DESIGNING A WEB-APPLICATION IN AWS INFRASTRUCTURE

Description:

In this project I am going to deploy a web-application in AWS infrastructure.

The web application would be highly scalable and fault tolerant to prevent from uncertain spike in traffic.

Key Services that needs to be deployed:

- **Vpc**: To make the web application secure I am going to deploy everything using the VPC free service provided by AWS. This VPC would include a IP range in CIDR block 10.0.0.0/16, That gives a total of 65,536 IP address out of which 5 would be used for Internal AWS own service and as there wouldn't be any software license issue I would keep the tenancy as default.

- **Subnet:** I have created three subnets with non-overlapping CIDR range, out of which two would be public subnet and one would be private one because we need a minimum of two public subnets for ALB. The third subnet would be a private one and would be used to launch EC2 machine in it.

- **Route Table**: Here in this project I have created two route table Public route table and private routed table where public route table will direct the ALB to Internet Gateway and private route table is for EC2 in private subnet.

- **Security Group**: I have created two security group named as PrivateSecurityGroup and PublicSecurityGroup where public security group would restrict the traffic to EC2 only from ALB and a SSH port would be open with IP as the IP of the BASTION host.

- **ALB-Security-group:** The Alb security group is defined to take input from end user over HTTP and HTTPS protocol
- **EC2:** I have created two EC2 machine:

    - EC2 that would be serving the web application, the type of EC2 machine that needs to be deployed is kept as parameter and also for AMI-ID it is parametrised.
    - The second EC2 instance would be serving as BASTION host whose sole purpose is to SSH into main EC2 and see the configuration whenever required. As this EC2 sole purpose is to SSH only I have kept it at t2. micro

- **EIP:** I have created an Elastic Ip and attached it to BASTION host in Public Subnet and attached public Security Group, IGW to it.

- **Attachment of ALB:**

I have attached an ALB that would be serving as an end-point to the end users. This ALB has HTTP listener only as for HTTPS I have to use ROUTE 53 and ACM which are costly service.

The HTTP listener has two ports open Port-80 and Port-443

The ALB has HTTP listener which would read the end user request and transfer the traffic to the two Target group which contains private EC2 machine in it

There are two target groups which are connecting to two different subnets located in two different AZ's.

- **CloudWatchAlarm:** I have created a Cloudwatch Alarm that would be tracking CPU utilisation of both EC2 machine and if it goes beyond 70 percent.

Then a SNS topic would be triggered whose subscription are given as parameter. It could either be a phone number or email.

Salient points that needs to be implemented to create the design:

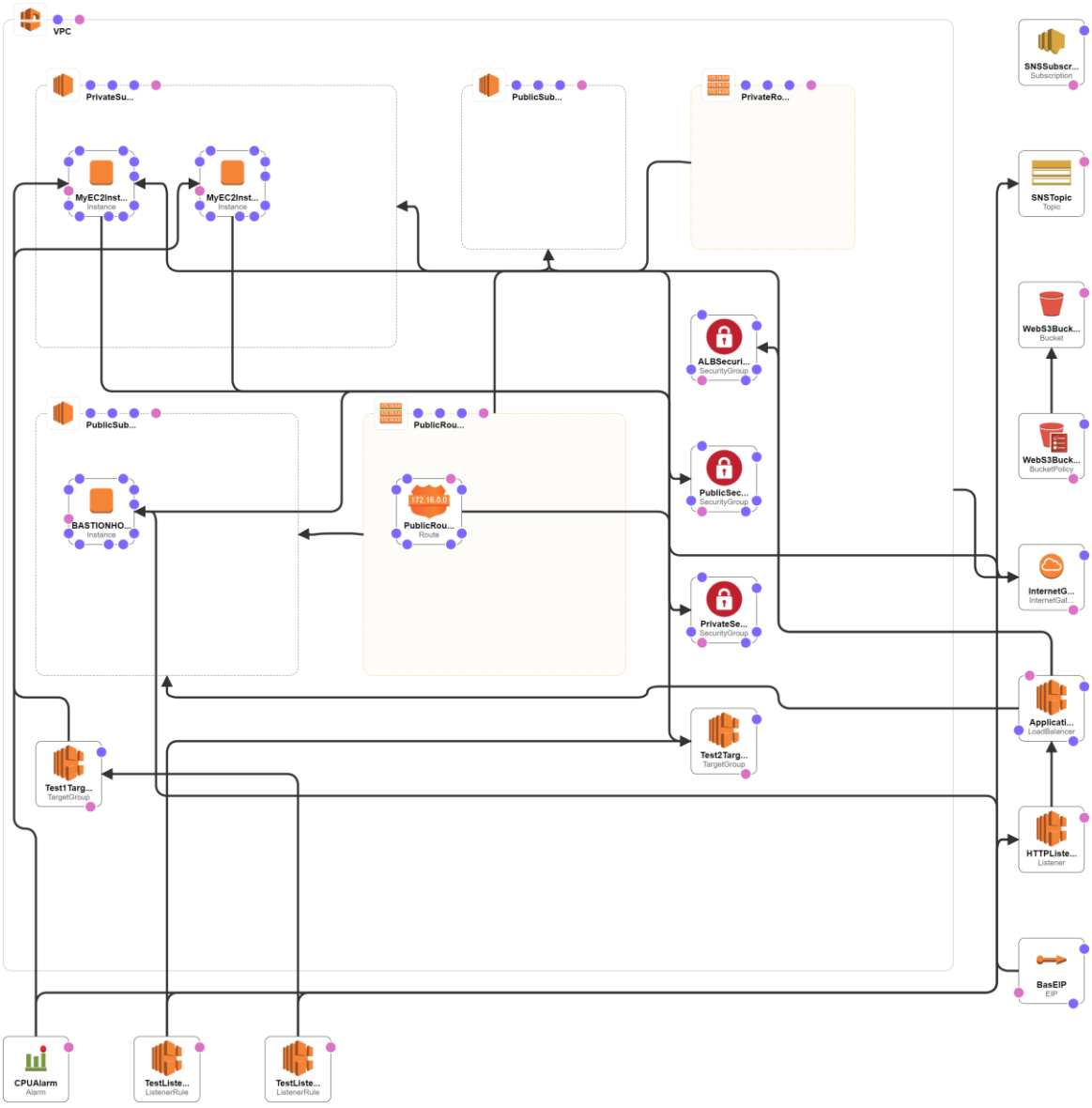
- Mounting of extra EBS volume to /var/log using EC2 user data
  USER DATA:

```
#!/bin/bash
yum update -y
sudo mkfs -t ext4 /dev/xvdb/
sudo mkdir /var/log
sudo mount /dev/xvdb /var/log
```

- Create a S3-log-bucket that would keep the log. The log would be transferred from EC2 /var/log.
  There would be Bash script that would be set up in EC2 machine, this bash script will be check the mount point if the mount point exceed a threshold say 70% of its total space. All the logs would be transferred to S3 location using AWS CLI command.

- Created a generic script that would check the mount point and drop a mail in case in gets high

Design Diagram:



WORK-FLOW:

- User request for the service using Alb DNS that would be configured using Route 53 Cname way.
- The request crosses the ALB and hits the EC2 in the background, where the distribution would be in round-robin method of weightage 1:1
- The process completes in EC2 and requested data is sent to user
- If the processing gets high and CPU utilisation gets above 70% a Cloudwatch alarm would be set up to trigger a SNS
- A autoscaling group with launch configuration will also trigger to spin up new EC2 machine by this cloudwatch alarm
- All the data that is in rest or in transit using SSE AE-256 in S3 and for HTTPS it is secured by ACM
- BASTION host would be used to SSH into private EC2 whenever required

Parameters:

EnvironmentName : This would define what kind of EC2 machine needs to be launched.

development: t2.micro

productionLowLoad : m5a.8xlarge

productionHighLoad: r 6g.large

ImageId: Image Id of the ec2 machine for particular region and type:

KeyName : pem or ppk key name, This would be used to ssh from BASTION host

SubscriptionEndPoint : The end user who will be receiving notification either Email Or SMS