**Design Document**

1. **Design Considerations**

The design is showed **in Figure 1**

* 1. **Availability Considerations**

At high level, the end system is described in Figure 1, where the server is part of an auto scaling group ( which prevents it from being accidently destroyed) and process web requests through an elastic load balancer.

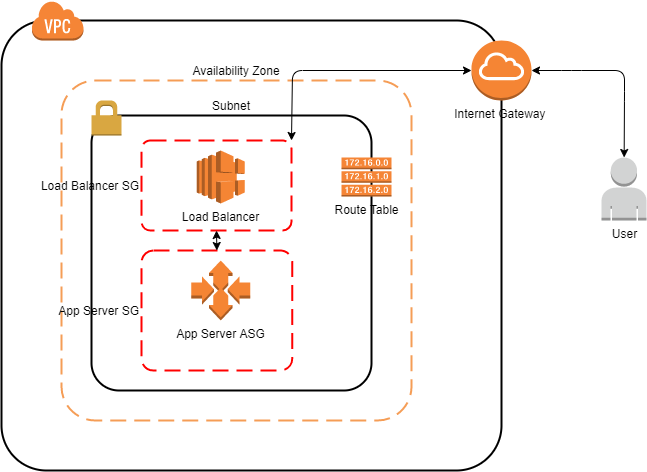
This design only use one availability zone since this is only a test. But the code can easily be extended to use three availability zones

* 1. **Security Considerations**

There are 5 layers protection for this system.

1. VPC protection
2. Subnet(Route table)
3. Security group rules( Load balancer allows ssh and http ingress, App server ASG allows only TCP port 4567)
4. Internet Gateway
5. At application level, the OS’s firewall ( i.e. iptables) can also be tuned to manage traffic.

The App server is put in a public subnet since it is an Internet facing instance. In practice, internal servers such as DB server should be put in a private subnet and connect to outside world using NAT gateways.



**Figure 1**

1. **Implementation**

I used Ansible to implement this test**.**

1. **How to run**
   1. **Presumptions**

My development system has following tools:

* Ansible: 2.7.10
* Python: 3.6
* Botocore and boto3
* Pip3( used to install python-netaddr)

It is also assumed that you have AWS credentials in your ~/.aws folder and your role has the privilege to create VPCs and EC2s

* 1. **Run steps**

1. $ *git clone* [*https://github.com/carlchen8617/REA.git*](https://github.com/carlchen8617/REA.git)
2. $ *cd REA*
3. (optional) $ *vi ansible/vars /main.yml* . I use ap-southeast-2(Sydney) and CIDR block 10.0.0.0/16. You may change these two vars to suit your circumstances.
4. While in REA directory.

Run

*$ ansible-playbook ./ansible/local-play.yml --tags "REA"*

1. If everything goes well, you should see Ansible print put something similar to the following:

ok: [localhost] => {  
    "msg": [  
        "OK Done!, Wait for approx three minutes and put the following URL in your browser and get greeted: [lb-1053362185.ap-southeast-2.elb.amazonaws.com](http://lb-1053362185.ap-southeast-2.elb.amazonaws.com/)"  
    ]  
}

Thank you