**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.

**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>)

We can use this below piece of python code along with the linux curl command to get the desired results.

curl -s http://169.254.169.254/latest/dynamic/instance-identity/document | python -c "import sys, json; print(json.load(sys.stdin))['region']"

**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

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.

**Ans #1:**

# creating a new vpc with dns resolution support

resource "aws\_vpc" "vpc\_Mytest" {

cidr\_block = "10.0.0.0/16"

enable\_dns\_support = true

enable\_dns\_hostnames = true

tags = {

Name = "Mytest VPC"

BuildWith = "terraform"

}

}

# adding public subnet

resource "aws\_subnet" "public\_subnet" {

vpc\_id = "${ aws\_vpc.vpc\_Mytest.id }"

cidr\_block = "10.0.1.0/24"

map\_public\_ip\_on\_launch = true

availability\_zone = "us-east-2a"

tags = {

Name = "Public Subnet"

BuildWith = "terraform"

}

}

# adding private subnet

resource "aws\_subnet" "private\_subnet" {

vpc\_id = "${ aws\_vpc.vpc\_Mytest.id }"

cidr\_block = "10.0.2.0/24"

availability\_zone = "us-east-2a"

tags = {

Name = "Private Subnet"

BuildWith = "terraform"

}

}

# adding internet gateway for external communication

resource "aws\_internet\_gateway" "internet\_gateway" {

vpc\_id = "${ aws\_vpc.vpc\_Mytest.id }"

tags = {

Name = "Internet Gateway"

BuildWith = "terraform"

}

}

# create external route to IGW

resource "aws\_route" "external\_route" {

route\_table\_id = "${ aws\_vpc.vpc\_Mytest.main\_route\_table\_id }"

destination\_cidr\_block = "0.0.0.0/0"

gateway\_id = "${ aws\_internet\_gateway.internet\_gateway.id }"

}

# adding an elastic IP

resource "aws\_eip" "elastic\_ip" {

vpc = true

depends\_on = ["aws\_internet\_gateway.internet\_gateway"]

}

# creating the NAT gateway

resource "aws\_nat\_gateway" "nat" {

allocation\_id = "${ aws\_eip.elastic\_ip.id }"

subnet\_id = "${ aws\_subnet.public\_subnet.id }"

depends\_on = ["aws\_internet\_gateway.internet\_gateway"]

}

# creating private route table

resource "aws\_route\_table" "private\_route\_table" {

vpc\_id = "${ aws\_vpc.vpc\_Mytest.id }"

tags {

Name = "Private Subnet Route Table"

BuildWith = "terraform"

}

}

# adding private route table to nat

resource "aws\_route" "private\_route" {

route\_table\_id = "${ aws\_route\_table.private\_route\_table.id }"

destination\_cidr\_block = "0.0.0.0/0"

nat\_gateway\_id = "${ aws\_nat\_gateway.nat.id }"

}

# associate subnet public to public route table

resource "aws\_route\_table\_association" "public\_subnet\_association" {

subnet\_id = "${ aws\_subnet.public\_subnet.id }"

route\_table\_id = "${ aws\_vpc.vpc\_Mytest.main\_route\_table\_id }"

}

# associate subnet private subnet to private route table

resource "aws\_route\_table\_association" "private\_subnet\_association" {

subnet\_id = "${ aws\_subnet.private\_subnet.id }"

route\_table\_id = "${ aws\_route\_table.private\_route\_table.id }"

}

**Ans #2:**

We can use this below piece of python code along with the linux curl command to get the desired results.

curl -s http://169.254.169.254/latest/dynamic/instance-identity/document | python -c "import sys, json; print(json.load(sys.stdin))['region']"

**Ans #3:**

def nested\_get(input\_dict, nested\_key):

internal\_dict\_value = input\_dict

for k in nested\_key:

internal\_dict\_value = internal\_dict\_value.get(k, None)

if internal\_dict\_value is None:

return None

return internal\_dict\_value

print(nested\_get({"a":{"b":{"c":'d'}}},["a","b","c"]))