# Associate & Software Engineer Technical Assessment

Welcome to the ---- technical assessment for Software Engineers. This is an exercise developed to try and offer a more realistic insight into what your role would be when joining ---- and give you a chance to show your ability, skill and approach in an environment that hopefully will be more comfortable for you. Some things to note about this assessment:

* This is effectively an “open book” assessment, feel free to use any internet resources that you would normally use and whatever environment you are happy with.
* There is a deadline for this assessment that you would have gotten with the email with these instructions. This deadline is intended to give you an opportunity to find a block of time that suits you for this effort. It is **not** intended to be the closing time from when you got the email for this assessment. We’re looking for approx. few hours of effort, which gives you a chance to deliver your solution with some polish, while also trying to respect your time.
* The assignment is testing the candidate’s ability to develop Infrastructure automation code using Terraform.

## Terraform Assessment

As part of accessing your Infrastructure as code skills, we would like to focus on technical skills for Terraform.

The objective of this assessment is to create a small micro EC2 instance inside a default VPC which can provision Apache http and output a simple custom message to index.html page. Also, as part of provisioning, it should create a S3 bucket. The name of the s3 bucket should be passed as a suitable variable. Also, the environment type can be determined at runtime using a variable (for example – qa/dev)

For this assessment, consider the following resources to be provisioned by Terraform - EC2, Route53, S3 bucket, IAM role and policies appropriate to the environment and suitable security group applied and other security controls to be factored in, SSH access to the EC2 access to be provisioned.

Prerequisites

1. A personal free tier membership to an AWS account with appropriate access.
2. AWS CLI installed to configure access
3. Terraform installed
4. Git installed and a public git repo to pull the code once finished.
5. Setup an S3 bucket where you can store your remote backend state file.

## Instructions

As part of this assignment, you are required to create Infrastructure as code git repository which can create the above specified infrastructure resources in AWS.

You can use whichever terraform CLI/Provider version you feel comfortable with but please outline on your readme file which version it supports.

Terraform Inputs:

Some of the inputs which you can capture as variables for your project

1. Aws\_region
2. Bucket name
3. Environment ( for simplicity “qa” or “dev”)

( PS- feel free to use your own naming conventions and this variable list can be customized , extended to include which other variables you feel appropriate)

Data Providers:

1. Availability zones
2. Aws\_ami to filter the latest amazon linux2 hvm image.

Requirements:

1. To track resources per environment, each resource should have a list of common tags

|  |  |
| --- | --- |
| Aws\_region | us-east-1 |
| Environment | qa/dev |

1. As part of provisioning of the EC2 instance ( micro ) , install Apache HTTP and start the service. Create a custom message so it can be viewed from index.html page
2. Create a S3 bucket per environment depending on the project name and environment and setup appropriate IAM access for the bucket according to the policies in the table below.
3. Depending on which environment, create an S3 bucket for the working environment. The requirement is to have the following permissions per environment

|  |  |
| --- | --- |
| QA | DEV |
| S3:GET, Resource “bucket\_name” | S3:\*, Resource “bucket\_name” |
|  | Ec2:Describe\*, Resource “\*” |
|  |  |

**Terraform Outputs:**

1. Please output the public IP address which is created and where apache can be reached from.
2. The name of the s3 bucket which is created per environment.

**Testing:**

1. It should be possible to view the index.html page
2. If environment = dev at runtime, it should be possible to upload a file from the EC2 instance to it.
3. If environment = dev at runtime, it should be possible to describe tags from the aws-cli for the given instance.
4. If environment = qa at runtime, it should be possible only to have read only access to the s3 bucket.

**Assessment Deliverables:**

1. High-level documentation points including the version of terraform used.
2. Please push your code to a public git repository and email the repository url.
   * It should be possible for us to clone the repository and run your code and see the resources created within our own Aws account displaying appropriate terraform outputs.

Note:

Post assessment is complete, please ensure you destroy the resources from your account, so you are not charged outside the free tier allowance.

Best of luck!