



**EPAM Cloud & DevOps practice**

# **INFRASTRUCTURE AS CODE.**

## **TERRAFORM**

---

**Final task.**

**Legal Notice:** This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM®.

---

CONFIDENTIAL | Effective Date: 9-May-2022

## 1 HOME TASK DESCRIPTION

In this task, you will develop Azure based infrastructure with Terraform managing.

The infrastructure consists of two environments QA and PROD. Every environment includes VM, Network Interface Security group with network rules, Public IP, Virtual Network, Subnet and resource group

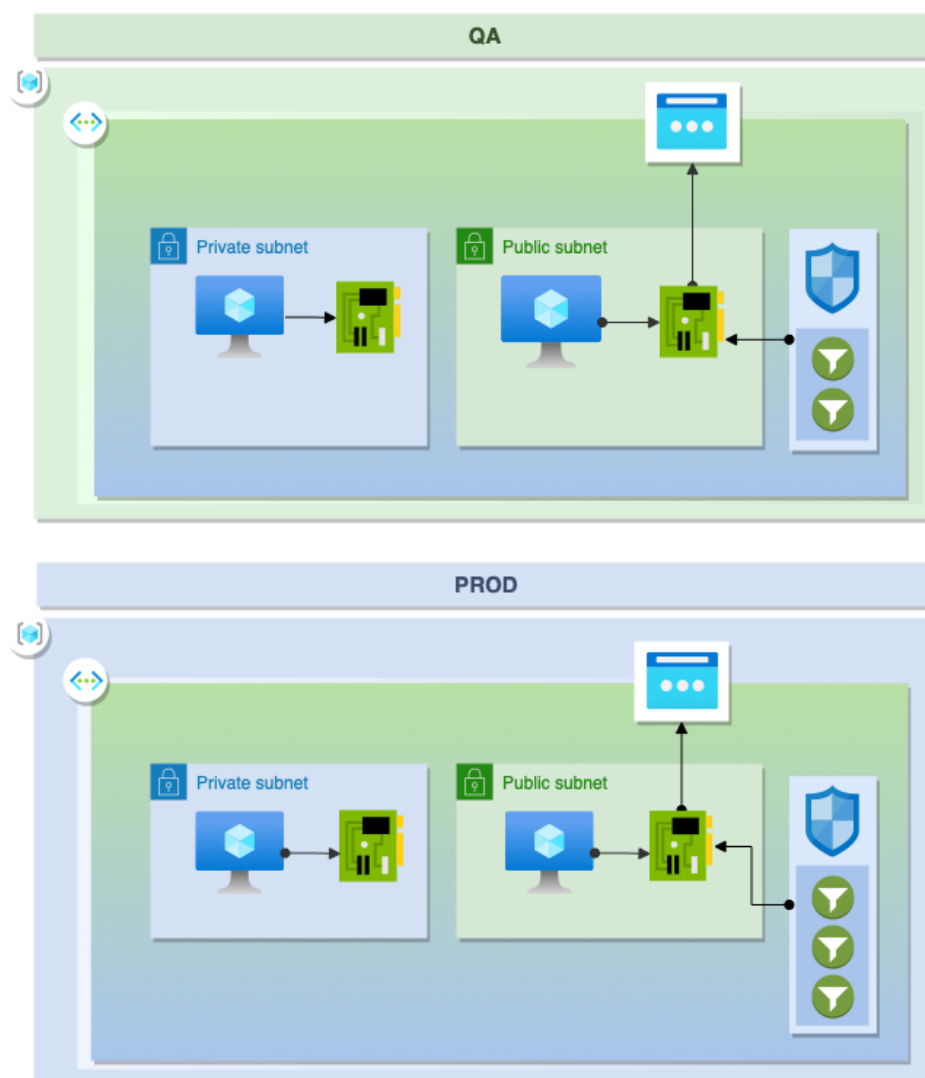
## 2 ENVIRONMENT STRUCTURE

### 2.1 QA

- Virtual Network has:
  - two subnets:
    - Private  
In the Private subnet placed virtual machine
    - Public  
In the Public subnet placed 1 virtual machine
  - One security group with 2 network rules:
    - Allow\_SSH
    - Allow\_RDP
- The network interface attached to the VM
- Public IP attached to the Network Interface
- All resources are placed in a resource group

### 2.2 PROD

- Virtual Network has:
  - two subnets:
    - Private  
In the Private subnet placed virtual machine
    - Public  
In the Public subnet placed virtual machine
  - One security group with 3 network rules:
    - Allow\_SSH
    - Allow\_RDP
    - Allow\_Monitor
- The network interface attached to the VM:
  - NIC is attached to the VM
- One Public IP is attached to the one Network Interface in the public subnet
- All resources are placed in a resource group



*Pic. 1 – Infrastructure scheme*

### 3 ESSENTIALS

1. Environments should be independent: env creates(updates) without deleting and changing another one
2. Code should be done with using backends, separate for PROD and QA
3. As you are going to create 2 environments, all resources should have -env suffix, depending on the environment, where the resource is located.
4. Use Modules
5. Be free in using loops, conditions, data, outputs etc.
6. Any solution should be logical and easily proofed
7. Be creative while using variables, don't use only strings if it's making the code better
8. Plan and Apply should be executable without error
9. Before homework check apply code for both environments

### 4 TASK

Create proper terraform configuration based on infrastructure and code requirements.

## 5      **USEFUL LINKS**

- [Built-in Functions](#)
- [Workspaces](#)
- [For expressions](#)
- [meta-arguments: for\\_each](#)
- [meta-arguments: count](#)
- [meta-arguments: Lifecycle](#)
- [Data-sources](#)
- [Variables and Outputs](#)
- [Modules](#)
- [Types of Values](#)
- [Sensitive-data](#)