Automated HA Deployment of MyBB

Steps to run the Script

* Login to Console using below credentials
* Create an Ec2-Keypair for SSH access to instances
* launch a Stack from the automation template
* Once Automation Template is ended launch the MyBB appliation using load balancer name mention in the output variable
* you can evaluate the work by below credentials
* URL :- **https://moingodil.signin.aws.amazon.com/console**

username :- Viewer

Password :- p0\*Dj\*OgRFK\*

* the above account has read only access till 15 th feb 3 PM IST
* MyBB application Administrator Account:

Username: admin

Password: 1234

Requirements

* Deploy myBB (A PHP and mySQL based forum application) in AWS
* Use CloudFormation and AWS
* Need High Availability (span across more than one Availability Zone)
* Need Configurable Deployment
* Need Automated Deployment
* Need Scalability

What the template does

* Webservers under ELB across Multiple Availability Zone
* Multiple Availability Zone is used to optimize for High Availability across zone failures
* RDS Supported DB is used – AWS Managed scaling & availability
* Auto scaling group to scale web servers based on CPUUtilization
* Data Recovery can be performed using RDS Read Replica.
* Alarms are set for monitoring

Cloud Architecture

Internet Gateway

VPC

AmazonELB

AutoScaling Group

Web Server (EC2)

Web Server (EC2)

Web Server (EC2)

Multiple Availability Zones

Multi -AZ

DB s

MySQL Slave (Read Replica)

Mysql Master RDS Instance

1. Security :- The entire infrastructure is encapsulated into a VPC; all Internet traffic goes through:VPCInternetGateway to Internet for VPC components

* There are two main Security Groups which separate the internet-facing parts of the architecture,meaning the web-servers (and associated components), from the private parts, meaning the database mainly.
  + WebServerSecurityGroup: HTTP/HTTPS access permitted from outside.SSH permission is granted to desired IP ranged entered while creation of stack
  + DBSecurityGroup: Database, access permitted only from Web Server to DB Server.
* There are two Subnets associated within each availability zone:
  + WebServerSubnetA/B : web servers stack, the subnets are publicly accessible.
  + RDSSubnetA/B: database stack, the subnets are not publicly accessible.
* The routing for these subnets is as follows:
  + RouteTable: opens traffic from the webserver subnets to the Internet.
  + PrivateRouteTable : .for security reasons the private subnets have no access to the outside world.

1. Scalability

* Web Stack: the web servers are governed by an AutoScalingGroup and sit under an ElasicLoadBalancer for load-balancing and fault-tolerance.
  + Vertical scaling: web servers can be upgraded to larger memory/compute/storage capacities without downtime.
  + Horizontal scaling\*: The AutoScalingGroup implements policies for scaling up or down based on CPU usage metrics of the nodes (implemented via CloudWatch Alarms).
* Multi A-Z Database: The Mysql cluster has a Multi-AZ master (writer) DB and also a read replica at this time.
  + Vertical scaling: instances can be upgraded to larger memory/compute/storage capacities without downtime.
  + Horizontal scaling: with Multi AZ the DB is already replicated to multiple availability zones. Read replicas in Amazon RDS for MySQL provide a complementary availability mechanism to [Amazon RDS Multi-AZ Deployments](https://aws.amazon.com/rds/details/multi-az/). You can use read replica promotion as a data recovery scheme if the source DB instance fails
* Networking infrastructure:-All networking components such as ELBs, Autoscalling Groups,InternetGateway, VPC Router,etc are scaled-out by the AWS ecosystem.

1. High Availability

* Web stack: Both the Autoscalling Groups and ELB instances which govern the web servers currently span over two Availability Zones.
* Database Stack: You also benefit from enhanced database availability when running Multi-AZ deployments. If an Availability Zone failure or DB Instance failure occurs, your availability impact is limited to the time automatic failover takes to complete: approx two minutes. We use Read Replica to increase performance and Data recovery in case source DB instance fails.

1. Monitoring

* The ELB for the web servers commits logs into an S3 Bucket (every 5 minutes).
  + Following CloudWatch alarms are defined:
    - * Scaling up alarm: send email on Autoscalling Group events (when CPU usage is over 70%).
      * Scaling down alarm: send email on Autoscalling Group events (when CPU usage is under 40%).
      * Billing alarm: send email when costs exceed a threshold USD amount specified in parameter while creating a stack.
      * Unhealthy Host Host Alarm
      * High Request Latency Alarm
  + Alarms will notify the "OperatorEmail " address when triggered.

1. Further improvements

* The Route53 can be use to avoid ELB as a single point of failure. Route53 can redirect traffic to another ELB in case current ELB fails to respond.
* Adding S3/CloudFront support for uploaded files using AWS PHP SDK
* More Alarms for High Network Activity/instance issues
* Add an \*\*ElasticCache/Memcache\*\* deployment to improve performance.
* Add Tags to Resources
* use more than 2 availability zone