Cloud DevOps on AWS Platform – POC

**Create web application to achieve bellow objective.**

At the end of this POC, you will have created a web site using the following Cloud Services: EC2, EBS, ALB, S3, Secrets Manager, Session Manager, RDS and CloudWatch alarm using **Terraform, Infrastructure as a code**.

You will have created an Elastic-Load-Balancer, two **Linux/Windows** server running Apache/IIS and a MySQL Database. Both the Apache/IISs servers will have websites running on ports 80 (A total of 2 different web pages).

The Elastic-Load-Balancer will distribute traffic to the **2 Linux servers OR 2 Windows servers** in a round-robin fashion. This means that requests to the Elastic-Load-Balancer on port 80 will be re-directed to the Apache/IIS servers listening on port 80.

**Stage 1: Building VPC private and public subnet**

* Use ap-southeast-1 region to create infra.
* Create 2 private and 2 public subnets in different availability zones [ap-southeast-1a & ap-southeast-1b]
* Each subnet should have minimum required IPs to host at least 5 EC2 servers.
* Configure one NAT Gateway in each availability zone and update route tables to route outbound traffic to internet via respective NAT in the availability zone.
* Route table configuration must be implemented as per the best practice.

**Stage 2: Prerequisite setup and building the web servers**

* Create secret for database password under secrets manager.
* Create IAM role for EC2 instance with access to use session manager and secrets manager.
* Launch 2 instances – Two Linux/Windows to meet the following objectives:
* The instances should be of type “**t2.micro**”.
* Web servers should not have public ip addresses attached.
* The **1st Linux/Window instance** should reside within region ap-southeast-1 within availability zone ap-southeast-1a & **2nd Linux/Window instance** should reside in availability zone ap-southeast-1b
* Apart from root volume, each instance should use **a 1 GiB attached EBS volume** and contain valid partition tables with one partition. **The partition should contain a valid file system**.
* The file system residing on the EBS volumes should be **mounted automatically** upon reboot of the instances.
* The instances should serve web pages via appropriate services such as Apache/IIS. These services should start automatically upon boot.
* The instances should serve a web page “**index.html**” containing well-formed HTML displaying the text "**Hello VF-Cloud World – running on Linux/Windows1 (Or Linux/Windows2)** – on port 80. The HTML files should reside on the file system within the **previously created EBS volume** and be **served as the default document from the web server root**.
* The instance should use Security Groups effectively to allow administration and serve HTTP.
* Configure the private instance in such a way, that you can connect to the instance using session manager. You don’t need a bastion or jump instance in this case.

**Stage 3: Configuring the Application Load Balancer**

Create a Load-Balancer (ALB) with the following specification:

* The ALB should be created in the ap-southeast-1 region.
* The ALB should accept connections on ports 80.
* The **Healthy Threshold** for the ALB to be **set to 2**.
* Deliver traffic to the instances created in Stage 1 – i.e. both the Linux/Windows servers will be registered to this Elastic Load Balancer. Requests on load balancer would be load balanced on the servers.

**Stage 4: Configuring MySql Database**

* Create a **MySql RDS database** with minimum size.
* Don’t hardcode the password in your terraform code, rather **use secrets manager** to fetch database password.
* Mandatory config details:
* **Port: 3306**
* **Multi-AZ: No**
* **Engine: MySql**

**Stage 5: Monitoring Alarm**

* Create CloudWatch Alarm for Memory and CPU usage for the EC2 instances. Set threshold as 80%.

**Upon completion, please provide following:**

1. Terraform code (Entire Folder in Zip Format) along with Terraform StateFile.
2. Screen-Shots of following:

* Connect to instances using session manager. Run ‘**df -k**’ command to show all the filesystems on both the instances. These screen shot should be named as **screen-shot1a** and **screen-shot1b**.
* Change directory to filesystem hosted on attached EBS volume. ‘**cat index.html**’ on both the instances. These screen shot should be named as **screen-shot2a** and **screen-shot2b**.
* On an instance, fetch RDS password from secret manager using AWS CLI commands and connect to database using CLI command. The screen shot of the commands and output should be named as **screen-shot3**.
* Run **stress test** on one of the instances to exceed CPU usage above 80%. Once CloudWatch alarm is in alarm state, Take snapshot and name as **screen-shot4**.