

# Overview of the AWS Architecture for a startup project

Nazmul Kaonine

## 1. Solution Diagram

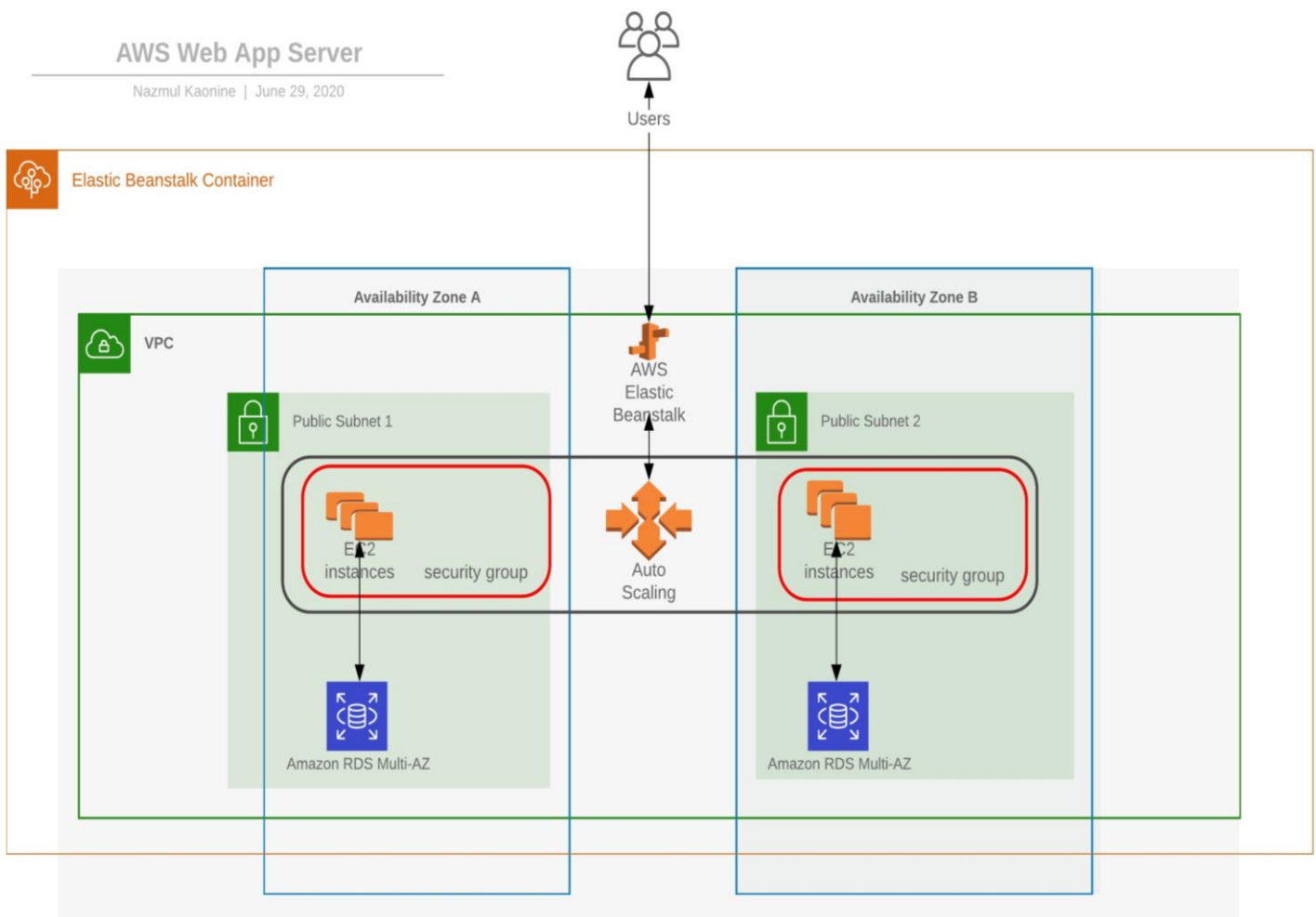


Fig 1: AWS Solution Diagram (created using [www.lucidchart.com](http://www.lucidchart.com))

The diagram reflects a high-level view of the solution containing an elastic beanstalk environment, ec2, VPC and other services that will be discussed in the latter part of this paper.

## 2. AWS Services & justifications

The following AWS services were used to implement the solution for the architecture:

### a) AWS Beanstalk:

An application and environment were created as shown in the images below. **Reason:** Easy-to-use group of services for deploying web apps.

Fig 2: Application name: TakeHomeExamApp

<input type="radio"/>	TakeHomeExamApp	Takehomeexamapp-env-1	2020-06-27 22:08:57	2020-06-27 22:08:57 UTC+1000	arn:aws:elasticbeanstalk:us-east-1:437080257856:application/TakeHomeExamApp
<input type="radio"/>	Takehomeexamapp-env-1	Ok	TakeHomeExamApp	2020-06-27 22:12:12 UTC+1000	2020-06-27 22:43:20 UTC+1000
					Takehomeexamapp-env-1.eba-6mtdcc3q.us-east-1.elasticbeanstalk.com

Fig 3: Environment name: Takehomeexamapp-env-1

Inside the above environment, the solution will be built using configuration settings.

### b) Amazon EC2:

Shown below are the list of instances at a particular time in the Amazon EC2 service portal. **Reason:** Enables computing services in the cloud.

<input type="checkbox"/>	Takehomeex...	i-03525b766c16c7a6b	t2.micro	us-east-1b	running	2/2 checks passed	None	
<input type="checkbox"/>	Takehomeex...	i-0c703418c492384af	t2.micro	us-east-1a	running	2/2 checks passed	None	

Fig 4: List of instances

To verify if the instances were created using the custom AMI, here is an image of the instance details.

VPC ID	vpc-0f2914f7c68d16e71 (TakeHomeExamVPC)	AMI ID	TakeHomeExamAMI (ami-09158d064243a500a)
Subnet ID	subnet-069d4d4999e20d78a (Subnet 2)	Platform details	Linux/UNIX
Network interfaces	eth0	Usage operation	RunInstances

Fig 5: Details of a created instance

### c) Custom AMI:

Amazon Linux AMI was created that was used to create the LAMP server solution. **Reason:** All instances were built using same computing config.

AMI ID    ami-09158d064243a500a                      AMI Name    TakeHomeExamAMI

Fig 6: Custom AMI name: TakeHomeExamAMI

### d) Custom Security Group allowing SSH & HTTP requests:

A custom security group was created to approve SSH and HTTP requests. **Reason:** All instances will be allowing requests using this protocol.

Security groups    [awseb-e-geuq29eijx-stack-AWSEBSecurityGroup-1AN95DI65VCC9. view inbound rules.](#)  
[view outbound rules](#)

Fig 7: The security group from that was customized.

The inbound rules are shown below:

HTTP	TCP	80	sg-0a0cd91c18d2ec335 (awseb-e-geuq29eijx-stack-AWSEBLoadBalancerSecurityGroup-W9519AXLNMKB)	-
SSH	TCP	22	0.0.0.0/0	-

Fig 8: Requests allowed: SSH & HTTP

### e) Load Balancer:

A load balancer was enabled inside the Elastic Beanstalk of type classic. **Reason:** To enable personalized auto-scaling.

Load balancer                      [Listeners: 1](#)  
Load balancer type: classic

Fig 9: Load balancer type: classic

### f) Auto-Scaling:

The following configuration settings shows that the environment increases / decreases instances (by 1) between 2 to 8 instances if the upper threshold percentage of 60 or lower threshold percentage of 30 are met upon Network Output. **Reason:** This provides better performance by distributing workload.

Capacity	Lower threshold: 30
	Max: 8
	Metric: NetworkOut
	Min: 2
	Period: 5
	Placement:
	Scale down increment: -1
	Scale up increment: 1
	Scaling cooldown: 360 seconds
	Statistic: Average
	Unit: Percent
	Upper threshold: 60

Fig 10: Auto-scaling settings

#### **g) RDS:**

A RDS was deployed in multiple availability zones. **Reason:** This ensured multiple copies of data providing safer data storage.

Availability: High (Multi-AZ)

Fig 11: RDS deployed with high availability settings.

#### **h) Virtual Private Cloud (VPC with subnets):**

This custom vpc contains two public subnets deployed in two different regions “us-east-1a” and “us-east-1b”. **Reason:** The vpc allows the implementation on the startup’s own data architecture. The public subnets in multiple regions saves the system if one of the regions fail thus providing a safer system. They also have an internet gateway and route tables.

Database subnets: subnet-0d46d1023b22053df, subnet-069d4d4999e20d78a  
 Instance subnets: subnet-0d46d1023b22053df, subnet-069d4d4999e20d78a  
 Public IP address: enabled  
 VPC: vpc-0f2914f7c68d16e71  
 Visibility: public

Fig 12: 2 VPC with public subnets

**i) Custom key pair:** All instances were made using custom key pair. **Reason:** To enable easy instance accessibility.

EC2 key pair: takeHomeExamKeyPair

Fig 13: Custom Key pair name: takeHomeExamKeyPair

#### **j) Email notification for important events:**

An email was set up to receive event notification of the system. **Reason:** One use is to receive updates of system failure to take immediate action.

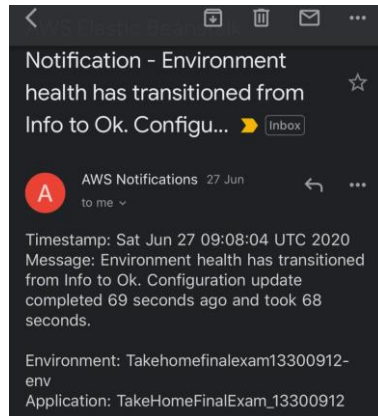


Fig 14: Event email notification from AWS

#### Overall solution health:

The solution implemented displays a "OK" health indicating that the services are up-to-date and running.



Fig 15: Health status of the Elastic Beanstalk system

## Assumptions

1. The auto scaler will increment or decrement the number of instances upon high or low workload respectively.
2. AWS services account has sufficient credit to let the services thrive as long as its needed.
3. Platform used for the server is PHP 7.3 running on 64bit Amazon Linux/2.9.7.
4. Database engine is kept as mysql. (username: naz pass: Naz123456)
5. Availability zones for instances is set to any.
6. Type of instances is t2.micro.