nuagenetworks

Web Application Assignment

In order to assess a candidate's ability to meet the demands of this role, we require everyone to complete a small project. This project is designed to test the candidate's knowledge in different areas and their ability to learn on the job. The evaluation will be based not only on the completion of the project, but also on the quality of the work.

We assume that you might not be familiar with everything listed, but we expect that you are able to figure things out and overcome any obstacle you face using whatever means possible.

The assignment contains five tasks, each task involve several skills from system administration to automation and scripting. The candidate is encouraged to approach this assignment as an overview of the tasks that he/she will have to do in our team. More importantly, this assignment will help us identify driven candidates who are willing to go above and beyond.

Assignment description

The goal of this assignment is to create a web application that can be scaled up easily using automation scripts. You can use any public or private cloud or any hypervisors technologies to install you servers.

NOTE 01	Each task includes one or several deliverables. Deliverables are intended to be delivered on time (fixed deadline).

NOTE 02	**Python and Shell are the only scripting languages accepted.**

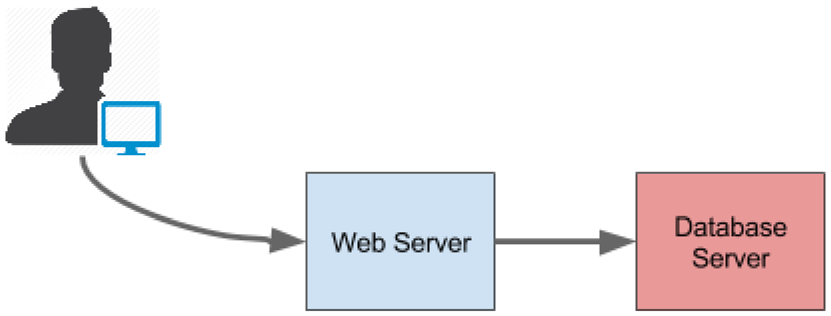
General Advice	**KISS (Keep It Simple and Stupid). Simple and stupid solutions are most of the time the better !**

General Evaluation Criteria

- Understand each technology/component used
- Documentation quality
- Source code and configuration simplicity

Create a Web App

The first task will be to create a web application. Each tier needs to be deployed on his own server. There is no specific guidelines as you can use any client-side frameworks, any server-side programming languages and any databases. The only rule is to have a simple REST API (one or two resources and should implement the basic methods GET/POST/PUT/DELETE), as the next task will use it.



Deliverables:

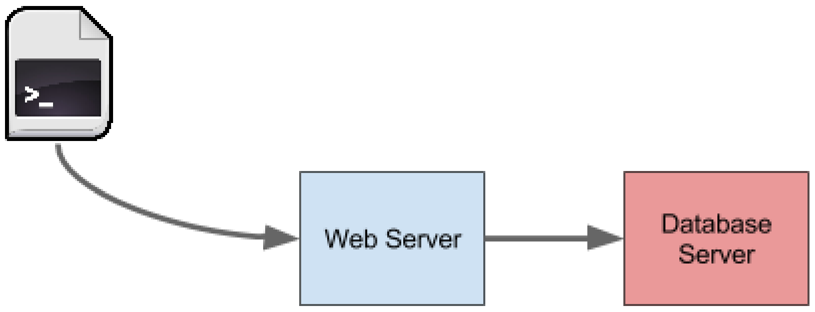
- Web Application documentation (design, architecture, ...)
- Web Application source code (Github)
- Live demonstration on dead-line

Evaluation Criteria

- Simplicity of the solution
- Code & documentation quality
- Choice of technologies
- REST API simplicity
- Nice UI **(Bonus point)**

Interact with your API

Now that you have built your application, you have to create a python script to interact with your simple REST API. The script must be able to demonstrate your application capabilities. The script must be developed in Python only.



Deliverables:

- Python script source code (Github)
- Updated Web Application documentation
- Live demonstration on dead-line

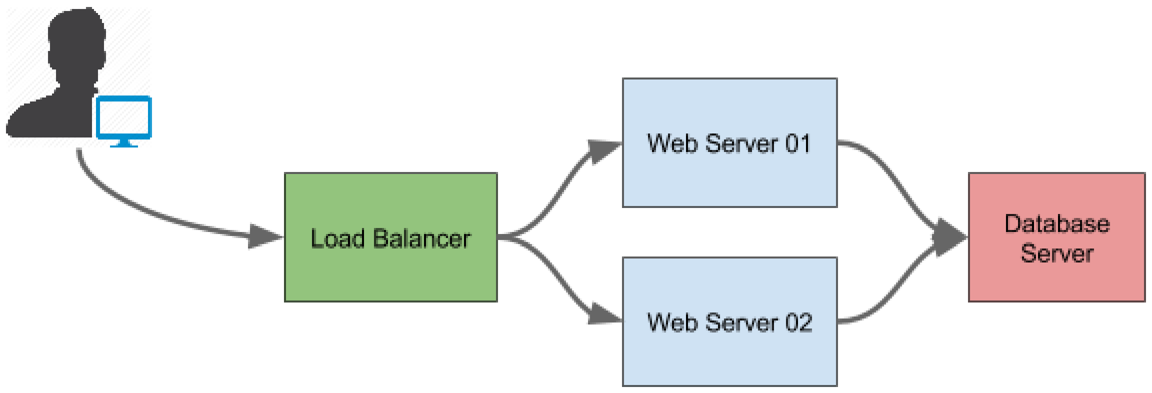
Evaluation Criteria

- Code & documentation quality
- Application capabilities coverage
- Packaging, ease of use on any Operating System (except Windows) **(Bonus point)**
- Error handling
- Script logging **(Bonus point)**

Scale-up the application

Now that you have built and tested your web application, you observed that your web server response time keeps increasing as you have more and more users. Even the static content takes several seconds to be sent to the client. It's time to scale-up your web application.

In this task, we ask you to configure your own load balancer to scale-up your web application. You can use any of these load balancers: HAProxy or NGINX.



Deliverables:

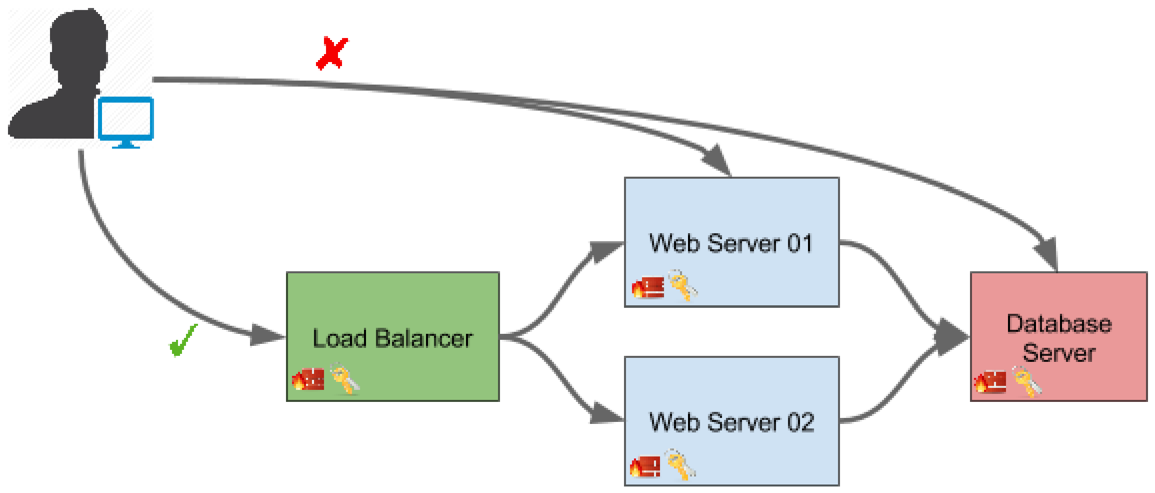
- Load Balancer configuration file
- Updated Web Application documentation
- Live demonstration on dead-line

Evaluation Criteria

- Simplicity of the solution
- Choice of the load balancer technology
- Load balancer redundancy **(Bonus point)**

Secure your servers

As you may have seen, security is one of the hottest topic nowadays. In this task, you have a secure your web application. Any flow that is not required must be blocked. Follow any hardening guidelines you will find relevant in your case. IPTABLES will be used as the only firewall technology. The following diagram is just an example to illustrate the task description.



Deliverables:

- IPTABLES configuration for each server
- Updated Web Application documentation
- Live demonstration on dead-line

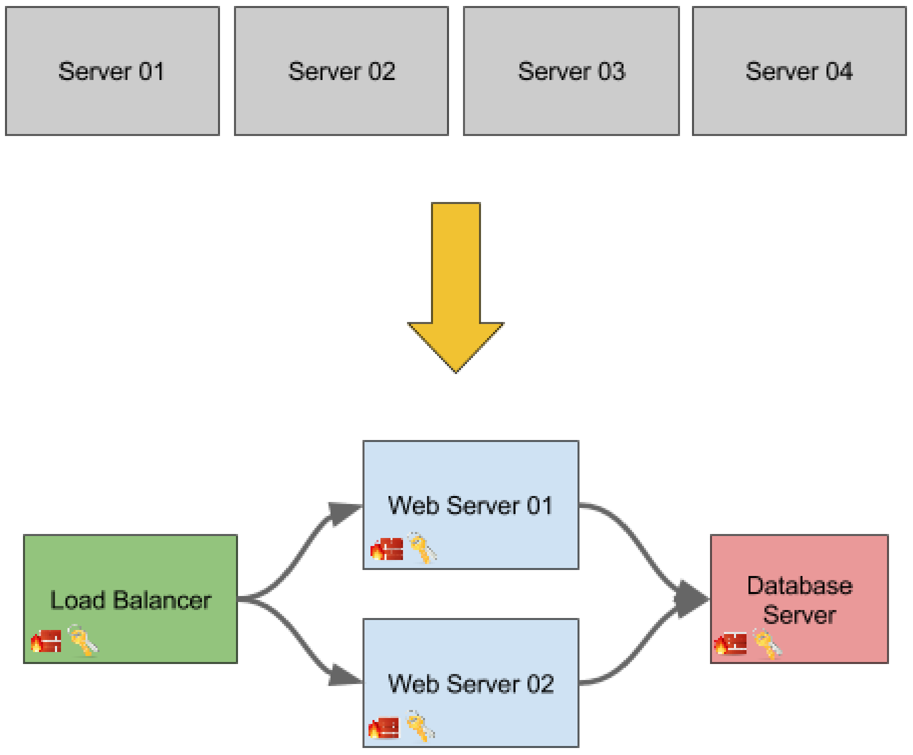
Evaluation Criteria

- Locking up ports correctly
- Enable rules violation logging **(Bonus point)**
- Server access security (SSH keys, root user, …)

Automate the deployment

As you are the only maintainer of this web application, you need to have tools to easily scale-up/scale-down the application, to secure the servers and to automate various configurations. This task will focus on your automation skills.

One guideline: automate what you did in all the previous tasks using Ansible.



Deliverables:

- Ansible playbook(s), role(s) and module(s) (Github).
- Updated Web Application documentation
- Live demonstration on dead-line

Evaluation Criteria

- Level of automation
- Able to deploy a new web server without manual installation, we assume that the web server operating system is installed and configured with an IP address. (**Bonus point**)
- Simplicity of the playbook(s)