Contents

Problem Statement

MyBB is open source forum software built on php and MySql. Given the powerful set of features around forums mybb brings to the table, A lot of enterprises deploy the same onto AWS, Azure, Google. cloud. While the mybb installer is fairly straightforward in terms setting up app and the MySql database. The challenge for most enterprise comes at the point of addressing various concerns around

- ✓ Security
- ✓ Monitoring
- ✓ High Availability
- ✓ Scalability.
- ✓ Management application- addresses deployment specific settings and running application insights.

The conversation in this document is pertaining mybb version 1.8.11, the same can extended to mybb2 and later version as well. The expectation is to extend the MyBB deployment to address the above listed concerns.

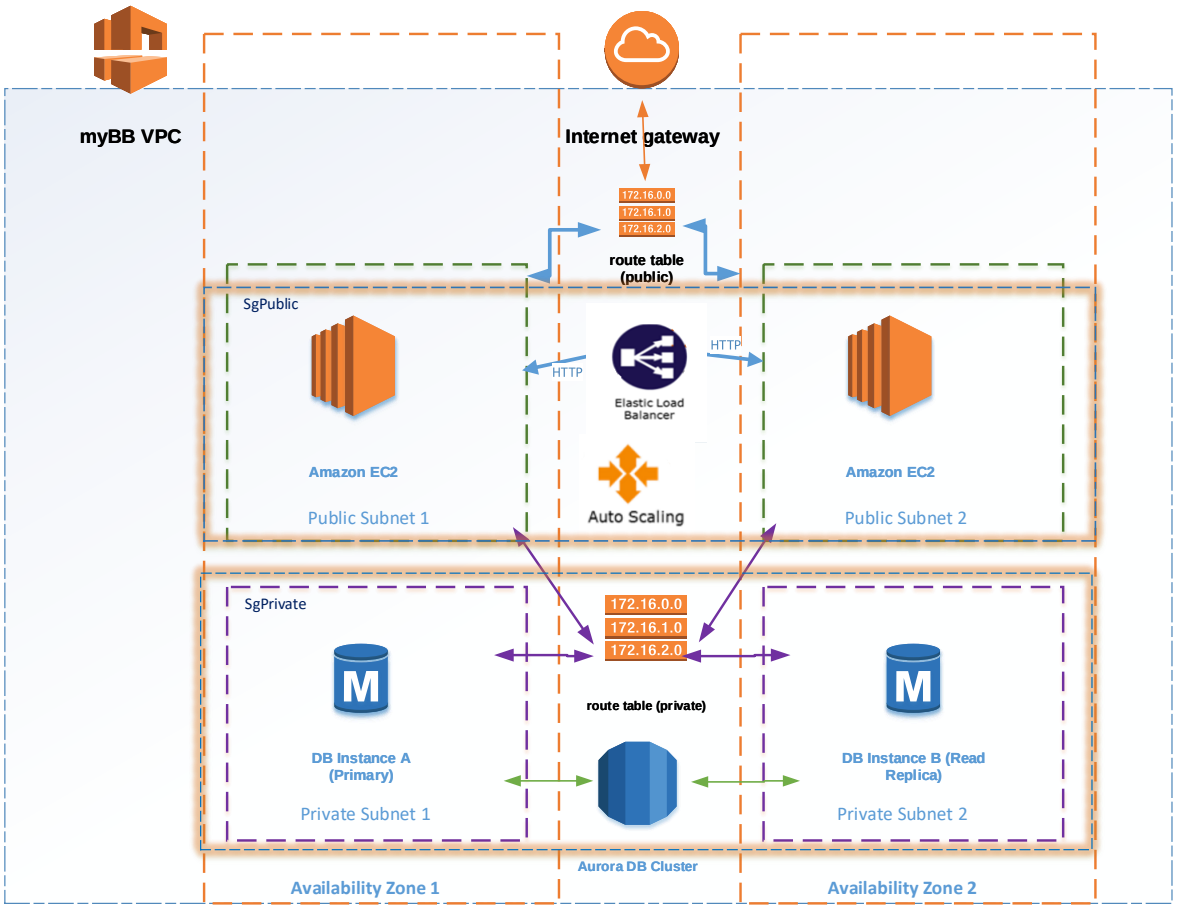
Solution Concept

Most cloud offerings today have ready to use construct which address the concerns around security, monitoring, high avail, scalability, Management application to address deployment specific setting and running application insights been specific to the application requires bespoke development.

MyBB is a tiered application architecture consisting of a web, application and database tiers. The web and app tier are compute workloads and database. The further section explains the solution in detail

- ✓ Deployment Architecture and concern addressability.
- ✓ Improvement Suggested
- ✓ Solution Specifics
- ✓ Executing the stack (cloud formation).
- ✓ Evaluation access.
- ✓ Management Application walkthrough.

Deployment Architecture



The MyBB cloud formation script is used to deploy the complete MyBB application and infrastructure in AWS on the region of choice.

MyBB application will be deployed in a VPC (**vpc-mybbadmin**). The MyBB application infrastructure is deployed in this VPC. *The VPC and constructs defined below addresses the security requirements.*

For communication to the internet an internet gateway (**inetgw-mybbadmin**) is attached to the VPC.

To start with routing isolation, we have used 2 route table **PublicRouteTable** (attached to internet gateway) and **PrivateRouteTable** for facilitating internal traffic are implemented.

Public subnets (**PubSubnet1, PubSubnet2**) are used across 2 availability zones, the web and app tier will be deployed in these subnets. Considering these are web application and need to be internet facing hence these are in the public subnet.

Private subnets (**PvtSubnet1, PvtSubnet2**) are used across 2 availability zones, the database tier the Aurora RDS cluster and RDS instance will be deployed in these subnets.

To restrict the traffic going in/out of the instance we use Security groups namely

- **SgPublic**: HTTP/HTTPS/SSH access permitted from outside.

- **SgPrivate**: Database, access permitted only from web tier to DB tier (using defined ingress rules inbound (SgPrivate) and outbound (SgPublic) restricting to aurora port)

Web Tier

Web tier the instances are deployed in public subnets across an elastic load balancer which provides for load – balancing and fault tolerance. Instances managed by auto scaling group with scale up on (> **90% CPU**) and scale down (< **40% CPU**). The ELB addresses the concerns for availability by automatically **routing** traffic across multiple availability zones, receives traffic by detecting unhealthy instances and rerouting traffic across the remaining healthy instances.

In addition, it also addresses the concerns on scalability via load-balancing traffic across instances with crosszone enabled.

From scaling stand point the auto-scaling provides for the horizontal scaling for vertical scaling instance type can be changed to best meet the requirements.

Database Tier

Aurora RDS cluster deployed with Multi-AZ deployment with a master and read replica provides for high availability (fail over to secondary) and for scaling needs vertical scaling change instance type would address the same. Horizontal scaling can be managed with multiple read replicas (~ 15).

Note: Scalability and Availability of the MyBB solution is addressed both at the web and database tier as mentioned in above paragraph.

Monitoring the overall solution

ELB logs are written to S3 bucket. CloudWatch Alarms for following scenario are defined.

- Scaling up alarm- Sends an email out in case the CPU of the web instance goes over 90%
- Scaling down alarm - Sends an email out in case the CPU of the web instance below over 40%

Improvements Suggested

- **Cloud Formation script** could read the deployment settings from Dynamo DB instead of parameter using custom resource.
- **CloudWatch Alarms** for
    - Database instance failures
    - Abnormal BW usage
- **Use Route 53 or a public dns name**: Current implementation we use the dns name of the load balancer, add a cname parameter for the same in the template and implement via Route 53, public dns.
- **Cost Optimization:**
    - Gather cloud watch logs for the instances to automatically down size instance size if we see metrics shown under- utilization.
    - Move the web instances to spot for further cost optimization and mybb to docker implementation.
    - Gather database cloud watch metrics to identify movement to a smaller instance size (db.t2.small).
- Network ACL would be better preferable over security groups.
- **Outbound access for DB servers via NAT**: It could be useful to allow outbound access to the Internet for the machines residing in the private subnet of the VPC (for updates and such) by using a **NAT Gateway**.
- **ElastiCache/Memcache** deployment to improve performance.

Solution Specifics

The complete MyBB application deployment and infrastructure set up is available in a CloudFormation script can be found here.

Management application is written .NET, angularjs deployed on elastic beanstalk, implementation and installation details can be found here.

Executing the Stack

To execute the stack

- Create an EC2 Keypair (SSH Access)
- Launch the cloud formation stack from AWS Console or the management application
- Go to the URL in the “ELBDNSName" output variable for the live MyBB application.

Evaluation Access

AWS Console access
URL: https://747298667540.signin.aws.amazon.com/console
Username: mybbuserOne
Password: b!ctgzUqqADs

Notes:
Currently there are **2 stacks** deployed in region **uswest1**
Read-only access granted for: **CloudFormation, EC2, RDS, S3, SNS and CloudWatch**.

MyBB application
URL: http://ec2-52-53-151-145.us-west-1.compute.amazonaws.com
Administrator Account:
Username: admin
Password: 1234

Management Application

URL: http://ec2-52-53-119-60.us-west-1.compute.amazonaws.com/login.aspx
Username: xo@crosssover.com
Password:mexx@2010

Management Application Walkthrough

Management Application built on .NET , angular js and Dynamo DB , deployed on Elastic BeanStalk. The dynamo db has 2 tables users (stores user name password), storage settings (stores deployment specific settings for mybbadmin)

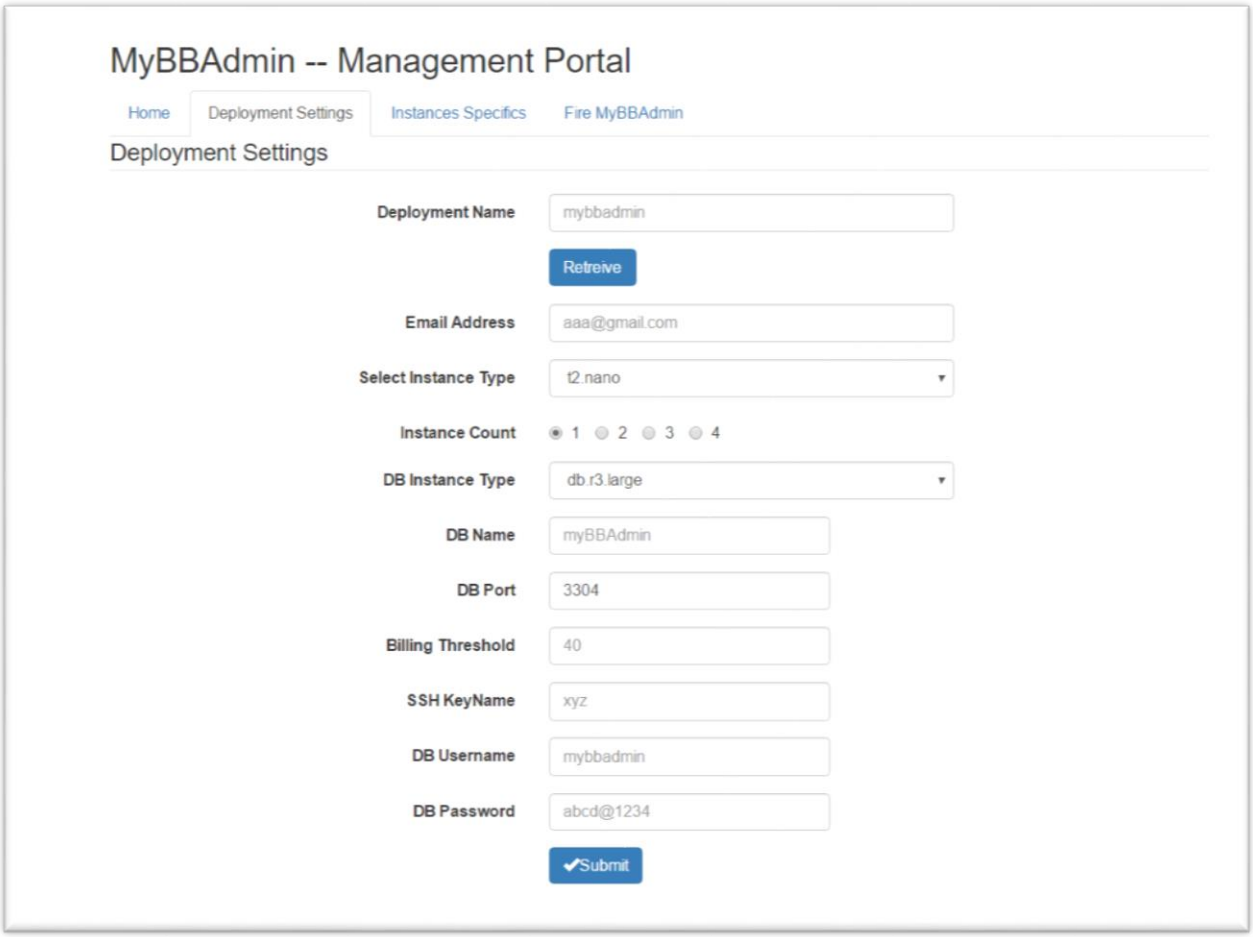
Login

Login into the application via http://ec2-52-53-119-60.us-west-1.compute.amazonaws.com/login.aspx

Username: xo@crosssover.com
Password:mexx@2010

Deployment Settings

MyBB Application deployment parameters for MyBB application indicated in the screen below



Deployment Settings

You can retrieve the existing mybb deployment settings by

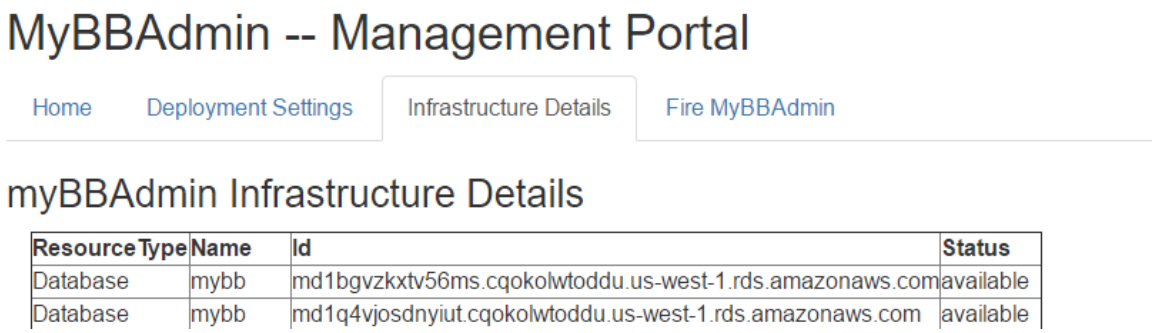
1.Entering Deployment Name: mybbadmin

2.Press -> Retrieve.

3.update existing deployment based on deployment name or add new.

Running Application Specifics

The running instances and database instance along with state information is visible in Infrastructure details screen



Deploy MyBBAdmin Stack

The stack can be deployed based on the deployment settings set in the 'Deployment Settings' UI via Fire MyBBAdmin.