**DevOps**

**Challenge**

**Intro**

First of all, congratulations for reaching this step of the hiring process. The idea with this challenge is to evaluate your analytical and technical skills.

**The Project**

Our development team needs a “production-ready” deployment of the microservice ***jumia\_phone\_validator*** (README for more details).

Your main goal is to create the infrastructure code to automate all the infrastructure and microservice deployment, which means that all configurations and processes must be automated (servers provision, docker images builds, docker containers, firewall management, etc).

We will give you access to AWS cloud provider so that you can use it to deploy the services.  
  
We will give you access to a GitHub account where you can find the ***jumia\_phone\_validator*** application.

**Goals**

● Automatically configure servers: install required software,

configure firewalls and configure SSH access

● Automatically deploy the microservice ***jumia\_phone\_validator***

with all the required configurations

● Automatically deploy the [database] service with all the

required configurations

● **Don’t use** any roles from Ansible Galaxy

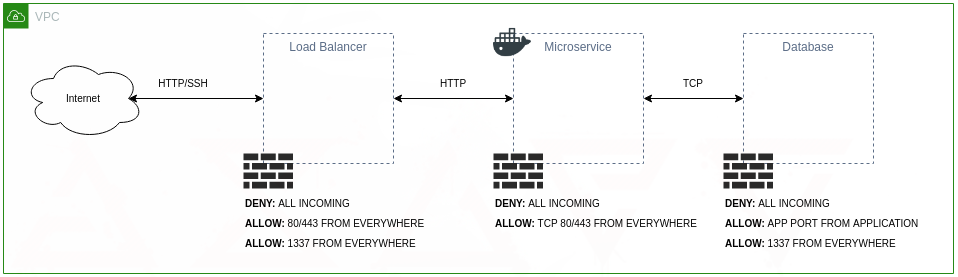
**Requirements**

In this section we will define to you the requirements for the

servers, network configuration and service deployment.

In the picture below you can see the diagram view of the

platform to deploy.



**Servers provisioning**

● Servers must have **one user** with root privileges

● Servers access should be **SSH only** (**port 1337**) with SSH Key

● Password authentication should be **disabled**

● Root login should be **disabled**

● Firewall should **deny all the incoming** traffic and only expose the required ports to the internet (e.g. SSH port, HTTP port)

**Servers features**

**1) Load-balancer**

a) Run a application load balancer to expose the microservice API via port **80 and 443**

**2) Database**

a) Run the **PostgreSQL** service on a **RDS**b) Create database: ***jumia\_phone\_validator***

c) Create database user (**jumia**) with secure password

d) Grant privileges to **read** and **write**

**3) Microservice**

a) Run the microservice ***jumia\_phone\_validator*** on a **Docker** container (take into account the

requirements described in the README file), with the **latest tagged** version.

b) Connect to the PostgreSQL database.

**Network configuration**

Using the operating system firewall (e.g. iptables, ufw, etc):

**1) Load-balancer**

a) Accept SSH connections **from any IP address**

b) Accept traffic from any IP address on **port 80 and 443**

**2) Database**

a) Accept SSH connections **from any IP address**

b) Accept requests from the **microservice** IP address

**3) Microservice**

a) Accept SSH connections **from any IP address**

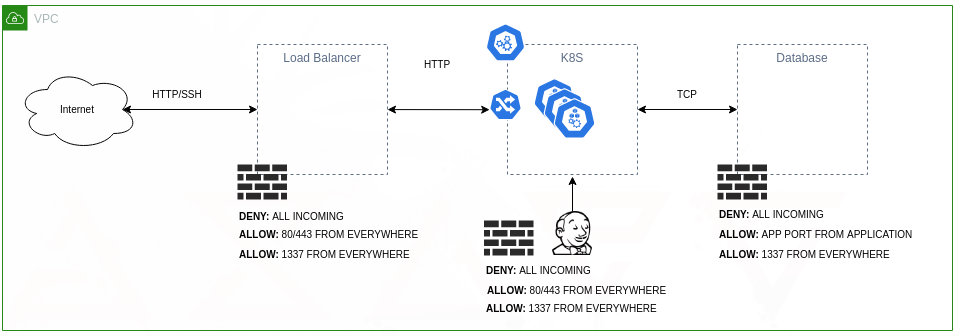
b) Accept requests from the **load-balancer** IP address

**Deliverables**

* A GitHub public git repository for the Ansible code with all the commits made during the development of this project
* Ansible playbook to deploy the microservice
* Ansible playbook to deploy the PostgreSQL
* Ansible playbook to configure the servers firewalls
* Terraform providers and modules to configure AWS resources
* The SSH access to all the created servers
* The service ***jumia\_phone\_validator*** running per described ahead.

**Bonus Points:**

* Provision a Kubernetes cluster EKS or managed with:
  + 3 Worker Nodes
  + 1 Node per Availability Zone
* Run the container inside the created Kubernetes cluster with:
  + 3 replicas, each assigned to a different node.
* Create a Jenkins server with a pipeline to deploy ***jumia\_phone\_validator.***
* Use Helm as a package manager to Kubernetes.



**Next Steps**

There are no right or wrong answers.   
This exercise was designed to evaluate as much as possible how you tackle a problem, your critical thinking and how you approach Infrastructure as Code.   
After its completion, we will have a f2f conversation to review your code and tasks performed to understand the reasons behind your choices and decisions.  
  
Good luck.