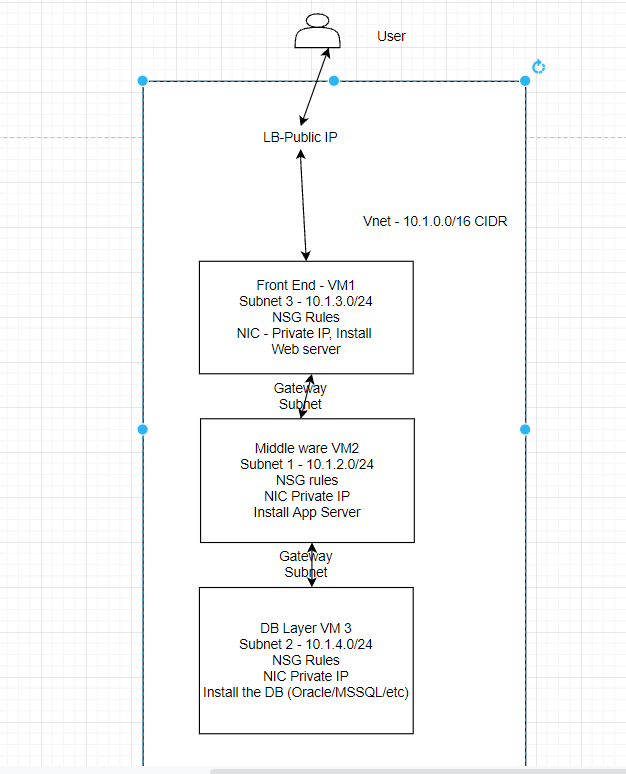
A 3-tier environment is a common setup. Use a tool of your choosing/familiarity create these resources. Please remember we will not be judged on the outcome but more focusing on the approach, style and reproducibility.

**Solution Approach 1**: Create a Vnet and divide that Vnet into 3 subnets.

Each subnet will be having the VMs as per below diagram.

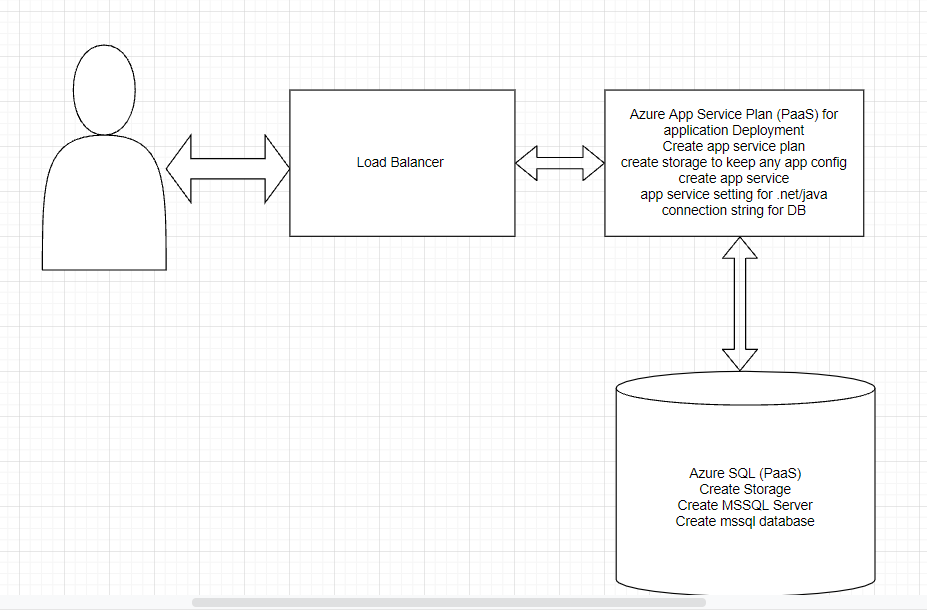
Create the VMs using Terraform scripts which is attached below.



**Solution Approach 2**: This approach is for Cloud Native and using Cloud PaaS services.

Create terraform scripts.



User the below azurerm to create the resources for Azure SQL and Azure App Service plans.

MSSQL:

azurerm\_resource\_group

azurerm\_storage\_account

azurerm\_mssql\_server

azurerm\_mssql\_database

App Service:

azurerm\_resource\_group

azurerm\_app\_service\_plan

azurerm\_app\_service 🡪 site\_config🡪app\_settings🡪connection\_string

**Solution Approach 3**:

Container Journey if you have docker images available

DB as stateful sets

App as normal deployment

PV and PVC as Azure file storage for MSSQL and MYSQL. For Postgres and Oracle does not work because both needs hard links. Need some extra configuration.

Helm Chart for the application and DB separately

Ingress Controller as Nginx can be deployment

Cloud Load balancer will be created

Ingress rules will be created and status will be forwarded to Load Balancer.

**Challenge #2**

We need to write code that will query the meta data of an instance within AWS and provide a json formatted output. The choice of language and implementation is up to you.

**Bonus Points**

The code allows for a particular data key to be retrieved individually

Hints

·         Aws Documentation (<https://docs.aws.amazon.com/>)

·         Azure Documentation (<https://docs.microsoft.com/en-us/azure/?product=featured>)

·         Google Documentation (<https://cloud.google.com/docs>)

**Solution**: All the Cloud providers provide the REST API end points for each service. We need to use that REST API End point to query whatever we need and get the JSON formatted meta data. For Example, To query the Azure VM meta data, we can use attached power shell code or call the Azure DevOps pipeline or Query AKS REST API End Points.

The below points need to be taken care before query of rest end point.

1. We need to know the REST API End Point. Just Query on google
2. Need to have credential for authentication and authorization in form of token.

**Sample code**:



**Challenge #3**

We have a nested object, we would like a function that you pass in the object and a key and get back the value. How this is implemented is up to you.

Example Inputs

object = {“a”:{“b”:{“c”:”d”}}}

key = a/b/c

object = {“x”:{“y”:{“z”:”a”}}}

key = x/y/z

value = a

**Solution**: We can make use of Python predefined library. Use the reducer and deep\_get from python or python Dictor.

**Sample Code**: I have googled it.

from functools import reduce

def deep\_get(object, keys):

return reduce(lambda d, key: d.get(key) if isinstance(d, dict) else None, keys.split("."), object)

Hints:

*We would like to see some tests. A quick read to help you along the way*

*We would expect it in any other language apart from elixir.*

[*A quick read to help you along the way*](https://hexdocs.pm/elixir/master/Kernel.html#get_in/2)

Once this has been completed please send us the output so we can get the ball rolling.